Development and validation of a novel marker tracking approach based on the low-cost Microsoft Kinect v2 sensor for assessing lower limb biomechanics during single-leg squat and treadmill gait

Author: ALESSANDRO TIMMI

ORCID: 0000-0001-9831-3344

Thesis submitted in total fulfilment of the requirements of the degree of

Doctor of Philosophy

March 2018

THE UNIVERSITY OF MELBOURNE
FACULTY OF MEDICINE, DENTISTRY & HEALTH SCIENCES
ST VINCENT’S DEPARTMENT OF SURGERY
Abstract

Pubescent females are twice more likely to suffer a non-contact ACL injury than their male counterparts. This disparity has been correlated with multiple concurrent factors, including biomechanical, anatomical and hormonal changes. ACL ruptures require serious and costly surgical interventions, which could be avoided if subjects at higher risk of injury were more carefully monitored and trained. Three-dimensional motion analysis is required to identify individuals at risk of ACL injury. Multi-camera optical systems are the gold standard for 3D motion capture, but they are very expensive and cumbersome. The aim of this thesis was to make motion analysis more accessible, developing an affordable and compact 3D motion tracking methodology, alternative to conventional multi-camera systems.

A novel tracking approach was developed using Microsoft Kinect v2, employing custom-made coloured markers and computer vision techniques. This methodology was denoted as Kinect coloured marker tracking (KCMT). The accuracy of KCMT relative to a conventional Vicon motion analysis system was measured performing two Bland-Altman analyses of agreement, the first using single-leg squat (SLS) as benchmark task, the second using treadmill locomotion.

The objective of the first study was to determine if KCMT-derived sagittal joint angles of the lower limb were accurate enough to allow discerning individuals at risk of ACL injury from those not at risk. Eleven healthy participants were asked to perform three SLS trials, while three-dimensional marker trajectories were simultaneously recorded using Vicon and KCMT respectively. Joint angles from the two systems were calculated via inverse kinematics using OpenSim. The limits of agreement (LOA) of the joint angles were −16°, 13° for hip flexion, −12°, 0° for knee flexion and −12°, 9° for ankle flexion. These results indicated that the agreement between KCMT and Vicon was joint dependent, and that further work was required for the novel methodology to replace conventional marker-based motion capture systems for the identification of ACL injury risk from SLS data.

In the second study, an improved data collection protocol for the KCMT was used. Twenty participants were recruited, and markers placed on bony prominences near hip, knee and ankle. Three-dimensional coordinates of the markers were recorded during treadmill walking and running. The LOA of marker coordinates were narrower than −10 and 10 mm in most conditions, however a negative relationship between accuracy and treadmill speed was observed along Kinect depth direction. LOA of the knee angles measured in the global coordinate system were within −1.8°, 1.7° for flexion in all conditions and −2.9°, 1.7° for adduction during fast walking, suggesting that KCMT may be capable of discerning between subjects at risk of ACL injury and controls. The proposed methodology exhibited good agreement with a marker-based system over a range of gait speeds and, for this reason, may be useful as low-cost motion analysis tool for selected biomechanical applications.
Declaration

This is to certify that:

- the thesis comprises only my original work towards the degree of Doctor of Philosophy except where indicated in the Preface,
- due acknowledgement has been made in the text to all other material used,
- the thesis is fewer than 100,000 words in length, exclusive of tables, bibliographies and appendices.

Alessandro Timmi
Melbourne, 2nd March 2018
Preface

I hereby confirm the complete originality of this thesis. Chapter 9 was published in the journal *Medical Engineering & Physics*. I contributed 51% to the content of this chapter. The reference for this work is the following:


In the above study, I was involved in the study design, software development, subjects’ recruitment, data collection, data analysis, statistical analysis, manuscript drafting and manuscript review. Mr Gino Coates was involved in the study design, software development, data analysis and manuscript review. Ms Karine Fortin was involved in subjects’ recruitment, data collection and manuscript review. Dr David Ackland was involved in study design, data collection and manuscript review. Prof Adam Bryant was involved in study design, data collection and manuscript review. Prof Ian Gordon was involved in study design, statistical analysis and manuscript review. Prof Peter Pivonka was involved in study design, data collection and manuscript review. All authors read and approved the final manuscript.

The remaining chapters of this thesis contain unpublished material not submitted for publication.

I acknowledge the MIFRS and MIRS scholarships provided by The University of Melbourne, which allowed me to conduct this research. Part of this thesis was supported by a Defence Health Foundation grant (2015).
Acknowledgements

I would like to express my sincere gratitude to my supervisors, Prof Peter Pivonka, Prof Adam Bryant and Dr David Ackland for their continuous support during my PhD studies. They have always provided me with extensive professional and personal guidance, and I appreciate all their contributions of time and ideas to make my PhD experience productive and stimulating. I’m also grateful to Prof Ian Gordon, for his advice with statistics and for his encouragement during the challenging process of publishing our paper.

I am grateful to my fellow lab mates Karine Fortin and Tim Sayer for their help with data collection in the motion capture lab at the Centre for Health, Exercise & Sports Medicine of The University of Melbourne. Without their efforts with recruiting participants and recording data using the Vicon motion capture system, this research would not have been possible. I’m also very grateful to my friend and fellow PhD student Gabriele Imbalzano for the time spent together and for his optimism, which made this experience much more enjoyable. My profound gratitude goes to Sensei Con Kassis, for welcoming me into his Dojo, for making me feel an important part of his Karate family, for being a great mentor, and for generously sharing his wisdom and technical knowledge.

Lastly, I would like to thank my mum, dad, brother and grandparents in Italy for supporting me spiritually throughout these four and a half years, and for always remaining positive and cheerful despite the long distance between us. Most of all, I would like to thank my wife Silvia for her love, for taking a leap of faith to come to Australia when I was offered this PhD position, and for her continuous support and encouragement during this journey.
# Table of contents

1.1 ACL injury risk in young girls ................................................................. 20
1.2 Internal risk factors for ACL injury in girls ............................................. 22
  1.2.1 Stage of development ...................................................................... 22
  1.2.2 Hormonal changes .......................................................................... 22
  1.2.3 Musculoskeletal changes ................................................................. 23
    Knee joint laxity .................................................................................. 23
    Flexibility ............................................................................................ 23
    Muscular development ........................................................................ 24
  1.2.4 Landing biomechanics ..................................................................... 24
  1.2.5 Pronation ....................................................................................... 26
1.3 Assessment of ACL injury risk ................................................................. 26
  1.3.1 Markerless technology .................................................................... 27
  1.3.2 Musculoskeletal modelling .............................................................. 30
1.4 Aims and structure of this thesis ............................................................... 34
1.5 References ............................................................................................ 37
2.1 Introduction ............................................................................................ 45
2.2 Methods ................................................................................................. 46
  2.2.1 Participants .................................................................................... 46
  2.2.2 Experimental protocol ................................................................... 46
  2.2.3 Data analysis ................................................................................ 47
    Ad hoc Kinect v2 markerset ................................................................. 47
    Custom OpenSim model for Kinect v2 ................................................. 49
    Statistical analysis ............................................................................. 50
2.3 Results .................................................................................................. 51
2.4 Discussion ............................................................................................. 53
2.5 References ............................................................................................ 56
3.1 Depth measurement ............................................................................... 58
3.2 Inclusion criteria of this literature review ............................................... 59
  3.2.1 Temperature drift ........................................................................ 60
  3.2.2 Accuracy of the centre of the depth sensor ..................................... 61
  3.2.3 Precision of the centre of the depth sensor .................................... 64
  3.2.4 Uniformity of accuracy of the depth sensor ................................... 65
  3.2.5 Uniformity of precision of the depth sensor .................................. 66
3.2.6 Effect of colour and reflectivity ................................................................. 68
3.2.7 Effect of motion ............................................................................................ 70
3.2.8 Optical lens distortion .................................................................................. 72
3.3 Discussion ........................................................................................................... 72
3.4 References .......................................................................................................... 74
4.1 Introduction ......................................................................................................... 76
4.2 Software architecture ......................................................................................... 76
4.3 Hue, saturation and value .................................................................................. 77
4.4 HSV filter ........................................................................................................... 78
  4.4.1 Factors influencing the selection of HSV tolerances ..................................... 79
  4.4.2 Experimental determination of optimal HSV tolerances .......................... 80
  4.4.3 Hue range for red primary .......................................................................... 81
4.5 Choosing colours based on HSV ranges .......................................................... 83
4.6 Designing coloured markers ............................................................................ 84
4.7 Marker detection algorithm ............................................................................. 84
  4.7.1 Marker detection using colour, infrared and depth data ............................ 85
  4.7.2 Marker centre correction ........................................................................... 88
4.8 Conclusion ......................................................................................................... 89
4.9 References ......................................................................................................... 89
5.1 Abstract .............................................................................................................. 91
5.2 Introduction ........................................................................................................ 91
5.3 Methods ............................................................................................................. 92
  5.3.1 Kinect v2 coordinate system ....................................................................... 92
  5.3.2 Vicon coordinate system ........................................................................... 92
    Ergocal .................................................................................................................. 93
    Vicon static calibration algorithm ....................................................................... 94
  5.3.3 Coordinate transformation in the Single-leg squat study ............................ 97
    Pose estimation hardware: L-frame ..................................................................... 98
    Pose estimation algorithm .................................................................................. 99
    Coordinate transformation from Kinect to Vicon space .................................. 101
    Transformation to OpenSim coordinate system .............................................. 102
  5.3.4 Coordinate transformation in the Treadmill Study .................................... 104
    Pose estimation hardware: reflective markers and 3D-printed array ............... 104
    Pose estimation algorithm .............................................................................. 105
    Coordinate transformation from Vicon to Kinect space .................................. 107
  5.3.5 Transforming multiple markers at the same time ...................................... 108
7.2.2 LOA with multiple observations per individual ........................................ 147
7.2.3 Confidence intervals of the LOA with multiple observations per individual..... 147
7.3 Conclusion ........................................................................................................ 148
7.4 Appendix ........................................................................................................... 149
  7.4.1 Confidence intervals of the LOA ................................................................. 149
7.5 References ......................................................................................................... 150
8.1 Abstract ............................................................................................................ 153
8.2 Introduction ..................................................................................................... 154
8.3 Methods ........................................................................................................... 154
  8.3.1 Participants ................................................................................................ 154
  8.3.2 Experimental protocol ............................................................................. 155
  8.3.3 Data analysis ............................................................................................. 159
     Pre-processing .................................................................................................. 159
     OpenSim models ............................................................................................ 160
     Scaling............................................................................................................. 161
     Inverse kinematics and synchronization ....................................................... 161
     Inverse dynamics ........................................................................................... 162
     Statistical analysis ......................................................................................... 162
8.4 Results ............................................................................................................ 162
8.5 Discussion ....................................................................................................... 164
  8.5.1 Limitations ................................................................................................. 167
8.6 Conclusion ...................................................................................................... 168
8.7 Appendix ........................................................................................................ 170
  8.7.1 OpenSim scaling parameters ................................................................... 170
  8.7.2 OpenSim inverse kinematics parameters ............................................... 172
  8.7.3 Number of markers and degrees of freedom ......................................... 173
8.8 References ...................................................................................................... 174
9.1 Abstract .......................................................................................................... 176
9.2 Introduction ..................................................................................................... 177
9.3 Methods ......................................................................................................... 178
  9.3.1 Participants ............................................................................................... 178
  9.3.2 Experimental protocol ........................................................................... 178
  9.3.3 Coloured marker tracking ...................................................................... 179
  9.3.4 Data analysis ............................................................................................ 180
9.4 Results ............................................................................................................ 181
9.5 Discussion ...................................................................................................... 184
List of figures

Figure 1. Magnetic resonance imaging (MRI) of a right knee, with focus on the ACL (red dashed line). Left: coronal view of the extended knee. Centre: coronal view of the flexed knee. Right: sagittal view of the extended knee. ................................................................. 20

Figure 2. Primary ACL reconstructions by age and sex from the Norwegian National Knee Ligament Registry. Data from Renstrom et al. (2008)................................................................. 21

Figure 3. Testosterone and estrogen hormonal concentrations across all Tanner stages in boys and girls. Data from Wild et al. (2012). ................................................................. 22

Figure 4. Decreased ratio of hamstring-to-quadriceps strength generates anterior-directed shear force on the tibia (red arrow), which is the primary cause of strain on the ACL (red dashed line) when the knee is near full extension ................................................................. 24

Figure 5. Microsoft Kinect v1, also denoted as Kinect for Xbox 360 ........................................................................ 28

Figure 6. The Kinect v1 disassembled. Behind the three lenses, there are from left to right: the infrared projector, the RGB camera and the infrared depth camera. Photo by Walter Galan (IFIXIT, licensed under CC BY-NC-SA 3.0).................................................................................. 28

Figure 7. Kinect v2 teardown. The RGB and depth camera are visible (first and second from left respectively), as well as the infrared LEDs (centre). Photo by Walter Galan (IFIXIT / CC BY-NC-SA 3.0). ................................................................. 30

Figure 8. Chart illustrating a possible OpenSim workflow. Extensions were used to indicate the following file types: C3D (Coordinate 3D files), TRC (Track Row Column files, also known as marker files), GRF (Ground Reaction Forces files), OSIM (OpenSim Model files), MOT (Motion files), STO (Storage files), XML (Extensible Markup Language files). ........................................................................ 33

Figure 9. Flowchart representing the initial outline of the research project (top) and the final outline of this thesis (bottom). After the accuracy of Kinect v2 pose estimation algorithm was found to be insufficient to discern subjects at risk of ACL injury from controls, a novel methodology to track coloured markers (KCMT) using Kinect v2 depth data was developed. Furthermore, the agreement between the proposed approach and a marker-based optical motion capture system (Vicon) was assessed ................................................................. 36

Figure 10. The Vicon markerset used in this study. Eight markers, here labelled in green, were used during static trials only ........................................................................ 46

Figure 11. The skeleton tracked by Kinect v2 pose estimation algorithm (in red), composed of 25 virtual joint centres (in white) ........................................................................ 47

Figure 12. The 25 virtual markers (in pink) attached to the OpenSim model, replicating the virtual joint centres tracked by the Kinect v2 pose estimation algorithm ........................................................................ 48

Figure 13. Left: anatomical locations of spine virtual joints (in white) were estimated from screenshots recorded using Kinect Studio. Right: the measured proportions were reproduced in the OpenSim markerset (in pink). ........................................................................ 49

Figure 14. Corresponding degrees of freedom of the spine in the OpenSim Gait2392 model (left) and in Kinect v2 skeleton (right). Top: lumbar bending. Middle: lumbar extension. Bottom: lumbar rotation ........................................................................ 50

Figure 15. The animation of a DLS generated using the OpenSim model and Kinect-derived joint angles. The incorrect pelvis tilt (left) was fixed by locking the lumbar extension to 0° in the OpenSim model (right) ........................................................................ 50

Figure 16. Synchronized hip, knee and ankle flexion angles derived from Vicon and Kinect data, recorded during DLS trials ........................................................................ 52
Figure 17. Bland-Altman plots of the hip, knee and ankle flexion angles derived from Vicon and Kinect v2 data, recorded during DLS trials. Differences (grey circles) and regression-based LOA (black lines) between the two measurement systems are reported. ........................................ 53
Figure 18. Differences between Kinect measurements and true distances. Data are from the current thesis and from reviewed studies. All datasets display similar trends for the wiggling error, typical of TOF cameras. Although the same approach for removing the constant bias was applied, there are residual differences between signals, which are due to the different protocols used for data collection and analysis. The figure legend reports in parentheses the corrections applied to each dataset, e.g. change of sign (sign) or bias subtraction (bias). .... 62
Figure 19. A sketch representing the data collection protocol used by Corti et al. (2015). The dashed rectangle indicates the starting position $k0$ of the target measured with respect to Kinect v2. The solid rectangle indicates the position $k$ of the target after a displacement, whose true value $d$ was measured by the robotic arm supporting the target. ................................. 62
Figure 20. Screenshots of Kinect v2 point cloud. Motion affects depth readings, generating artefacts on the surface of the spherical marker and of all moving objects in general. A) Slow speed. B) Moving fast towards the negative X direction. C) Moving fast towards the positive X direction. ............................................................................................................................ 71
Figure 21. The HSV (hue, saturation and value) polar coordinate system (picture by SharkD, licensed under CC BY-SA 3.0). ......................................................................................................................... 78
Figure 22. Colour image (left) and corresponding binary image (right) of a marker. In this case, the HSV filter was set to preserve the dark blue pixels of the marker. ........................................ 79
Figure 23. Colour image of a fast-moving blue marker recorded using Kinect v2, displaying the effect of motion blur. ......................................................................................................................... 81
Figure 24. Hue (left), saturation (middle) and value (right) of a colour image, featuring a fast-moving blue marker. Hue was less affected by motion blur, compared to saturation and value. ......................................................................................................................... 81
Figure 25. Hue ranges selected for KCMT markers (solid rectangles) and other tested colours (empty rectangles) .......................................................................................................................... 83
Figure 26. Toolkit for marker design: (a) coloured markers and corresponding paint tubes; (b) clear round surface savers were used as bases for coloured markers, to protect the thin layer of paint; (c) water-based construction glue used to attach the bases to the coloured markers. .......................................................................................................................... 84
Figure 27. A flowchart with pictures representing the essential steps of the marker detection algorithm .......................................................................................................................... 85
Figure 28. Flowchart of the developed marker detection algorithm, based on colour, infrared and depth data from Kinect v2.................................................................................................................................. 87
Figure 29. The centre "C" of the spherical marker was obtained extending the vector "OP" by the marker radius $r$ .......................................................................................................................... 88
Figure 30. Kinect v2 coordinate system is right-handed and its origin is located at the centre of the depth sensor .......................................................................................................................... 89
Figure 31. The coordinate system of the Vicon Ergocal static calibration object is centred on its inferior 90° corner. Source (Vicon 2007). .......................................................................................................................... 93
Figure 32. The Ergocal used in the motion analysis laboratory was aligned to one of the corners of the 2nd force plate .......................................................................................................................... 93
Figure 33. A flowchart of the Vicon static calibration algorithm (Woolard 1999; Vicon n.d.; Vicon 2007). .......................................................................................................................... 94
Figure 34. Determination of the rotation matrix of $O − xyz2$ with respect to $O − xyz1$..... 95
Figure 35. Determination of the position vector $d12$ of $O − xyz2$ measured in $O − xyz1$.. 96
Figure 36. Coordinate transformation of point P from the coordinate system before static calibration \( O - xyz1 \) to the coordinate system after static calibration \( O - xyz2 \) ........................................ 97

Figure 37. The custom L-frame was made of wood and its two arms were assembled using nails and metal plates. A square (left picture) was used to obtain a 90° angle. In this figure, top view (left) and bottom view (right) are reported ..................................................... 98

Figure 38. (Left) The L-frame positioned on the corner of the 2\(^{nd}\) force plate of the motion analysis laboratory. Four coloured calibration markers were attached. (Right) The arms of the L-frame were parallel to Vicon horizontal axes, and the inferior corner corresponded to the origin .................................................................................................................. 98

Figure 39. The position vector \( \textbf{dkv} \) was determined using the centroid \( C \) of the 3 aligned markers, expressed in Vicon and Kinect coordinate systems ................................................. 101

Figure 40. A generic point \( P \) tracked by Vicon and the KCMT system. The position of Vicon origin \( \textbf{dkv} \) was expressed in Kinect space ................................................................................................................................. 102

Figure 41. The description of the OpenSim coordinate system is referred to the model in its default position ........................................................................................................................................ 103

Figure 42. Left: motion data were initially expressed in Vicon coordinate system. Centre and right: two 90° rotations around the global coordinate system sequentially applied to transform motion data from Vicon to OpenSim space ........................................................................................................ 103

Figure 43. After the coordinate transformation from the Vicon to the OpenSim coordinate system, the model animated with SLS data was displayed in the correct upright position. 104

Figure 44. Four reflective markers were attached on the top surface of Kinect v2, to determine its orientation in Vicon space ...................................................................................................................... 105

Figure 45. A 3D-printed array, featuring 2 reflective markers at the extremities, and a coloured marker at their midpoint. The midpoint of the two reflective markers corresponds to the centre of the coloured marker ............................................................................................................. 105

Figure 46. P is the centre of a 3D-printed marker array, which was tracked by both systems ..................................................................................................................................................... 107

Figure 47. Lines AC and BP are perpendicular. P is the unknown intersection between the two lines. .................................................................................................................................................. 113

Figure 48. A random signal (s1) and a delayed copy of it (s2). The delay, corresponding to 50 samples in this case, was represented by the red vertical line in the second plot. Figure based on example by Mathworks (2018). .......................................................................................................................... 120

Figure 49. Cross-covariance between s1 and s2. The maximum of the cross-covariance is found in correspondence of −50, i.e. s2 must be shifted backwards by 50 samples to be aligned with s1. Figure based on example by Mathworks (2018). .......................................................................................................................... 120

Figure 50. Flowchart of the developed synchronization algorithm. Variables are reported in italics, whereas functions are reported in bold ................................................................................................................................ 123

Figure 51. Procedure for data synchronization: the 6 possible cases, determined by the amplitude of the shift and by the relative lengths of the two trials, were solved via one out of three algorithmic options .................................................................................................................................. 125

Figure 52. The larger the absolute value of t, the smaller the p-value ................................................................................................................. 135

Figure 53. Scatter plot of difference against average of the two methods, based on data from Bland & Altman (1999) ................................................................................................................................. 142

Figure 54. Within-subject standard deviation of the differences of paired measurements against within-subject mean of the measurements. The circle size is proportional to the number of observations. In Bland & Altman (2007), this plot incorrectly reports the within-subject mean of the differences on the vertical axis. .................................................................................................................. 142
Figure 55. Bland-Altman plot with bias and limits of agreement, based on data from Bland & Altman (1999). ................................................................. 144
Figure 56. Scatter plot of differences between methods against the order in which the measurements were made (points are represented by the subject number). Note that in Bland & Altman (2007), the values of points labelled 1 and 4 were incorrectly swapped along the vertical axis. ........................................................................................................ 144
Figure 57. Vicon markerset, denoted as Schache-CHESM v0.92F. Note: this image is for reference only; trials analysed in this study were collected in barefoot condition. .... 155
Figure 58. A fully marked-up participant, featuring the markerset v1.6 tracked by KCMT. The crosses indicate the body positions of the 4 virtual joint centres tracked by the Kinect v2 pose estimation algorithm and used as part of the KCMT markerset in this study. Part of Vicon reflective markerset is also visible. ...................................................................................... 158
Figure 59. Comparison between Vicon (left) and KCMT (right) markersets, attached to their respective OpenSim models. Note: markers are not to scale in the KCMT model. .... 159
Figure 60. Bland-Altman plots reporting the agreement between KCMT and Vicon for the lower limb joint angles in the sagittal plane during SLS ........................................... 163
Figure 61. Bland-Altman plots reporting the agreement between KCMT and Vicon for the lower limb joint moments in the sagittal plane during SLS ........................................... 163
Figure 62. Sagittal joint angles during SLS, obtained from KCMT (blue) and Vicon (green) and plotted against normalised time. Solid lines represent the mean of all trials, whereas the shaded areas were obtained as mean ± SD of all trials. The dots represent the minima/maxima of each averaged dataset ....................................................................................... 164
Figure 63. Sagittal joint moments during SLS, obtained from KCMT (blue) and Vicon (green) and plotted against normalised time. Solid lines represent the mean of all trials, whereas the shaded areas were obtained as mean ± SD of all trials. The dots represent the minima/maxima of each averaged dataset ....................................................................................... 164
Figure 64. A 3D-printed marker array used in the agreement study reported in Chapter 9 of this thesis. The midpoint of the two reflective markers corresponds to the centre of the coloured marker. ........................................................................................................ 167
Figure 65. Custom 3D-printed arrays, with two passive retro-reflective markers tracked by the Vicon motion capture system and a blue coloured marker tracked by Kinect v2. The centre of the coloured marker and the midpoint of the two reflective markers are coincident. .... 178
Figure 66. Markerset used for validation, including 6 retro-reflective markers tracked by the Vicon motion capture system and 3 coloured markers tracked by the KCMT system .... 180
Figure 67. Bias and limits of agreement versus treadmill speed for each marker coordinate .................................................................................................................................................. 181
Figure 68. Bias and limits of agreement versus treadmill speed for the surrogate knee flexion and adduction angles across all frames of the gait trials (a and d plots), and in correspondence of specific gait events: foot-strike (FS, b and e plots) and toe-off (TO, c and f plots) ........ 182
Figure 69. Four reflective markers were attached to the top surface of Kinect v2, to determine its orientation in Vicon space ...................................................................................... 189
Figure 70. Projection of the knee angle onto the XY plane of Kinect v2 coordinate system, denoted as surrogate knee flexion angle (θ). The hip, knee and ankle markers are indicated as H, K and A, respectively ................................................................................................. 191
Figure 71. Projection of the knee angle onto the YZ plane of Kinect v2 coordinate system, denoted as surrogate knee adduction angle (ϕ). The hip, knee and ankle markers are indicated as H, K and A, respectively ................................................................................................. 191
Figure 72. Without colours to identify each marker, other criteria must be used to reconstruct and label the 3D trajectories of each marker. (A) Marker coordinates before reconstruction. (B) Marker trajectories after reconstruction. (C) Marker trajectories after labelling. 193

Figure 73. Flowchart of the proposed labelling algorithm ................................. 195

Figure 74. A graph representing the 7 coloured markers used in the Single-leg squat study. Each node represents a marker and each line represents a rigid connection between two markers, indicating that they are attached to the same body segment (i.e. bone). It should be noted that ASIS markers are disconnected from the rest of the graph, because no rigid segment can be defined connecting the ASIS to the femur .................................................. 197
List of tables

Table 1. A comparison between the specifications of Kinect v1 and Kinect v2 ............... 29
Table 2. Results of the Bland-Altman analysis of agreement between Kinect- and Vicon-derived joint angles. Since the LOA were regression-based, they were not constant across the range of motion (horizontal axes in Figure 17). Therefore, the LOA could not be described by individual numbers. To provide a better representation of these linearly variable LOA, their values measured in correspondence of the minimum and maximum degree of flexion for each joint were reported in this table. ........................................................................................................... 53
Table 3. Reviewed papers and metrological categories discussed in each of them ............... 59
Table 4. Summary of the most significant results from different studies on the uniformity of accuracy across the Kinect v2 depth sensor. In this table, when referring to the centre of the depth sensor, a small number of pixels located at the borders and corners were excluded. Results from Steward et al. (2015) were not included in this table, because the distance Kinect-target was not specified in that study. .......................................................................................................................... 66
Table 5. Summary of the most significant results from different studies on the uniformity of precision across the Kinect v2 depth sensor. In this table, when referring to the centre of the depth sensor, a small number of pixels located at the borders and corners were excluded. 68
Table 6. Summary of the results reported in the reviewed studies about the effect of material types and colours on the accuracy of Kinect v2 depth measurements. Abbreviations: NS = not specified, BG = background, BK = black, W = white, Y = yellow, R = red, B = blue, G = green. 69
Table 7. Summary of the sources of error affecting Kinect v2 depth measurements, their impact on accuracy and precision, and possible workarounds to minimize their effect....... 73
Table 8. Theoretical and actual x, y and z coordinates of the coloured calibration markers expressed in the L-frame reference system. Actual coordinates were measured after attaching markers on the L-frame. Coordinates in bold are different from their theoretical values..... 99
Table 9. Shifts calculated using CCR and CCV from treadmill gait data. CCV provided more accurate and precise lag estimation across all pairs of signals. Starred values indicate the best estimate of the shift. All shifts are expressed in number of samples. ............................................ 121
Table 10. Vicon Schache-CHESM v0.92F markerset was composed of 46 reflective markers, 8 of which (indicated by a *) were only used in static trials ................................................................. 157
Table 11. KCMT markerset v1.6 included 4 virtual joints centres tracked by Kinect v2 pose estimation algorithm and 7 coloured markers................................................................. 158
Table 12. Joint coordinates of the OpenSim Gait2392 model, customized for use with Vicon data. All coordinates were clamped and the corresponding ROMs were reduced in almost all cases, to avoid incongruent solutions to the IK problem............................................. 160
Table 13. Joint coordinates of the OpenSim Gait2392 model, customized for use with KCMT data. All coordinates were clamped and the corresponding ROMs were reduced in almost all cases, to avoid incongruent solutions to the IK problem.............................................. 161
Table 14. Results of the Bland-Altman analysis of the joint flexion angles, observed in two previous studies using Kinect v2 markerless pose estimation algorithm (McGroarty et al. 2016; Mentiplay et al. 2015) and in the present study using KCMT (shaded rows). A Vicon system was used as gold standard in all cases. ................................................................. 165
Table 15. Measurements used to scale the OpenSim Gait2392 model using Vicon data. The distances between pairs of markers from the Schache-CHESM v0.92F markerset were used to compute scale factors for the specified body segments along the three axes................. 170
Table 16. Marker weights used to scale the OpenSim Gait2392 model using Vicon data. Higher weights were given to markers placed on bony landmarks. Weights are relative, for example a weight of 1 compared to 10 is the same as 100 compared to 1000. .......................... 170
Table 17. Measurements used to scale the OpenSim Gait2392 model using KCMT data. The distances between pairs of markers from the KCMT v1.6 markerset were used to compute scale factors for the specified body segments along the selected axes. ............................. 171
Table 18. Marker weights used to scale the OpenSim Gait2392 model using KCMT data. Higher weights were given to markers placed on bony landmarks. Weights are relative, for example a weight of 1 compared to 10 is the same as 100 compared to 1000. .......................... 171
Table 19. Marker weights used for the IK analysis of Vicon data. Markers used in static trials only were identified by an * and were disabled in this analysis. .............................. 172
Table 20. Marker weights used for the IK analysis of KCMT data .............................. 172
Table 21. Results of the Bland-Altman analyses of agreement, reported as biases, limits of agreement (LOA) and margins of error (MOE, intended as half the width of the 95% confidence intervals of the limits of agreement)................................................................. 183
Table 22. Pairs of markers attached to the same bone were defined as rigid segments, because their distance was constant ................................................................. 196
Table 23. The results of the Treadmill Study presented in Chapter 9 of this thesis indicate that the KCMT system is more accurate than Kinect v2 pose estimation algorithm, and that it may be sufficiently accurate to identify subjects at risk of ACL injury, based on comparison with data from Yamazaki et al. (2009). ................................................................. 199
Abbreviations and acronyms

ACL: anterior cruciate ligament.
API: application programming interface.
ASIS: anterior superior iliac spine.
BG: background.
CHESM: Centre for Health, Exercise & Sports Medicine at The University of Melbourne.
DVJ: drop vertical jump.
FOV: field of view.
GC: garbage collection, the automatic memory management routine in .NET framework.
GRF: ground reaction forces.
GUI: graphical user interface.
HSV: hue, saturation and value, a colour model.
ICC: intraclass correlation coefficient.
ID: inverse dynamics.
IK: inverse kinematics.
IR: infrared.
IRROI: infrared region of interest.
KCMT: Kinect coloured marker tracking.
LOA: limits of agreement.
MTP: metatarsophalangeal (joint).
NS: not specified.
RANSAC: random sample consensus.
RGB: red, green and blue, a colour model.
RMSD: root-mean-square deviation.
RMSE: root-mean-square error.
ROI: region of interest.
ROM: range of motion.
SD: standard deviation.
SDA: standard deviation average.
SDK: software development kit.
SLS: single-leg squat.
TOF: time of flight, the depth sensing technology used in Kinect v2.
v2: version 2.
UInt8: unsigned 8-bit integer.

Units of measurement

°: degree (angle).
°C: degree (Celsius).
cm: centimetre.
fps: frames per second.
h: hour.
Hz: hertz.
m: metre.
min: minute.
Symbols

\( M \) = a matrix. Sometimes reported with its sizes, e.g. \( M_{m \times n} \), where \( m \) is the number of rows, and \( n \) the number of columns.

\( v \) = a vector in \( \mathbb{R}^3 \).

\( P \) = a point in \( \mathbb{R}^3 \), which is also a vector in \( \mathbb{R}^3 \).

Glossary

**Coordinate system**: a 3D reference frame composed of 3 orthogonal axes and an origin. Also denoted as *space* or *3D workspace*.

**Pose**: position and orientation of a coordinate system or a rigid body in 3D space.

**Calibration**: determination of the pose of a coordinate system with respect to another coordinate system. Also denoted as *registration* or *extrinsic calibration*.

**Coordinate transformation**: conversion of a set of measurements from one reference frame to another. It is a combination of a rotation and a translation, and is also denoted as *rigid transformation*.
CHAPTER 1
Introduction

1.1 ACL injury risk in young girls

The anterior cruciate ligament (ACL) connects the anterior intercondylar area of the tibia (posterior to the attachment of the medial meniscus) to the posteromedial side of the lateral condyle of the femur. The ACL is the primary restraint to anterior translation of the tibia with respect to the femur (Butler et al. 1980; Fukubayashi et al. 1982). Furthermore, it is a secondary restraint to internal and valgus tibia rotations (Zantop et al. 2007; Masouros et al. 2010).

ACL injury continues to be the largest single problem in orthopaedic sports medicine, with the incidence of non-contact ACL tears being much higher in female than male athletes in sports such as basketball and handball (Renstrom et al. 2008). In the past few decades, the incidence of ACL injuries in girls has risen with their increased participation in sports (Loudon et al. 1996). Compared to boys, girls incur more than twice ACL ruptures from the onset of puberty (11-12 years) (Arendt & Dick 1995; Arendt et al. 1999; Boden et al. 2010; Shea et al. 2004; Renstrom et al. 2008). Data from the Norwegian National Knee Ligament Registry indicate an incidence of 34 primary ACL reconstruction per 100,000 citizens, with females having the most ACL reconstructions in the 15-19-year age group (Figure 2). The majority of ACL injuries (approximately 70%) are caused by a non-contact mechanism (McNair et al. 1990; Boden et al. 2000).

Sports such as soccer, basketball, ski, volleyball, netball and gymnastic have the highest ACL injury risk because they require quick decelerations in movements such as landing, cutting, changing direction and pivoting (Boden et al. 2000; Griffin et al. 2006; Renstrom et al. 2008; Kobayashi et al. 2010; Otsuki et al. 2014). Data from the National Collegiate Athletics Association (NCAA) indicated that, in the pre-collegiate age group (14-18 years), the rate of non-contact ACL injuries in basketball was the highest, i.e. 4 times that of males. It decreased
to 3.6 at the college level, and approximated 1 at professional level, suggesting that the rate of ACL injuries decreases as female athletes mature and the level of play increases (Renstrom et al. 2008).

Figure 2. Primary ACL reconstructions by age and sex from the Norwegian National Knee Ligament Registry. Data from Renstrom et al. (2008).

ACL injuries during puberty are problematic for several reasons. First, the risk of developing knee osteoarthritis is significantly increased after an ACL injury (Griffin et al. 2006; Lohmander et al. 2007; Myklebust, Holm, et al. 2003; Caine & Golightly 2011), irrespective of the treatment (surgical versus conservative) provided to the patients (von Porat et al. 2004). Second, reconstructive surgery of the ACL in skeletally immature patients might affect regular growth. Third, reconstructive surgery and rehabilitation prevent participation in sports activities for a long period, impacting on healthy physical development. Furthermore, besides of age, ACL reconstruction is a costly procedure, which could be avoided with prevention based on monitoring of high-risk subjects and implementation of evidence-based training protocols (Myklebust, Engebretsen, et al. 2003). To obtain maximum prophylactic effects, such neuromuscular training programs must incorporate a variety of exercises, including balance, plyometrics, strengthening and proximal control training (Sugimoto et al. 2015).

Multiple factors have been attributed to the increased risk of ACL rupture in girls (Griffin et al. 2006; Wild et al. 2012). Such factors can be divided in two broad categories, i.e. external and internal risk factors. External factors include type of competition, footwear, shoe-surface interaction, and environmental conditions. Little is known about the influence of external factors: for example, there is no evidence that knee braces prevent ACL injuries, and there is conflicting evidence on the effect of increased shoe-surface friction. Therefore, this area merits further investigation (Griffin et al. 2006; Renstrom et al. 2008).

Internal factors are defined as follows: i) hormonal, e.g. changes in estrogen levels; ii) anatomical, i.e. changes in musculoskeletal structure, including reduced flexibility, lack of muscle strength spurt, hamstring strength development lagging behind that of quadriceps, and knee laxity; iii) biomechanical, i.e. changes in landing technique due to bad lower limb alignment and increased knee valgus. In the following sections, each internal ACL injury risk factor affecting girls during puberty will be illustrated.
1.2 Internal risk factors for ACL injury in girls

1.2.1 Stage of development

Detecting the stage of development in a young individual is fundamental, because it deeply affects lower limb biomechanics in relation to ACL injury risk (Hewett et al. 2006; Ford et al. 2003; Ford et al. 2010; Susan M. Sigward et al. 2012; S M Sigward et al. 2012). Since the onset and progression of puberty are not strictly related to age, the Tanner scale (Tanner et al. 1976; Tanner 1986) has been uniformly adopted as a method to classify adolescents of both genders according to their pubertal development stage. In this 5-stages scale, girls are rated for breast development and pubic hair growth.

1.2.2 Hormonal changes

Hormone receptors (e.g. estrogen, testosterone and relaxin) have been found in the human ACL, suggesting they may influence the biology of this ligament. There is general consensus that the likelihood of incurring an ACL injury is significantly greater during the preovulatory phase than during the postovulatory phase of the menstrual cycle (Renstrom et al. 2008; Shultz et al. 2015). However, further work is required to understand the underlying mechanism driving this increased likelihood.

In girls, increased production of estrogen at the onset of puberty has been observed, with only minor changes in levels of testosterone. Concentration of estrogen rises continuously until the beginning of adulthood (Figure 3). In contrast, testosterone concentration in boys increases with the onset of puberty, with only little changes in estrogen. Hormonal differences between genders may affect the risk of ACL injury. Given its higher concentration in females compared to males, estrogen may affect the mechanical properties of the human ACL, increasing the injury risk in females at the time of pubertal estrogen influx (Wild et al. 2012).

![Figure 3. Testosterone and estrogen hormonal concentrations across all Tanner stages in boys and girls. Data from Wild et al. (2012).](image-url)
Interestingly, a peak can be observed both in the number of primary ACL reconstructions (Figure 2) and in the estrogen level (Figure 3) in females before adulthood. The peak in the former graph is reported between 15 and 19 years of age, whereas the peak in the latter graph is observed in correspondence of Tanner stage V. Because Tanner stage V in females corresponds approximately to an age range between 12.5 and 18 years (WHO 2010), the two peaks are approximately aligned along the age scale, supporting the hypothesis that the estrogen influx in girls may increase their risk of injury by affecting the mechanical properties of their ACL.

Structural and mechanical integrity of ACL and other ligaments is compromised in a high estrogen environment (Slauterbeck et al. 1999; Liu et al. 1997; Yu et al. 1999; Woodhouse et al. 2007). Fibroblasts repair micro damage in ligaments, producing collagen type I (responsible for mechanical properties including stiffness and strength) and collagen type III (responsible for elasticity). Normal mechanical loading of the ACL and other ligaments (e.g. during walking) is essential in maintaining their integrity and homeostasis, because it increases type I collagen and consequently strength. However, increased estrogen has been associated with a reduction of type I and III collagen production (Lee, Liu, et al. 2004; Lee, Smith, et al. 2004), which counteracts the positive effect of mechanical loading during growth.

In recent years, the hormone relaxin has gained attention because it has been found in high concentrations in female athletes with ACL injuries compared with uninjured controls. Studies have indicated the capacity of this hormone to affect soft tissue remodelling of the ligaments, generating less organized and dense collagen structure, which may manifest as a weaker and laxer ACL. However, the underlying mechanism regulating the effect of sex hormones on ACL mechanical properties remain an important area of study (Shultz et al. 2015).

1.2.3 Musculoskeletal changes

Knee joint laxity
ACL has a primary role in knee stability. Higher knee laxity means higher ACL injury risk (Myer et al. 2008). The steep rise in estrogen levels in girls during puberty may affect ligament mechanical properties and, in turn, knee joint laxity and ACL injury risk. Pubescent females have higher knee hyperextension (genu recurvatum) and generalized joints laxity than males (Quatman et al. 2008). However, specifically how the substantial influx of estrogen in girls during puberty affects knee laxity is not well understood and requires further investigation (Wild et al. 2012).

Flexibility
Flexibility is defined as the extensibility of periarticular tissues to allow physiological motion of a joint or limb (Alter 2004). Peak height velocity (PHV) is a somatic biological maturity indicator and reflects the maximum velocity in standing height growth during adolescence. In correspondence of PHV the skeleton grows faster than skeletal muscles (Wild et al. 2012). This may reduce flexibility.

There is no consensus in literature regarding the role of knee flexion angle in ACL injuries. Koga et al. (2010) observed reduced knee flexion angles in a video analysis of 10 injury situations. Krosshaug et al. (2007) reported higher flexion angles in females compared to males in video analysis of 39 cases of ACL injury situations. Hewett et al. (2005) suggested that knee flexion angle at landing is not a predictor of ACL injury risk.
Muscular development

During landing, muscles controlling the knee have a primary role in protecting the ACL (Wild et al. 2012). There are gender differences in the velocity of strength increases in the two years following PHV. Due to testosterone, the increase in quadriceps strength is greater in boys and merely proportional to height and weight in girls (Round et al. 1999). Hamstring muscles display a similar trend in boys, while in females they don’t display a significant increase in strength (Costello et al. 2011). Moreover, the muscle strength ratio hamstring/quadriceps is significantly lower in girls, potentially causing over reliance of the quadriceps and underutilization of hamstring (Hewett et al. 2004; Ahmad et al. 2006; Barber-Westin et al. 2006). It should be noted that anterior-directed shear force applied to the tibia, such as that caused by contraction of the quadriceps muscles when the knee is near extension, is the primary cause of strain on the ACL (Renstrom et al. 2008). This muscular imbalance might lead to less knee stability and, thus, to a lower protection for ACL during dynamic landing, which in turn translates into a higher ACL injury risk.

![Figure 4. Decreased ratio of hamstring-to-quadriceps strength generates anterior-directed shear force on the tibia (red arrow), which is the primary cause of strain on the ACL (red dashed line) when the knee is near full extension](image)

1.2.4 Landing biomechanics

Poor landing technique during sports activities is a common cause of non-contact ACL injury, particularly in pubescent girls. It is characterized by: i) high peak vertical ground reaction forces (vGRF) during box drop landing; ii) altered neuromuscular coordination; iii) increased knee joint valgus. It should be noted that the valgus observed in the coronal plane does not entirely derive from rotation of the tibia around a stationary femur in the coronal plane, but is a combination of hip adduction, hip internal rotation, knee flexion and knee abduction, and is defined as dynamic knee valgus (Renstrom et al. 2008; Tamura et al. 2017).

No differences have been found in normalized (by body weight) peak vGRF during the double-leg drop vertical jump manoeuvre between boys and girls (Hewett et al. 2006) and across puberty (Quatman et al. 2006). Girls showed higher loading rates during Tanner stage II and III compared with IV and V, due to different muscular activation patterns, and overall compared with boys (Pappas et al. 2007; Kernozek et al. 2008). However, these studies only
investigated the vertical component of the ground reaction forces (GRF), while horizontal components may have a significant contribution to risk of ACL injury (Wild et al. 2012).

Females perform deceleration of the body during landing, cutting and pivoting using different neuromuscular patterns compared to males (Griffin et al. 2006) and demonstrate less hip and knee flexion, higher knee valgus, higher internal hip rotation, higher external tibia rotation and higher quadriceps contraction compared to hamstring contraction.

Change in lower limb and particularly knee alignment occurs in girls (Tanner stage IV and V), but not in boys during puberty (Hewett et al. 2004; Ford et al. 2010). During box-drop landing, there is an increase of peak knee valgus (30° vs 20°) and of total valgus knee motion in girls compared to boys (Ford et al. 2003). Dynamic knee abduction angle and abduction moment highly correlate with occurrence of ACL injury (Hewett et al. 2005).

Voluntary modification of the knee flexion angle during block jump landing was found to have patient specific effects on injury risk factors. Participants were asked to modify their technique, landing softly and increasing their flexion angle during landing. This change caused a reduction in vGRF and maximum flexion moment. Furthermore, the peak values for almost all parameters of interest were significantly delayed. Since ACL injuries typically occur during the beginning of a stance, these delays may lower the risk of ACL injuries. However, abduction angle and abduction moment were reduced only in those participants who initially landed in an adducted position and had no significant effect on the group that initially landed in an abducted position. Because these are two critical risk factors, modifying the knee flexion angle alone may not be sufficient to reduce the risk of injuries for all patients (Favre et al. 2016).

Single-limb drop landing is a task which considerably challenges the ACL (Boden et al. 2000; Griffin et al. 2000). Compared to boys, girls demonstrated increased quadriceps activation and lower hamstring/quadriceps activation ratio before foot contact, combined with higher internal tibial rotation (Nagano et al. 2007). Before puberty, both males and females demonstrate valgus knee position during single limb landing (Buchanan 2003). However, starting from the onset of puberty and beyond, males mainly land in a varus knee position, whereas females continue demonstrating valgus knee. The combination of knee valgus moment and internal tibia rotation moment has been found to increase the ACL strain notably more than either load applied individually (Shin et al. 2011).

Single-leg squat (SLS) does not challenge the ACL as much as landing and deceleration tasks do. However, SLS has been successfully used as a diagnostic test to gain an indication of lower limb alignment and control in females versus males (Zeller et al. 2003). Girls showed an initial loss of knee control, represented by a combination of valgus position, higher hip adduction and foot pronation. Therefore, a decline in lower limb alignment during SLS can be a good indicator of an even less controlled landing or deceleration movement. Willson et al. asked 24 male and 22 female athletes to perform SLS and measured the frontal plane projection angle (FPPA) of the knee at 45° flexion (Willson et al. 2006). They found that females and males moved towards opposite directions: females typically tended to increase their FPPA, whereas males tended to move towards more neutral knee alignment. The authors suggested that the SLS may be used as a test to screen for individuals whose knee mechanics may stress the ACL. In a study about the differences in kinematics of SLS between ACL-injured patients and healthy controls (Yamazaki et al. 2009), the uninjured leg of female ACL-injured participants demonstrated significantly more knee flexion (7.7° on average) and external hip rotation (7.4°), and less hip flexion (~18.1°) compared to female controls. An increased knee valgus
was also observed ($5.2^\circ$), however its $P$ value was slightly above the level of significance ($P$ value $= 0.08$, $\alpha = 0.05$). Furthermore, female participants displayed significantly more external hip rotation and knee valgus in both the injured and uninjured legs compared to male participants. These gender differences in joint kinematics indicate that increased dynamic knee valgus contributes to ACL non-contact injury risk in women. The authors suggested that, compared to single-leg landing, SLS could be used as a simpler and safer clinical examination for large cohorts of subjects.

1.2.5 Pronation

Overpronation has been linked many times to increased ACL injury risk. Copland demonstrated that passive tibial rotation (rotational knee laxity) is greater in people with overpronation compared to control subjects at $5^\circ$ of knee flexion (Coplan 1989). ACL injured subjects have greater navicular drop test scores than non-injured subjects (Beckett et al. 1992). Because of the anatomical function of the ACL, prolonged pronation of the foot and ankle complex produces excessive internal tibia rotation, and thus may produce a preloading effect on the ACL. This preloading concept may serve as a partial explanation for the high percentage of injuries to the ACL during non-contact sport activities.

In agreement with Beckett et al., Woodford-Rogers et al. reported greater navicular drop test scores and anterior knee joint laxity in ACL injured subjects compared to non-injured subjects matched for sport, team, position and skill level. They also suggested that pronation can be limited by selecting appropriate footwear (Woodford-Rogers et al. 1994).

In line with the aforementioned studies, Loudon et al. found a strong relationship between a standing posture of genu recurvatum with subtalar joint overpronation and prevalence of non-contact ACL injuries in female athletes (Loudon et al. 1996). The authors suggested that these individuals' proprioceptive system is calibrated on a faulty position. Therefore, when they perform dynamic actions involving stresses beyond normal, the ACL may be at greater risk.

1.3 Assessment of ACL injury risk

Understanding the mechanisms of ACL injury is fundamental to develop specific methods for preventing sports injuries. Different methodologies have been used to study such mechanisms: video analysis of actual injuries, in vivo studies measuring ligament strain via surgically implanted strain gages, cadaver studies, and measurement and computational modelling of ACL challenging tasks (Renstrom et al. 2008).

Video analysis is the only approach which provides kinematic information from the actual event. Studies of this kind agree in reporting that injuries occurred during cutting or landing situations, with the knee relatively straight and often in valgus position. However, analysis is generally based on visual inspection, providing results with poor accuracy and precision. In vivo studies provide accurate, quantitative strain data captured during ACL-challenging tasks. However, they require surgical introduction of a DVRT (differential variable reluctance transducers) on the subjects’ ACL (Fleming et al. 2003; Cerulli et al. 2003), a practice with strong ethical implications. Ex vivo studies are valuable surrogates of in vivo studies. Since they are typically based on the use of cadaveric knee samples loaded into a testing machine, they can provide accurate quantitative results. However, a limitation of these studies is that the knee kinematics and muscle forces applied to the sample do not replicate the full time history profiles of ACL-challenging activities such as landing (Bakker et al. 2016).
Three-dimensional motion analysis studies are the gold standard for assessing lower limb kinematics and kinetics, because they provide accurate quantitative insights into the multi-planar biomechanics of joints during ACL-challenging tasks, without the ethical implications of in vivo studies.

Different technologies are available for tracking human motion. Bi-plane X-ray fluoroscopy has the capacity to accurately and non-invasively measure human joint motion in vivo during dynamic weight-bearing activities (Ackland et al. 2011). Accuracy of bi-plane fluoroscopy for the knee has been reported at 0.24 mm and 0.16° for translations and rotations, respectively. However, fluoroscopy systems have several limitations that currently prevent their wide adoption in motion analysis studies. First, they must be operated in an isolated radiation-complaint facility by a trained medical technician. Second, most off-the-shelf C-arm fluoroscopy units have a sampling frequency of 25 fps, which prevents their application with high-speed tasks. Third, their restricted imaging capture volume (typically up to 30 x 30 x 30 cm) imposes limitations on the number of joints and the types of dynamic tasks that can be captured. For example, capture of knee joint motion during walking is generally restricted to the stance phase of treadmill gait. Fourth, they expose the participant to a low radiation dose (typically 80 μSv for a 20 s measurement). In its present form, X-ray fluoroscopy technology is unlikely to be adapted to the imaging of multiple joints or whole-body joint motion simultaneously, as this would potentially expose the patient and operator to excessive radiation.

Marker-based optical motion capture systems have been employed widely in clinical motion analysis. They are typically based on a set of high-speed infrared cameras tracking multiple markers, which can be either passive (retro-reflective) or active (infrared LEDs). These systems have several advantages compared to fluoroscopy, such as larger capture volumes (ideal for full body motion capture), higher capture framerates (suitable for high speed tasks) and no radiation exposure. In a study using a 25 mm marker attached to a measurement robot, a Vicon-460 (Oxford, UK) motion capture system provided a single-marker accuracy of 63±5 μm (Windolf et al. 2008). However, these systems are not always practical in the clinical setting, given the substantial setup costs and space requirements (Whittle 2007). A high-end 12-camera Vicon motion capture system costs approximately AU$ 250,000 (Thewlis et al. 2013). Furthermore, marker tracking error and skin-motion artefact can lead to poor approximations of the kinematics of the underlying bones (Ackland et al. 2011). To overcome these limitations, markers have been mounted on pins and inserted into the underlying bones, but the invasive nature of this alternative approach has limited its application in human movement studies.

Wearable inertial sensors can be less expensive and cumbersome because they don’t require a set of high speed infrared cameras, but are subject to position measurement errors as a consequence of acceleration integration (Muro-de-la-Herran et al. 2014). Markerless tracking has recently gained popularity in the biomechanics community, especially after the introduction of a low-cost and portable device denoted as Kinect (Microsoft, Redmond, US). In the following section, markerless motion capture is described in further detail.

1.3.1 Markerless technology

In 2010, Microsoft introduced a low-cost and portable gaming device denoted as Kinect for Xbox 360 or Kinect for Windows v1 (Microsoft, Redmond, US, Figure 5). Kinect provides full-body 3D motion capture without need for skin markers, thanks to an in-built pose estimation algorithm. This device features an RGB camera, a depth camera and an infrared (IR) projector.
The first generation of Kinect uses *structured light* for depth sensing, a technology licensed by the Israeli company PrimeSense and not fully disclosed (Zhang 2012). The IR projector emits a set of IR dots. Because the pose of the IR projector with respect to the IR camera as well as the pattern of IR dots are known, each dot can be reconstructed in 3D using triangulation. This process results in a depth map of the scene observed by the Kinect depth sensor. Based on this depth map, the pose estimation algorithm included in the Kinect System Development Toolkit (SDK) is able to track 20 joint centres of up to 2 users positioned in front of Kinect (Girshick et al. 2011; Shotton et al. 2011; Shotton et al. 2012; Taylor et al. 2012; Kohli & Shotton 2012).

With a cost of AU$ 200, Kinect is far cheaper compared to conventional motion capture systems, is portable, and doesn’t require markers or wearable sensors to be attached on the participant’s skin. However, it is also less accurate and, being a single-camera system, more prone to occlusions.

![Image of Microsoft Kinect v1](image1)

*Figure 5. Microsoft Kinect v1, also denoted as Kinect for Xbox 360*

![Image of Kinect v1 disassembled](image2)

*Figure 6. The Kinect v1 disassembled. Behind the three lenses, there are from left to right: the infrared projector, the RGB camera and the infrared depth camera. Photo by Walter Galan ([IFIXIT](#), licensed under [CC BY-NC-SA 3.0](#)).
Before the release of Kinect by Microsoft, other groups have worked on markerless human motion capture. Corazza et al. (2009) developed a markerless motion capture system based on multiple colour cameras and an automatically-generated subject-specific anatomical model. However, this system required a minimum of 8 cameras, making it expensive and cumbersome compared to Kinect. Furthermore, it was not made commercially available. These two reasons made this solution unsuitable for the purposes discussed in this thesis. Similarly, a multi-camera markerless motion capture system denoted as OpenStage 2.0 was commercially released by a company named Organic Motion, however this system was very costly and has been recently discontinued.

Several studies have assessed the validity and reliability of the first generation of Kinect as a low-cost motion tracking device in clinical functional analysis and rehabilitation. In postural control, Kinect showed comparable reliability and excellent concurrent validity to a high-end Vicon system (Clark et al. 2012); however, proportional biases for some measures from Kinect were reported. In gait analysis, some spatio-temporal parameters were found to be more accurately measured than others, due to hardware and software limitations of Kinect (Clark, Bower, et al. 2013). In assessing the validity of Kinect in providing lateral trunk lean feedback for gait retraining (Clark, Pua, et al. 2013), it was observed that a global calibration significantly improved raw results data from a mean angular error of $3.2 \pm 2.2^\circ$ to $1.7 \pm 1.5^\circ$, and a more complex (individualized) calibration further reduced the error to $0.8 \pm 0.8^\circ$. Used in functional analysis, Kinect and optical systems provided similar reproducibility, but the measured ranges of motion were found to be different (Bonnechère et al. 2014).

In July 2014 Microsoft publicly released a second generation of this device, denoted as Kinect for Windows v2 or Kinect for Xbox One (Figure 7) and costing approximately AU$ 200. This newer version featured a different depth sensor based on time-of-flight (TOF) technology (Li 2014) and more capable pose estimation, increasing the number of tracked joints from 20 to 25. The change of depth sensing technology was accompanied by substantial upgrades in terms of resolution, field of view and noise reduction of the depth camera. The RGB camera resolution was increased as well (Table 1).

<table>
<thead>
<tr>
<th></th>
<th>Kinect v1</th>
<th>Kinect v2</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Depth sensor resolution (px)</strong></td>
<td>320 x 240</td>
<td>512 x 424</td>
</tr>
<tr>
<td><strong>RGB sensor resolution (px)</strong></td>
<td>640 x 480</td>
<td>1920 x 1080</td>
</tr>
<tr>
<td><strong>Framerate (fps)</strong></td>
<td>30</td>
<td>30</td>
</tr>
<tr>
<td><strong>Field of view (H x V)</strong></td>
<td>57° x 43°</td>
<td>70° x 60°</td>
</tr>
<tr>
<td><strong>Skeleton joints tracked</strong></td>
<td>20</td>
<td>25</td>
</tr>
<tr>
<td><strong>Full skeletons tracked</strong></td>
<td>2</td>
<td>6</td>
</tr>
<tr>
<td><strong>USB standard</strong></td>
<td>2.0</td>
<td>3.0</td>
</tr>
</tbody>
</table>

Due to its affordability, portability and capability to track human motion without the need for skin markers, Microsoft Kinect v2 was identified as a possible motion capture device to be used in the present thesis. Being a novelty, data on the accuracy of the markerless pose estimation algorithm of Kinect v2 were not available in literature until the following year (Mentiplay et al. 2015; Xu & McGorry 2015; Wang et al. 2015). Such results were reported in Chapter 2 of this thesis and were compared with those from a preliminary study carried out.
by the author of the present thesis. In this preliminary study, trajectories of virtual joint centres estimated using Kinect v2 were used to calculate lower limb joint angles via inverse kinematics. Based on the improved hardware and software specifications of Kinect v2, it was hypothesized that this device would provide increased tracking accuracy compared to Kinect v1, and that Kinect v2-derived joint angles would be accurate enough to discern between subjects at risk of ACL injury and controls.

Figure 7. Kinect v2 teardown. The RGB and depth camera are visible (first and second from left respectively), as well as the infrared LEDs (centre). Photo by Walter Galan (IFIXIT / CC BY-NC-SA 3.0).

In the present thesis, inverse kinematics analyses (required to calculate joint angles) were carried out using OpenSim (Stanford US), a widely adopted musculoskeletal modelling software application. In the following section, an overview of OpenSim and its features is presented, together with a typical musculoskeletal workflow.

1.3.2 Musculoskeletal modelling
The *in vivo* assessment of muscle function would require unpractical, invasive and sometimes even unethical practices. For this reason, musculoskeletal computational modelling has become the gold standard methodology for obtaining quantitative data such as muscle-tendon lengths, muscle moment arms, muscle forces, muscle activation patterns and ligament loads from motion data in a non-invasive way.

Introduced in 2007, OpenSim (Stanford, US) has become a widely accepted standard software package for musculoskeletal analyses (Delp et al. 2007). It is an open-source application which enables researchers to build, exchange and analyse computational models of the musculoskeletal system and dynamic simulations of movement. Models are based on literature, imaging and direct measurements on cadavers (Delp et al. 1990; Yamaguchi & Zajac 1989; Anderson & Pandy 1999; Anderson & Pandy 2001). Simulations results are usually validated by comparison with previous studies featuring *in vivo* or *in silico* experiments.

OpenSim has been largely adopted as essential tool in the estimation of kinematics and kinetics of the knee during landing tasks. Google Scholar returned 2500 results to the query “OpenSim+knee” as of September 2018. Laughlin et al. (2011) compared *soft* versus *stiff* drop landings, finding that simple verbal instructions can improve lower limb alignment that decreases the load on the ACL. Morgan et al. (2014) investigated the relative forces produced by muscles spanning the knee during single-limb landing. They found that both the quadriceps and gastrocnemii muscle force estimates were significantly greater than those generated by
the hamstrings. High gastrocnemius forces corresponded with increased joint compression and lower ACL forces. The authors concluded that elevated quadriceps and gastrocnemius forces during landing may represent a generalized muscle strategy to increase knee joint stiffness, protecting the knee and ACL from injury risk, and suggested that this finding may be used as the foundation for novel muscle-targeted training intervention programs aimed to reduce ACL injuries in sport.

Thompson et al. (2017) analysed motion data of 51 preadolescent female soccer athletes using OpenSim, to assess the effects of an injury prevention program on biomechanical risk factors for an ACL injury. They found that peak knee valgus moment during double leg squat was reduced in the intervention group compared to the control group (−0.57 vs 0.25 %body weight × height, respectively). Tran et al. (2016) collected motion data of 10 male and 10 female participants during double-leg jump landing and analysed them using OpenSim. They found that a 30° toe-in foot position during landing exacerbates biomechanical risk factors associated with ACL injury, whereas a 30° toe-out landing position decreases these factors, compared to a 0° foot position.

Bakker et al. (2016) proposed a combined in-vivo/computational/in-vivo approach to measure ACL strain due to sagittal plane muscle forces during dynamic activities. The authors collected marker trajectories and ground reaction forces of 7 recreational athletes performing single-leg jump landing. Then they analysed these data in OpenSim, to calculate joint angles and muscle forces. These results were used to drive an in-vitro model, composed of an instrumented cadaveric knee mounted on a computer-controlled loading device featuring six electromechanical actuators. This machine was designed to move the knee in the sagittal plane and to apply dynamic muscle forces at the insertion sites of the quadriceps, hamstring, and gastrocnemius muscle groups and the net moment at the hip joint. The findings of this study were used to develop a regression model to estimate the peak ACL strain from sagittal plane kinematics and anatomical parameters, and suggested that landing with more erect posture (decreased hip and trunk flexion) increases the strain in the ACL, a conclusion in agreement with previous studies. The regression formula may be used as a screening tool to identify individuals at risk of ACL injury, and to assess the effects of training programs on estimates of ACL strain.

In a typical workflow based on OpenSim (Figure 8) motion data are initially cleaned, labelled and then converted from the motion capture format (e.g. .c3d) to OpenSim-compatible format (e.g. .trc and .mot). This conversion can be performed using one of the third-party software toolboxes reported on the OpenSim website, such as MATLAB OpenSim Tools v2 (Lichtwark et al. 2014). This operation requires a substantial degree of customization by the user, based on the specific data collection protocol (i.e. laboratory coordinate system, use of force plates, selected markerset, etc.) and on the aim of the study that, for example, may influence the choice of the data filtering technique.

After this pre-processing phase, data are passed to OpenSim and the following steps are performed. First, the musculoskeletal model is scaled to match the participant’s anthropometry. Using generic musculoskeletal models based on measurements from cadavers of adult donors is a standard practice in musculoskeletal modelling. Second, inverse kinematics analysis (IK) is performed to calculate joint angles from marker positions. Third, inverse dynamics analysis (ID) is used to calculate joint moments, using joint angles and external forces as input. Fourth, static optimization (SO) algorithm is executed to resolve the net joint moments into individual muscle forces, minimizing the sum of squared (or other
power) muscle activation. Because the standard OpenSim musculoskeletal models do not include ligaments, an additional modelling procedure would be required to calculate the force experienced by the ACL. However, this procedure is beyond the scope of the present thesis and will not be discussed.
Figure 8. Chart illustrating a possible OpenSim workflow. Extensions were used to indicate the following file types: C3D (Coordinate 3D files), TRC (Track Row Column files, also known as marker files), GRF (Ground Reaction Forces files), OSIM (OpenSim Model files), MOT (Motion files), STO (Storage files), XML (Extensible Markup Language files).
1.4 Aims and structure of this thesis

Although the assessment of 3D lower limb biomechanics is fundamental to identify and monitor subjects at risk of ACL injury, this kind of analysis is currently limited to selected facilities which can afford to buy and maintain the expensive and cumbersome motion capture equipment required for these studies. For example, a high-end 12-camera Vicon system costs approximately AU$ 250,000 (Thewlis et al. 2013) and requires a dedicated room, which must be large enough to include the capture volume (whose size may vary depending on the tasks to be analysed) and the surrounding tripods holding each camera in position. In contrast, to timely screen large cohorts of young individuals, motion analyses should be more accessible, so that they could be performed in schools, gyms and clinical practices, where large groups of adolescents could more easily participate.

Initially, the aim of this thesis was to understand if lower limb kinematics measured using the markerless pose estimation algorithm provided by Microsoft Kinect v2 was sufficiently accurate to discern subjects at risk of ACL injury from controls. In this case, the next step would have been to carry out an agreement study between Kinect v2 pose estimation algorithm and a marker-based optical motion capture system, measuring lower limb kinematics on a cohort of young girls performing ACL-challenging tasks. However, results from a preliminary analysis of agreement between the two systems indicated that the off-the-shelf Kinect v2 was not sufficiently accurate for this application (see Chapter 2 of this thesis). Based on this conclusion, the aim of the present thesis became threefold. First, to develop a portable and affordable 3D motion tracking system to assess lower limb biomechanics. The proposed solution had to be single-camera, to reduce its footprint, and had to cost approximately 1% the price of a high-end motion capture system, to make it affordable for schools, gyms and clinical practices. Second, to assess the agreement between this novel methodology and a multi-camera optical motion capture system. Third, to understand if the accuracy of the novel system is sufficient to identify subjects at risk of ACL injury.

It should be noted that the third aim was achieved by comparing the accuracy of the proposed methodology with the difference in lower limb kinematics between subjects at risk of ACL injury and controls found in literature. Although the results of this comparison were promising, future studies will be required to confirm them experimentally. Specifically, the latest iteration of the proposed methodology will need to be used to track a cohort of young girls during ACL-challenging tasks, to demonstrate the ability of this approach to differentiate individuals at high from those at low risk of ACL injury.

The present thesis is structured as follows (Figure 9). Given the poor accuracy achieved by Kinect v2 pose estimation algorithm (see Chapter 2), an alternative tracking approach based on depth data provided by Kinect v2 was proposed. To verify the feasibility of this alternative approach, a literature review on the accuracy of Kinect v2 depth measurements is reported in Chapter 3, to understand if the limited accuracy of the pose estimation algorithm depends on the algorithm itself or on the underlying depth data. Chapter 4 illustrates the novel tracking approach, denoted as Kinect coloured marker tracking (KCMT) and based on the use of custom-made coloured markers and computer vision techniques.

To compare this novel measurement method with an established one, their data must be expressed in the same coordinate system (i.e. aligned in 3D space) and synchronized (i.e. aligned
in time). Therefore, to compare the proposed KCMT approach against a Vicon motion capture system, custom algorithms for coordinate transformation and synchronization are presented in Chapter 5 and 6, respectively. Chapter 7 explains why the Bland-Altman analysis of agreement is the best statistical technique to compare two measurement methods, and how this technique was implemented in the present thesis to compare KCMT- and Vicon-derived variables.

After laying out the methodological foundations required to compare the novel and the gold-standard motion tracking system, two agreement studies between KCMT and Vicon are reported in Chapters 8 and 9, focused on single-leg squat and treadmill locomotion, respectively. The study reported in Chapter 9 was published on the journal Medical Engineering & Physics (Timmi et al. 2018). Lastly, Chapter 10 presents some ongoing and future developments, and Chapter 11 reports the conclusion of this thesis, including a comparison between the accuracy of the proposed methodology and the difference in lower limb kinematics between subjects at risk of ACL injury and controls found in literature.
After the accuracy of Kinect v2 pose estimation algorithm was found to be insufficient to discern subjects at risk of ACL injury from controls, a novel methodology to track coloured markers (KCMT) using Kinect v2 depth data was developed. Furthermore, the agreement between the proposed approach and a marker-based optical motion capture system (Vicon) was assessed.
1.5 References


Butler, D., Noyes, N. & Grood, E., 1980. Ligamentous restraints to anterior drawer in the human


Tran, A.A. et al., 2016. The effect of foot landing position on biomechanical risk factors associated with anterior cruciate ligament injury. Journal of Experimental Orthopaedics,


Zeller, B.L. et al., 2003. Differences in kinematics and electromyographic activity between men

CHAPTER 2

Proof-of-concept study on the accuracy of Kinect v2 pose estimation algorithm for measuring lower limb flexion angles during double-leg squat

2.1 Introduction

As outlined in the first chapter, the initial aim of this thesis was to develop an affordable and compact motion tracking methodology, alternative to conventional marker-based motion capture systems, which could be part of a motion analysis pipeline to distinguish young individuals at high risk of ACL injury from those at low risk.

Microsoft Kinect v2, thanks to its markerless human pose estimation algorithm, its affordability and its portability, represented the ideal candidate as tracking device in the proposed motion analysis approach. In March 2014, the author of the present thesis obtained a pre-release Kinect v2 device (Microsoft 2014d), months earlier than its official release in July 2014, (Microsoft 2014e). At the time, no studies were available in literature about the accuracy of the second iteration of Kinect. However, this information was required to assess the feasibility of using Kinect v2 for motion analysis of ACL-challenging tasks. To fill this gap, the proof-of-concept study reported in this chapter was carried out.

The aim of this preliminary study was twofold: 1) to implement a data processing pipeline, based on signal processing techniques and inverse kinematics, to convert the coordinates of the virtual joint centres provided by Kinect v2 pose estimation algorithm into joint angles for the widely adopted OpenSim Gait2392 musculoskeletal model; 2) to provide a first estimation of the agreement between Kinect v2 markerless tracking algorithm and a marker-based Vicon motion capture system for measuring lower limb joint angles in the sagittal plane.

Due to the limited capture framerate of Kinect v2 (30 fps), landing tasks were avoided due to their high speed, and a slower and more controlled squatting task was preferred instead. As discussed in Chapter 1 of this thesis, there are several studies in literature supporting the use of single-leg squat (SLS) trials to assess the ACL injury risk in young individuals (Zeller et al. 2003; Willson et al. 2006; Yamazaki et al. 2009). Because the aim of this preliminary study was to assess the accuracy of Kinect v2 pose estimation algorithm, rather than the participant’s ACL injury risk, the less challenging double-leg squat (DLS) was selected as benchmark task instead of the SLS. Given the kinematic similarity between single and DLS when observed on a para-sagittal plane, and considering the preliminary nature of this study, the author believes it is fair to consider the results obtained from DLS as a first estimation of those that can be achieved from SLS.

The analysis of agreement reported in the present chapter indicated that the markerless tracking algorithm of Kinect v2 is not accurate enough to distinguish individuals at-risk of ACL injury from those not-at-risk, based on differences reported in literature (Yamazaki et al. 2009; Zeller et al. 2003; Willson et al. 2006; Yamazaki et al. 2009).
For this reason, the off-the-shelf Kinect v2 cannot be used instead of more expensive and cumbersome marker-based motion capture systems to identify subjects at risk of ACL injury. Based on this conclusion and on the hypothesis that Kinect v2 raw depth data are more accurate than the joint positions estimated by the markerless algorithm, an alternative tracking methodology for Kinect v2 was developed and presented in the subsequent chapters of this thesis.

2.2 Methods

2.2.1 Participants

A single, healthy participant (female, 36 years of age, 163 cm, 50 kg, BMI 18.9) was recruited and provided informed consent to this preliminary study, which was part of a larger project approved by the Human Research Ethics Committee at The University of Melbourne (ID 1442604). Since no data were available in literature about the accuracy of Kinect v2, and because this was a proof-of-concept study, no attempts were made to calculate the sample size required to ensure statistical significance of the results. The results of this analysis should be considered as a first attempt to look at the agreement between Kinect v2 and a conventional motion capture system, and any conclusion should be confirmed by more statistically robust studies.

2.2.2 Experimental protocol

Forty reflective markers (⌀14 mm) were attached to the participant’s skin (Figure 10). Eight of them were recorded only during static trials. Such trials were used to scale the OpenSim musculoskeletal model, to match the participant’s anthropometric measurements. This markerset was based on that presented by Dorn et al. (Dorn et al. 2012); however, no markers were attached on the arms because the participant was asked to perform the task keeping the arms crossed on the chest. Reflective markers trajectories were recorded using a 12-camera Vicon motion capture system (Vicon, Oxford, UK) sampling at 120 Hz. Marker trajectories were recorded, reconstructed, labelled and exported using Vicon Nexus v.1.8.5 software.
The trajectories of 25 virtual joint centres (Figure 11) were recorded using the markerless pose estimation algorithm provided by Kinect v2 (Microsoft, Redmond, US). This device was controlled by a custom C# application developed in collaboration with Mr Gino Coates and based on the Kinect for Windows SDK 2.0 (Microsoft 2014c) and the .NET Framework (v4.7.1, Microsoft, Redmond, US). Kinect v2 was fixed on a tripod facing the participant, at 1.7 m of distance and at 0.8 m from the ground. The device tilt angle was adjusted to ensure that its field of view contained the participant, including a safety margin. The nominal capture framerate of Kinect v2 is 30 fps, however its actual framerate is irregular and may be temporarily reduced for different reasons, e.g. random-access memory saturation. The two systems, Vicon and Kinect, were controlled by two separate computers.

The participant was asked to perform a series of 5 DLS trials, which were concurrently recorded using Vicon and Kinect. A total of 368 frames were recorded using Kinect, corresponding to approximately 12 s.

2.2.3 Data analysis

Ad hoc Kinect v2 markerset

Two different markersets were created in OpenSim: the first, reproducing the 40 Vicon reflective markers; the second, replicating the 25 virtual joint centres tracked by Kinect v2 pose estimation algorithm (Figure 12). The former was relatively straightforward to implement, because the positions of the experimental markers attached on the participant’s skin could be physically observed and measured with respect to different body landmarks. In contrast, creating an OpenSim markerset for the Kinect v2 virtual joint centres was more complex, because no physical references were available to estimate the positions of such virtual joint centres relative to body landmarks. Furthermore, Microsoft didn’t provide any details about the local coordinates and the degrees of freedom of the virtual joint centres of Kinect v2 skeleton. However, such parameters were required to scale the generic OpenSim model and to perform the inverse kinematics analysis.
To estimate these unknown parameters, several static and dynamic trials of a subject were recorded using Kinect Studio, an application provided with Kinect SDK 2.0. This application was used to record the depth data and the virtual skeleton of the subject, and to overlay them on screen (Figure 11). Observing such trials, the positions of Kinect virtual joint centres were estimated with respect to the subject’s physical joint centres, which were visually identified in the depth images. Then, these relative positions were reported into a custom OpenSim markerset (Figure 12).

Using this approach, markers representing the shoulder, elbow, wrist, hip, knee and ankle virtual joint centres were positioned at the origins \((x = y = z = 0)\) of the corresponding joint reference systems in the OpenSim model. The relative positions of the remaining joints (spine, feet and hands) were less obvious, so they were estimated taking measurements on screenshots from trials recorded using Kinect Studio (Figure 13, left). The measured proportions were then replicated in the OpenSim markerset (Figure 13, right).
To optimize the resulting OpenSim markerset, a trial and error procedure was applied. A series of scaling tests was performed in OpenSim and, at each iteration, the custom markerset was fine-tuned to minimize the error between the corresponding virtual and experimental marker positions, following the best practices reported in the OpenSim documentation (OpenSim 2017).

**Custom OpenSim model for Kinect v2**

The OpenSim *Gait2392* musculoskeletal model with arms featuring 23 degrees of freedom and 92 musculotendon actuators (Delp et al. 1990; Yamaguchi & Zajac 1989; Anderson & Pandy 1999; Anderson & Pandy 2001) was customized in order to reproduce the same degrees of freedom available in the Kinect v2 skeleton. Since in this study only kinematics was analysed, muscles were hidden in all figures representing this musculoskeletal model.

The Gait2392 model features a rigid spine which is connected to the sacrum via a ball-joint at L5 (Figure 14, left). This joint has three rotational degrees of freedom, i.e. lumbar extension, lumbar bending and lumbar rotation. In contrast, Kinect v2 spine is a rigid segment, rigidly connected to the pelvis (Figure 14, right). Its axial rotation is distributed across all the spine joints, but how this rotation is quantitatively split among the joints is unknown.
Figure 14. Corresponding degrees of freedom of the spine in the OpenSim Gait2392 model (left) and in Kinect v2 skeleton (right). Top: lumbar bending. Middle: lumbar extension. Bottom: lumbar rotation.

The animation of a DLS, generated using the OpenSim model and Kinect-derived joint angles, initially displayed an incorrect pelvis tilt in the sagittal plane (see red arrow in Figure 15, left). To resolve this issue, the lumbar extension of the OpenSim model was locked to its default orientation (i.e. 0° extension), because the same degree of freedom was absent in Kinect virtual skeleton. This customization in the OpenSim model prevented the incorrect pelvis tilt (Figure 15, right).

Figure 15. The animation of a DLS generated using the OpenSim model and Kinect-derived joint angles. The incorrect pelvis tilt (left) was fixed by locking the lumbar extension to 0° in the OpenSim model (right).

Statistical analysis

Vicon-derived marker coordinates were filtered using a 4th order low-pass, zero-phase Butterworth filter with cut-off frequency of 20 Hz (Bisseling & Hof 2006). Then, they were exported to .c3d format and converted to the OpenSim .trc format using Matlab OpenSim Tools v2 (Lichtwark et al. 2014). Using a static standing trial, the generic Gait2392 model was scaled
to match the subject’s anthropometry. Then, using the scaled model and dynamic trials as input, an inverse kinematics analysis was performed to calculate the joint angles during DLS.

The trajectories of Kinect-derived virtual joint centres were filtered using a median filter (span = 5 frames) to remove spikes, and a Savitzky-Golay filter (4th order, span 19 frames) to reduce noise. Using a static standing trial and the Ad hoc Kinect v2 markerset, the Custom OpenSim model for Kinect v2 was scaled to match the subject’s anthropometry. Joint angles were obtained via inverse kinematics and then up-sampled to 120 samples/s via spline interpolation, to match Vicon sampling rate.

Kinect- and Vicon-derived 3D trajectories did not share a common start time, because the two systems were controlled by two different computers. For this reason, the two sets of joint angles were synchronized and trimmed to the same length during post-processing, using the novel synchronization algorithm presented in Chapter 6 of this thesis.

A Bland-Altman analysis of agreement was performed between Vicon and Kinect-derived joint angles (Bland & Altman 1986). This analysis assumes that the differences between the two measurement methods follow a Normal distribution. If this assumption is true, 95% of the differences will lie between the limits of agreement. In this case, the Bland-Altman plot displayed a relationship between the difference and the average of Kinect and Vicon angles. Because the recommended log-transformation did not remove this relationship, the linear regression method suggested by Bland and Altman was applied (Bland & Altman 1999). In this case, the 95% limits of agreement (LOA) were defined by:

$$\text{LOA} = \hat{D} \pm 1.96 \sqrt{ \frac{\pi}{2} \hat{R} }$$

where $\hat{D}$ is the linear regression of the differences and $\hat{R}$ is the linear regression of the absolute values of the residuals of $\hat{D}$. The following steps illustrate how $\hat{D}$ and $\hat{R}$ were obtained.

Using MATLAB, a linear polynomial fit of the differences against the average of the two measurements methods was calculated:

$$[p] = \text{polyfit}(\text{average}, \text{differences}, 1);$$

$$\hat{D} = \text{polyval}(p, \text{average});$$

The residuals of this fit were calculated as:

$$R = \text{differences} - \hat{D};$$

A linear polynomial fit of the absolute values of the residuals against the average of the two measurement methods was determined:

$$[p_{\text{resid}}] = \text{polyfit}(\text{average}, \text{abs}(R), 1);$$

$$\hat{R} = \text{polyval}(p_{\text{resid}}, \text{average});$$

Lastly, $\hat{D}$ and $\hat{R}$ were used in the equation reported above to calculate the regression-based LOA.

2.3 Results

The plots of the synchronized joint angles showed that the agreement between Kinect v2 pose estimation algorithm and Vicon during DLS trials was significantly worse for the ankle flexion
angle compared to hip and knee flexion angles (Figure 16). Indeed, errors up to 25° in correspondence of peak ankle flexion could be observed between Kinect (grey line) and Vicon (black line). The regression-based LOA for the hip flexion angle ranged from −4°, 4° in correspondence of 0° of flexion, to 2°, 14° in correspondence of 80° of flexion. For the knee flexion angle, the LOA ranged from 1°, 12° at −70° of flexion, to −9°, 1° at 0° of flexion. For the ankle flexion angle, the LOA ranged from −19°, 1° at 0° of flexion, to 20°, 35° at 30° of flexion (Figure 17 and Table 2).

![Synchronized hip, knee and ankle flexion angles derived from Vicon and Kinect data, recorded during DLS trials](image)

*Figure 16. Synchronized hip, knee and ankle flexion angles derived from Vicon and Kinect data, recorded during DLS trials*
Figure 17. Bland-Altman plots of the hip, knee and ankle flexion angles derived from Vicon and Kinect v2 data, recorded during DLS trials. Differences (grey circles) and regression-based LOA (black lines) between the two measurement systems are reported.

Table 2. Results of the Bland-Altman analysis of agreement between Kinect- and Vicon-derived joint angles. Since the LOA were regression-based, they were not constant across the range of motion (horizontal axes in Figure 17). Therefore, the LOA could not be described by individual numbers. To provide a better representation of these linearly variable LOA, their values measured in correspondence of the minimum and maximum degree of flexion for each joint were reported in this table.

<table>
<thead>
<tr>
<th>Flexion angle</th>
<th>Degree of flexion</th>
<th>Lower LOA</th>
<th>Upper LOA</th>
<th>Width LOA</th>
<th>Bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hip</td>
<td>0°</td>
<td>−4°</td>
<td>4°</td>
<td>8°</td>
<td>0°</td>
</tr>
<tr>
<td>Hip</td>
<td>80°</td>
<td>2°</td>
<td>14°</td>
<td>12°</td>
<td>8°</td>
</tr>
<tr>
<td>Knee</td>
<td>−70°</td>
<td>1°</td>
<td>12°</td>
<td>11°</td>
<td>6°</td>
</tr>
<tr>
<td>Knee</td>
<td>0°</td>
<td>−9°</td>
<td>1°</td>
<td>10°</td>
<td>−4°</td>
</tr>
<tr>
<td>Ankle</td>
<td>0°</td>
<td>−19°</td>
<td>1°</td>
<td>20°</td>
<td>−9°</td>
</tr>
<tr>
<td>Ankle</td>
<td>30°</td>
<td>20°</td>
<td>35°</td>
<td>15°</td>
<td>28°</td>
</tr>
</tbody>
</table>

2.4 Discussion

In this proof-of-concept study a data processing pipeline was developed, to convert the coordinates of the virtual joint centres provided by Kinect v2 pose estimation algorithm into joint angles for the widely adopted OpenSim Gait2392 musculoskeletal model. Furthermore, a preliminary analysis of agreement between Kinect- and Vicon-derived lower limb flexion angles was carried out, at a time in which no similar studies were available yet, due to the novelty of the second version of the Microsoft device.
Due to the strong relationship observed between difference and average of Kinect and Vicon-derived measurements, constant LOA were not suitable to describe the agreement between the two systems. This relationship was particularly strong for the ankle flexion angle. As suggested by Bland and Altman in such cases (Bland & Altman 1999), regression-based LOA were used for all joint angles. This alternative approach produced narrower LOA compared to the constant ones, but still including 95% of the differences between the two methods of measurements. It should be noted that it is not desirable for the measurement error to depend on the magnitude of the measurement itself. Ideally, the LOA of agreement should be not only narrow, but also constant throughout the range of measurement, and the bias should be zero.

Yamazaki et al. estimated the differences in kinematics of SLS between ACL-injured patients and healthy controls (Yamazaki et al. 2009). Among other quantities, they compared the mean joint angle at maximum knee flexion between the uninjured limb of ACL-injured patients and the dominant limb of healthy participants. Considering only female individuals because they are more likely to incur non-contact ACL injuries (see section 1.1 ACL injury risk in young girls), their results showed the following statistically significant differences: 7.7° for the knee flexion angle and −18.1° for the hip flexion angle. In the preliminary study reported in the present chapter, the LOA between Kinect and Vicon measured at maximum flexion were 1°, 12° for the knee flexion angle, and 2°, 14° for the hip flexion angle. These results indicate that the Kinect v2 pose estimation algorithm is not accurate enough to detect significant differences for the knee flexion and may barely detect those for the hip flexion.

In 2016, i.e. 2 years after the present analysis was carried out, McGroarty et al. published a study on the accuracy of Kinect v2 pose estimation algorithm for measuring the peak knee flexion angle during overhead squat (McGroarty et al. 2016). Seven subjects performed 3 repetitions of this task, while motion data were concurrently recorded using a 6-camera Vicon system and Kinect v2. The Kinect-derived knee flexion was calculated as the angle between the projections of the upper and lower leg vectors (defined by hip, knee and ankle virtual joint centres) onto the Kinect YZ plane, where Y and Z were the vertical and depth directions respectively. Only mean and standard deviation of the absolute error between Kinect- and Vicon-derived measurements were discussed in the paper. However, the use of absolute values of the errors introduced an artefact in the calculation of mean and standard deviation, because the distribution of the error between Kinect and Vicon was incorrectly assumed to be one-sided. In fact, the raw measurements of the peak knee flexion angle reported in that study indicated that this distribution was two-sided. Therefore, using the aforementioned raw measurements, the author of the present thesis calculated the LOA, which were equal to −16°, 8° and −17°, 6° for the left and right peak knee flexion angle, respectively. In the proof-of-concept study reported in the present chapter, the LOA of the peak knee flexion angle between Kinect and Vicon were 1°, 12°, i.e. narrower compared to those found using data from McGroarty et al. This difference may be explained by the use of regression-based LOA in the present study, which generate narrower intervals compared to constant LOA, and by the small sample size. Nonetheless, the wide LOA based on data by McGroarty et al. confirmed the poor accuracy of the Kinect v2 markerless tracking algorithm already observed in the present study.

Mentiplay et al. performed a Bland-Altman analysis of agreement between Kinect v2 markerless tracking algorithm and Vicon for comfortable pace gait (Mentiplay et al. 2015). LOA of the hip and ankle flexion angles were linearly variable, meaning that the disagreement for those variables was proportional to the magnitude of the measured angle. This finding is in line with the results of the present study. The LOA were different between the study by Mentiplay et al.
and the present one, however their widths were comparable. In the study by Mentiplay et al., the width of the LOA of the hip flexion angle was 23° across the range of motion, whereas it ranged between 8° and 12° in the present study. For the ankle flexion angle, the width was 28° across the range of motion, while it ranged between 20° and 15° in the present study. Several factors may explain the different LOA found in the study by Mentiplay et al. compared to those reported in the present study: the different tasks (gait as opposed to DLS), the variable distance between participant and Kinect adopted in the study by Mentiplay et al. as opposed to the constant distance used in the present study, and the smaller sample size used in the present study. However, the widths of the LOA and the relationship observed between difference and average of the two measurement methods are in line with the results of the present study.

The accuracy of Kinect-derived joint angles is influenced by the accuracy of the virtual joint centres tracked by the pose estimation algorithm. Xu & McGorry compared Kinect v2 markerless algorithm with a marker-based system and found the accuracy to be posture- and joint-dependent (Xu & McGorry 2015). For an upright standing posture, they found the average error across all participants and joint centres was 87±15 mm. Poorest agreement was at the feet, with an average deviation of 207±71 mm reported for the left foot. Wang et al. oriented Kinect v2 in different configurations relative to the participants during both sitting and standing tasks (Wang et al. 2015). Considering only standing exercises with Kinect facing participants (best-case scenario), the mean error of the markerless algorithm for lower body joints ranged between 103 and 146 mm (SD ≈ 30 mm), with ankle and foot kinematics being most inaccurate. In turn, the accuracy of pose estimation depends on the accuracy of the underlying depth data measured by the Kinect v2 depth sensor (Girshick et al. 2011; Shotton et al. 2011; Shotton et al. 2012; Taylor et al. 2012; Kohli & Shotton 2012). The different sources of error affecting Kinect v2 depth map are illustrated in Chapter 3 of this thesis, and their effects are quantified based on data from several metrological studies found in literature.

The study presented in this chapter has some limitations: first, results are not statistically significant due to the limited sample size. However, they are in line with those published in subsequent works featuring larger sample sizes. Second, the analysed task is a DLS, therefore considerations found in literature regarding the SLS may not directly apply in this case. However, because the present study was focused on sagittal plane kinematics, and considering the similarity between SLS and DLS when observed on a para-sagittal plane, the author believes it is fair to consider the flexion angles measured during DLS as a reasonable surrogate of those measured during SLS, especially in a proof-of-concept study. Third, the measurement error found in this study was entirely attributed to the limited accuracy of Kinect v2 pose estimation algorithm. However, secondary factors such as skin motion artefact affecting reflective markers and the use of different data filtering techniques between Kinect and Vicon signals may have influenced the joint centres position and consequently contributed to the observed errors.

In conclusion, based on the results from the present and following studies, Kinect v2 pose estimation algorithm is not sufficiently accurate to assess the ACL injury risk by measuring the sagittal plane kinematics of a squat task. Indeed, the LOA were comparable or larger than the significant differences between at-risk and control subjects reported in literature. The poor agreement with Vicon for measuring lower limb flexion angles may depend on inaccuracy of the underlying depth data used as input by the pose estimation algorithm, and on intrinsic limitations of the pose estimation algorithm itself.

During this study, there was the opportunity to visually inspect the depth images generated by the Kinect v2 depth sensor. Observation of the level of detail of such images led to develop the
hypothesis that depth data might be significantly more accurate compared to the pose estimation algorithm, meaning that the pose estimation algorithm may be largely responsible for its own poor tracking accuracy. If this hypothesis was true, depth data might be used to develop a novel 3D tracking methodology, alternative to the in-built pose estimation algorithm. To assess the validity of this hypothesis, all sources of error affecting the Kinect v2 depth map are illustrated in the next chapter, and their effects are quantified based on metrological studies found in literature.

2.5 References


CHAPTER 3
Errors affecting the Kinect v2 depth map

3.1 Depth measurement

Kinect for Windows v2 (Microsoft, Redmond, US), also known as Kinect for Xbox One, is a gaming device which features a markerless human pose estimation algorithm. Its markerless motion tracking capability is based on the use of depth images, which are generated by an embedded depth camera. The technology used by the second generation Kinect for depth sensing is denoted as Continuous wave time-of-flight (CW TOF) (Lachat, Macher, Mittet, et al. 2015; Corti et al. 2015; Sarbolandi et al. 2015). This system works by illuminating the scene with a modulated light source and analysing the reflected light. The phase-shift between the illumination and the reflection is measured and translated into a distance measurement. The illumination is typically in the near-infrared range: for Kinect v2 it consists of a laser diode at 850 nm wavelength. An imaging sensor (hereinafter denoted as depth sensor) working in the same spectrum receives the light and converts the photonic energy into electrical current. The light entering the sensor has an ambient component and a reflected component, however only the latter carries depth information: the higher the ambient component, the lower the signal-to-noise ratio (Li 2014).

For each collected frame, the depth sensor returns a matrix (called depth map) with same size in pixels as the sensor (i.e., 512 x 424 px). For each pixel of the depth map, Kinect provides the distance value in metres between the sensor and the corresponding point in the scene. Each pixel of the depth map is then internally converted to a 3D point, whose coordinates are expressed in the so-called camera space. The origin of camera space corresponds to the centre of the Kinect depth sensor. The associated coordinate system is right-handed and its axes are oriented as follows: Z is the depth axis, i.e. it is perpendicular to the image plane and directed as the optical axis of the depth camera; Y is the vertical axis (directed superiorly with respect to the device); X is determined according to the right-hand rule.

Points in camera space comprise a 3D point cloud, and can be accessed via the Coordinate mapper (Microsoft 2014a), a class provided by Kinect v2 SDK. Because the Cartesian coordinates of the 3D points are calculated using the depth measurements as input, it is important to identify the different sources of error affecting Kinect v2 depth sensor and to quantify their effects. The aim of the present chapter is to review these sources of error based on studies reported in literature, and to classify them in categories which are relevant to the objectives of this thesis. The results reported in this chapter provide detailed information on the accuracy and precision that can be anticipated from the proposed Kinect coloured marker tracking (KCMT) system.

---

1 The 3D point cloud generated by Kinect v2 is an essential component of the novel marker tracking algorithm introduced in Chapter 4 and validated in Chapter 9 of this thesis.
3.2 Inclusion criteria of this literature review

The papers reviewed in the present chapter were retrieved during the period 2014—2016 using Google Scholar email alert system, based on the following keywords: Kinect2, Kinect, KinectV2, Kinect for Windows v2, Kinect Xbox One. Over time, additional searches were also manually performed, including more general keywords such as: depth, time-of-flight, accuracy, precision, characterization. Only publications focusing on the metrological characterization of Kinect v2 depth sensor were included. Those exclusively related to Kinect v1, or to specific applications of Kinect, were excluded. At the end of this literature review process, a total of 8 papers was shortlisted.

From these 8 studies, the paper by Goral & Skalski (2015) was excluded for a number of reasons: 1) the authors reported only limited statistical results to support their conclusions; 2) the paper wasn’t published in a peer-reviewed journal; 3) the conclusion that the random depth error is independent on the distance between target and Kinect v2 was in strong disagreement with the rest of the literature, which indicated that the random error is linearly dependent on the distance; 4) the presented data capture protocol was quite complex and worked only under specific conditions, which were too restrictive compared to the normal use of Kinect v2.

The remaining 7 peer-reviewed publications reported a wide range of experiments aimed at assessing different metrological characteristics of the Kinect v2 depth sensor. In this review, all these experiments and the corresponding results were grouped in the following 7 categories (Table 3): temperature drift, accuracy of the centre of the depth sensor, precision of the centre of the depth sensor, uniformity of the accuracy of the depth sensor, uniformity of the precision of the depth sensor, effect of colour and reflectivity, effect of motion.

Table 3. Reviewed papers and metrological categories discussed in each of them

<table>
<thead>
<tr>
<th>Temperature drift</th>
<th>Accuracy (sensor centre)</th>
<th>Precision (sensor centre)</th>
<th>Uniformity of accuracy</th>
<th>Uniformity of precision</th>
<th>Colour and reflectivity</th>
<th>Effect of motion</th>
</tr>
</thead>
<tbody>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
<tr>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
</tr>
</tbody>
</table>

Check mark (✓) = category analysed in the paper; Empty cell = category not analysed in the paper.

It should be noted that most studies reported the accuracy and precision of depth data separately for the centre and for the entire surface of the depth sensor. This approach was used...
because the periphery of the sensor is more affected by optical lens distortion and by decreased
signal intensity, making it less accurate and precise compared to the centre of the sensor (see
also section 3.2.4 Uniformity of accuracy of the depth sensor). For this reason, the uniformity of
precision and accuracy across the entire sensor were assessed via ad hoc analyses.

3.2.1 Temperature drift
The accuracy of Kinect v2 depth measurements is affected by the temperature of the internal
electronics. This source of error is common to electronic measurement devices and is called
temperature drift. To stabilize its internal temperature, Kinect v2 has an in-built fan that is
automatically switched on by a thermostat.

In a metrological characterization of Kinect v2 Corti et al. (2015), the device was placed 1 m away
from a planar wall and 20,000 depth readings where recorded from the central pixel of its depth
sensor. For the entire duration of the test (10 min), the internal fan remained off, because the
temperature didn’t reach the thermostat setpoint. The test was then repeated using an external
fan attached behind Kinect and switched on from the beginning. The first test showed a drift of
2.5 mm in the measurement, while the second (with the external fan) showed no drift, but only
a random error (noise) with a range of 0.6 mm.

These findings are in line with those reported by Lachat et al. (Lachat, Macher, Mittet, et al.
2015; Lachat, Macher, Landes, et al. 2015). Their papers report a similar test, performed at a
distance of 1.36 m for 1.5 h, during which 360 frames were collected. The depth measurements
were averaged across a central area of the sensor of 10 x 10 px. The results showed a drift of
about 5 mm during the first 30 minutes, however the signal became practically stable after the
internal fan switched on 20 min from the beginning of the test.

In a similar study by Sarbolandi et al. (2015), Kinect was positioned at 1.2 m from a planar wall,
in a room with a constant temperature of 21°C. Two hundred consecutive frames were collected,
whereas the following 450 were dropped, and this process was repeated for a total capture time
of 120 min. A reference plane was calculated as Random sample consensus (RANSAC) fit of the
first 200 frames. Then, the root-mean-square error (RMSE) and the standard deviation average
(SDA) were determined for each sequence of 200 frames with respect to the initial reference
plane. The results indicate that the error started at RMSE = 5.3 mm and, after the 20-min mark,
stayed practically stable at RMSE = 4.8 mm. The standard deviation average was constant at SDA
= 1.5 mm.

Fursattel et al. (2015) collected depth readings from a 9 x 9 px area at the centre of the sensor
and averaged them over a 1-s interval to reduce noise. The distance between Kinect and target
was not reported in the paper. Images were recorded for 120 min and data from the final 10
min were averaged and considered as steady state. The authors reported a difference of less
than 2 mm between start-up and steady state, which was reached after approximately 15 min
of warm-up.

Steward et al. (2015) positioned a white target with 99% reflectivity at 1 m of distance from
Kinect and recorded 30 frames every 5 min for 3 h. For each set of 30 frames, a least-squares
plane was calculated. Then mean and standard deviation of the distance Kinect-plane over time
were calculated. The trend of this distance indicated an initial 10-min warm-up period, and a
maximum drift of approximately 3.3 mm between start-up and steady state.

The reported findings indicate that the Kinect v2 depth sensor is subject to temperature drift,
which affects its accuracy by 5 mm at worst. However, this source of error can be practically
neutralized observing a warm-up time of about 20 minutes before using the sensor for capturing trials. This interval of time is sufficient to raise the temperature of the device above the thermostat setpoint, activating the fan and consequently stabilizing the internal temperature. In the studies presented in the subsequent chapters of this thesis, a warm-up phase of 20 min was allowed for Kinect v2 before capturing data.

3.2.2 Accuracy of the centre of the depth sensor

The accuracy of Kinect v2 may also be affected by a systematic error, sometimes referred to as wiggling error. This error is caused by the imperfect generation of the modulated infrared light, which is not perfectly sinusoidal (Fursattel et al. 2015). The deviation from this ideal shape causes an oscillating error in the depth measurements, which is function of the distance from the target. In this section, experimental results from the reviewed papers were digitized using a web app (WebPlotDigitizer, Rohatgi n.d.). Data were then corrected to account for the different protocols used for data collection and analysis, as specified in the rest of this section (Figure 18).

Lachat et al. (Lachat, Macher, Mittet, et al. 2015; Lachat, Macher, Landes, et al. 2015) moved Kinect v2 from 1 to 6 m of distance from a wall, with steps of 25 cm (plus an additional station at 0.8 m). For each station, 50 depth maps were collected with intervals of 1 s, using a central sensor area of 10 x 10 px. For each true distance, a measured distance was obtained from Kinect as spatial (across a 10 x 10 px central area) and temporal (over 50 measurements) average, thus minimizing the random error.

Since all deviations, calculated as true distance – Kinect measurement, were positive, the authors recommended to subtract a constant bias, corresponding to the distance between the depth sensor and the fixing screw of the device (Lachat, Macher, Mittet, et al. 2015). This offset was quantified as the error measured at the closest distance. Data from this experiment after removing the bias are reported in Figure 18 (red line). A change of sign was also applied to all deviations, to represent them as Kinect measurement – true distance. Considering only the range of distances providing maximum accuracy (0.8 – 4.5 m), the deviations varied between −0.4 and 0.9 cm.

In their second paper Lachat, Macher, Landes, et al. (2015), the authors corrected their dataset in a different way, i.e. subtracting the offset between the fixing screw of Kinect and its lens (about 2 cm). However, the origin of the depth axis is located on the plane of the depth sensor, not on the lens, and its position cannot be determined without disassembling the device. For this reason, in the present chapter, the approach suggested in the first study (i.e. subtracting the error measured at the closest distance) was utilized.
Figure 18. Differences between Kinect measurements and true distances. Data are from the current thesis and from reviewed studies. All datasets display similar trends for the wiggling error, typical of TOF cameras. Although the same approach for removing the constant bias was applied, there are residual differences between signals, which are due to the different protocols used for data collection and analysis. The figure legend reports in parentheses the corrections applied to each dataset, e.g. change of sign (sign) or bias subtraction (bias).

In a similar experiment, Corti et al. used a robotic arm to move a planar target from 0.8 to 4.2 m of distance from Kinect, with a step of 20 mm (Corti et al. 2015). At each distance, 4000 depth maps were recorded. For this test, only the depth values from the central pixel were analysed. The true measurement was identified as the displacement of the target from its starting position ($d$, Figure 19).

![Figure 19](image)

Figure 19. A sketch representing the data collection protocol used by Corti et al. (2015). The dashed rectangle indicates the starting position $k_0$ of the target measured with respect to Kinect v2. The solid rectangle indicates the position $k$ of the target after a displacement, whose true value $d$ was measured by the robotic arm supporting the target.

The error $e$ between true and measured displacement was calculated as:

$$ e = d - (k - k_0) $$

where $d$ was the true displacement obtained from the robotic arm supporting the planar target; $k$ was the distance measured by Kinect after the displacement; $k_0$ was the distance measured
by Kinect when the target was at the starting position of the test. To compare this study with others, the initial true distance Kinect-target (0.8 m) was added to the array of true displacements of the target, thus converting them to true distances between Kinect and the target.

Although the authors described this dataset as the difference between real and measured distances, a visual comparison with data from other studies suggested the opposite. Under the assumption that data were reported either with the wrong sign or with an incorrect description, no further change of sign was applied, i.e. data were considered as reporting the difference Kinect measurement – true distance (Figure 18, green line). The deviations ranged between −0.9 cm and 1.5 cm. This range is slightly larger compared to that reported by Lachat et al. (Lachat, Macher, Mittet, et al. 2015), probably because Lachat et al. averaged measurements across a 10 x 10 px area instead of using a single pixel, thus reducing the random component of error.

Sarbolandi et al. (2015) installed Kinect v2 on a motorized linear rail, positioned perpendicular to a planar wall. During the test, the distance between Kinect and the wall ranged from 0.5 to 5 m with a step of 2 cm, however only results up to 4 m were reported in the paper. A central pixel with coordinates 263, 203 px (columns, rows) was selected and 200 frames for each distance were collected. Measurements were averaged temporally (across the 200 frames) for each distance to minimize the random error. The deviations between depth measurements and ground truth ranged from −0.8 to 2.9 cm and, up to a distance of 2.5 m from the wall, were in good agreement with those from other studies (Figure 18, orange line).

Fursattel et al. (2015) used a white checkerboard mounted on a slider as target, and a stereo camera system to determine the true distance in the range 0.8 – 3 m. For each 1-cm step of the linear slider, the average of 25 depth maps was computed. Only depth readings corresponding to white checkerboards squares were used, to avoid colour-related effects (see section 3.2.6 Effect of colour and reflectivity). Although the study didn’t specify how error was defined, comparison with results from other studies suggested that deviations were expressed as true distance – Kinect measurement. Hence, their sign was changed and a conversion from m to cm was applied. Furthermore, to remove the constant bias (Lachat, Macher, Mittet, et al. 2015), the deviation obtained at 0.8 m of distance from the target was subtracted from all deviations (Figure 18, cyan line). The resulting wiggling error ranged between −0.5 and 1.3 cm and its trend was in good agreement with those from the other studies.

Pagliari & Pinto (2015) collected 100 depth maps for each distance from a wall in the range 0.8 – 4 m, using irregular steps with an average of 0.4 m. Custom intrinsic parameters were used to correct the distortion of Kinect depth camera. True distances were measured using a laser metre placed at the two extremities of the sensor, to minimize the orientation error. To avoid recording depth readings of the floor, the analysis was restricted to a central 200-by-200-px window of the sensor. For each acquisition step, the average of each pixel in the window was computed across the 100 images. Although not stated, it seems that a spatial average across all pixels in the window was also performed, to get a single deviation for each station. The authors didn’t specify how the error was defined, but comparing this dataset with those from other studies, it seems that deviations were expressed as Kinect measurement – true distance. For consistency with the other studies, deviations were converted from m to cm, and the error measured at 0.8 m of distance from the wall was subtracted from all deviations (Figure 18, magenta line). The wiggling error ranged from −0.7 to 0.9 cm and its trend was in good agreement with those from the other studies.
In 2014, a similar accuracy test was performed prior to the studies presented in this thesis (Timmi 2014, unpublished data). Kinect v2 was moved along a linear path on the floor, perpendicular to a flat rectangular target attached to the wall. A measurement tape mounted to the floor was used as ground truth. The true distances ranged from 0.5 to 3 m, and a step of 2.5 cm was used. Measurements were taken from a single central pixel, selected by the user for each station via a mouse-click on the depth image in correspondence of the target displayed on screen. Each measurement was automatically calculated as the average across 30 consecutive depth frames. Values collected at a distance shorter than 0.6 m were disregarded, due to the high level of noise observed in raw depth data. The wiggling error ranged from −1.4 to 0.3 cm and its trend was in excellent agreement with those from the other studies (Figure 18, blue line).

Although data collection and analysis protocols were different, all studies displayed similar results, especially in the range 0.8 – 2.5 m. The amplitude and trend exhibited by the wiggling error were consistent across all studies. The most rigorous experiments were those based on measurements from a single pixel (Corti et al. 2015; Sarbolandi et al. 2015; Timmi 2014), because no spatial averaging was performed on the raw depth values. Considering all studies, the centre of Kinect v2 depth sensor achieves its maximum accuracy in the range 0.8 to 3 m from the target. The systematic error is function of the distance and ranges from ±0.5 cm at 1 m, up to ±1.5 cm at 3 m. Interestingly, the observed range of maximum accuracy is similar to the sweet spot (0.8 – 3.5 m) recommended by Microsoft for Kinect v2 markerless pose estimation algorithm (Microsoft 2014b). Based on these findings, all studies performed in this thesis were executed within this depth range.

3.2.3 Precision of the centre of the depth sensor

Random error (also denoted as noise) reduces the precision of depth measurements provided by Kinect. Most of the results presented in this section were obtained from the same experiments already described in the previous section. However, in the present section, the standard deviation of the depth measurements is reported, instead of the mean.

Corti et al. (2015) used a robotic arm to move a planar target from 0.8 to 4.2 m of distance from Kinect, with a step of 20 mm. For each distance, 4000 depth maps were recorded, but only measurements from the central pixel were used for this test. The standard deviation of these measurements was plotted against the true distance Kinect-target. The resulting trend was linear, ranging from 1.2 mm at 1.5 m to 3.3 mm at 4.2 m, and featuring an initial noisy baseline.

Lachat, Macher, Landes, et al. (2015) moved Kinect v2 from 1 to 6 m of distance from a wall, with steps of 25 cm (plus an additional station at 0.8 m). For each true distance, 50 depth maps were collected with intervals of 1 s, using a central 10-by-10-px window of the sensor. For each depth map, a measured distance was calculated as spatial average across this window. The standard deviation of the 50-element sample for each station was plotted against the corresponding true distance. The resulting trend ranged from 0.5 mm at 1 m to 1.3 mm at 4.5 m, with a spike of 2.4 mm at 0.8 m. As for the wiggling error, the spatial averaging performed by Lachat et al. across the central window may explain the smaller deviations found in this study, compared to those reported by Corti et al.

Sarbolandi et al. (2015) calculated the standard deviation of a central pixel with coordinates 263, 203 px (columns, rows) using a sample of 200 consecutive frames per station. The standard deviation linearly increased from 1.2 mm at 0.5 m to 4 mm at 4 m.
Fursattel et al. (2015) performed a separate experiment to evaluate precision. They positioned Kinect at 0.8 m from a white board perpendicular to the optical axis. Depth values of 1000 consecutive measurements from the central pixel were recorded. Additionally, the same measurements were performed with a grey and a black paper foreground, to evaluate the dependency on the foreground colour. A standard deviation of 1 mm was found for the white and for the grey foreground, whereas 2 mm were observed for the black foreground. These results suggest that noise in depth measurements from the central pixel increases as the foreground becomes darker. This is due to the lower signal-to-noise ratio of the IR light reflected by darker targets (see section 3.2.6 Effect of colour and reflectivity).

Pagliari and Pinto (Pagliari & Pinto 2015) used the same dataset described in the previous section to calculate the standard deviation across a central 200-by-200-px window of the sensor. It is unclear whether the authors also performed a temporal average across the consecutive 100 depth maps recorded for each position. The plot of the standard deviation against the true distance displayed a linear trend, ranging from 1 mm at 0.8 m to 3 mm at 4 m of distance.

In summary, the results from the reported experiments are in reasonable agreement and suggest that the precision of the central area of Kinect v2 depth sensor depends on the distance from the target. As with accuracy, the experiments using a single pixel displayed worse results compared to those using a central area of the depth sensor. This is due to the noise-reduction effect of the spatial average. At worst, the precision of the central pixel of Kinect v2 ranged from 1.2 mm at 0.5 m to 4 mm at 4 m of distance from the target (Sarbolandi et al. 2015).

3.2.4 Uniformity of accuracy of the depth sensor

The accuracy of Kinect v2 is not uniform across the surface of its depth sensor, because the infrared light cone, and so the illumination of the scene, is not homogeneous (Corti et al. 2015). Lachat et al. measured the residuals between the point cloud of a planar wall and a least-squares plane of best fit. The experiment was repeated at 10 distances in the range 0.8 – 3 m. To reduce noise, 50 depth maps were recorded for each distance, with intervals of 1 s. The authors found that the point clouds were curved rather than planar, and that the residuals were normally distributed and centred in zero (Lachat, Macher, Mittet, et al. 2015). Furthermore, residuals were larger at the boundaries of the depth sensor, especially at the corners. At 0.8 m distance, residuals were comprised between −4 and 4 mm for the majority of the sensor area, except for corners where deviations reached up to 10 mm (Lachat, Macher, Landes, et al. 2015). At 1.25 m of distance, the trend was identical, but the residuals ranged from −6 mm to 6 at the centre, and reached up to 12 mm at the corners. This radial effect increased with the distance between sensor and target, reaching tens of centimetres at the corners at 3 m range. No clear explanation was reported about how the sign of the residuals was obtained. To maximize accuracy, the authors recommended to use only a central area of the depth sensor.

Corti et al. (2015) carried out a similar test. The 4000 depth maps acquired at 1250 mm from a planar wall were averaged pixel by pixel. Then, this average depth map was passed to the Coordinate Mapper of Kinect v2 SDK, obtaining a 3D point cloud, which in turn was used to compute a reference best-fitting surface. The deviations between the mean depth map and the fitted surface were confined into a band of less than 20 mm. The residuals were within −4 and 4 mm across most of Kinect depth sensor area and reached up to −10 mm in corner regions. The results indicate that Kinect v2 overestimates the depth in correspondence of the centre of the sensor and underestimates it in proximity of its corners.
The loss of signal intensity at the corners of the sensor is also called vignetting (Steward et al. 2015). Steward et al. captured depth maps of a flat and homogeneous wall spanning the entire field of view of Kinect; however, the authors didn’t specify the distance used for this test. The residuals between a depth map and a fitted plane were calculated. Then, a cross section of the residuals image was reported in a plot, featuring the pixel coordinate on the horizontal axis and the error on the vertical axis. Residuals were constant and equal to 5 mm between 0 and 150 px of distance from the centre of the sensor; then they increased sharply, reaching up to 20 mm towards the image periphery. The authors recommended to use only the central 300-by-300-px window of Kinect depth sensor for maximum precision.

Table 4. Summary of the most significant results from different studies on the uniformity of accuracy across the Kinect v2 depth sensor. In this table, when referring to the centre of the depth sensor, a small number of pixels located at the borders and corners were excluded. Results from Steward et al. (2015) were not included in this table, because the distance Kinect-target was not specified in that study.

<table>
<thead>
<tr>
<th>Study</th>
<th>Distance Kinect-target (m)</th>
<th>Sensor area</th>
<th>Residuals (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Lachat, Macher, Landes, et al. 2015)</td>
<td>0.8</td>
<td>Centre</td>
<td>±4 &lt; 10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Corners</td>
<td></td>
</tr>
<tr>
<td></td>
<td>1.25</td>
<td>Centre</td>
<td>±6 &lt; 12</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Corners</td>
<td></td>
</tr>
<tr>
<td>(Corti et al. 2015)</td>
<td>1.25</td>
<td>Centre</td>
<td>±4</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Corner</td>
<td>&gt; −10</td>
</tr>
</tbody>
</table>

In summary, the results reported in this section indicate that residuals may range, at worst, between −6 and 6 mm (when excluding the borders of the sensor) at a distance up to 1.25 m from the target. These residuals follow a radial trend and increase with the distance to the target, reaching up to tens of cm at the corners of the sensor at 3 m of distance from the target.

The results reported for the uniformity of the accuracy may depend on various factors. First, the radial trend of the residuals suggests that an optical distortion may be introduced by the lens of the depth camera (see section 3.2.8 Optical lens distortion). Second, the relationship between the residuals and the distance Kinect-target may depend on the intensity of the reflected IR light received by the depth sensor. Indeed, intensity of light as a function of the distance from the light source follows an inverse square relationship. In this case, the larger the distance Kinect-target, the lower the intensity of the reflected IR light received by the depth sensor. This lower intensity corresponds to a lower signal-to-noise ratio, which in turn may increase the systematic error. Therefore, the depth error increases with the square of the distance Kinect-target (Zhang 2012).

3.2.5 Uniformity of precision of the depth sensor

Precision, similarly to accuracy, is not uniform across the area of the depth sensor (Corti et al. 2015). Corti et al. positioned Kinect at a distance of 1250 mm from a planar wall and acquired 4000 depth maps. Then, they calculated the standard deviation of each pixel. The results indicated a relation between the random error (quantified via the standard deviation) and the radial coordinate on the sensor plane: corner pixels were noisier (SD ≤ 8 mm) than central ones (SD < 2 mm), due to the reduced amplitude of the IR illumination in corner regions (Sarbolandi et al. 2015). This analysis was extended to several distances from the wall in the range 750 – 3750 mm. The results indicated proportionality between standard deviation and distance from
the wall, with SD > 15 mm at the corners in correspondence of larger distances. However, for practical use conditions, the standard deviation can be considered SD ≤ 10 mm, if the borders and corners of the depth sensor are not used.

Sarbolandi et al. (2015) positioned Kinect on a motorized linear rail perpendicular to a white planar wall. The rail spanned the range 0.5 – 5 m, with a step of 2 cm. For each distance from the wall, the authors collected 200 depth maps from a region of interest delimited by two pixels with the following coordinates (columns, rows): 74, 4 px and 502, 416 px. The recorded depth measurements were converted to a 3D point cloud using custom intrinsic parameters. For each distance, a plane was fitted to the corresponding point cloud. The standard deviation of the distances between the point cloud and the corresponding plane was SD < 1.65 mm up to 4 m of distance from the target.

Pagliari & Pinto (2015) collected 100 depth maps of a complex office scene, with 4 m maximum depth. The standard deviation across the depth sensor showed the typical radial trend found in other studies, with values ranging between 0 ≤ SD ≤ 10 mm in the central area of the sensor, and up to SD = 40 mm in the corners. Objects closer to Kinect returned more precise measurements compared to those located further away. Furthermore, surfaces which were very dark, reflective, or with high angle of incidence with respect to Kinect optical axis, returned highly imprecise readings (SD > 25 mm).

Pagliari & Pinto also performed another similar experiment, this time using a flat reference surface parallel to the depth sensor plane. After positioning Kinect at 0.9 m from this surface, 100 depth maps were recorded, and the standard deviation was computed for each pixel. The standard deviation increased sharply at the corners (up to SD = 5 mm), whereas it ranged between 1 < SD < 2 mm across the rest of the sensor area, exhibiting the aforementioned radial trend. The experiment was repeated at different true distances from the surface. Only pixels with standard deviation SD < 5 mm were plotted, because this was considered by the authors as a reasonable threshold for medium quality 3D modelling. The standard deviation increased with the distance from the reference plane, ranging from about SD = 1 mm at 0.8 m up to SD = 3 mm at 3.7 m in the centre of the sensor. A radial trend was observed again: the number of pixels with SD < 5 mm was progressively smaller when the distance Kinect-surface increased, and missing pixels (i.e. those with SD > 5 mm) were located at the borders and corners of the depth sensor.

Precision depends on the distance Kinect-target and varies along the radial direction on the sensor plane. All studies are in good agreement (Table 5): ignoring the borders and corners of the depth sensor, random error ranges from about 1 mm at 0.8 m, up to 10 mm at 3.75 m (at worst). Borders and corners achieve substantially worse results, with SD up to 40 mm.
Table 5. Summary of the most significant results from different studies on the uniformity of precision across the Kinect v2 depth sensor. In this table, when referring to the centre of the depth sensor, a small number of pixels located at the borders and corners were excluded.

<table>
<thead>
<tr>
<th>Study</th>
<th>Distance Kinect-target (m)</th>
<th>Sensor area</th>
<th>SD of depth (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Corti et al. 2015)</td>
<td>1.25</td>
<td>Centre</td>
<td>SD &lt; 2</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Corners</td>
<td>SD ≤ 8</td>
</tr>
<tr>
<td></td>
<td>0.75 – 3.75</td>
<td>Centre</td>
<td>SD ≤ 10</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Corners</td>
<td>SD &gt; 15</td>
</tr>
<tr>
<td>(Sarbolandi et al. 2015)</td>
<td>0.5 – 4</td>
<td>Centre</td>
<td>SD &lt; 1.65</td>
</tr>
<tr>
<td>(Pagliari &amp; Pinto 2015)</td>
<td>0 – 4</td>
<td>Borders and corners</td>
<td>SD ≤ 40</td>
</tr>
<tr>
<td></td>
<td>0.8</td>
<td>Centre</td>
<td>SD ≤ 1</td>
</tr>
<tr>
<td></td>
<td>3.7</td>
<td>Centre</td>
<td>SD ≤ 3</td>
</tr>
</tbody>
</table>

3.2.6 Effect of colour and reflectivity

TOF cameras are affected by increased systematic and random error when measuring depth of objects with low near-infrared reflectivity (Sarbolandi et al. 2015). Because the near-infrared reflectivity of an object is determined by its colour, material and superficial finish, the following studies used targets with different properties to better understand and quantify this source of error.

Corti et al. (2015) attached samples of different materials (cardboard, adhesive film and fabric) and colours (black, white, yellow, red, blue and green) to a planar surface, one at a time. The board was positioned at 775 mm from Kinect and 4000 depth values were recorded. Mean and SD were calculated for each sample object. The SD was about 1 mm for all samples. Differences between the means were small: cardboard and adhesive film returned a difference of 0.8 mm between the nearest and the farthest detected colours, whereas for fabric this difference was 0.5 mm. The effect of colour was different for each material; therefore, no general conclusion was drawn on the interaction between colour and depth measurement.

This test was repeated using a wider spectrum of materials. The SD was again equal to about 1 mm for all samples. The following results were found for the mean of the depth measurements, reported in ascending order: aluminium (774.8 mm), plastic (775.2 mm), fabric (775.3 mm), cardboard (775.4 mm), and wood (775.5 mm). The results indicated that the most reflective material (aluminium) was detected as the closest to the camera (774.8 mm) and returned a slightly underestimated mean depth measurement, whereas the least reflective material (wood) was detected as the farthest away, with a slightly overestimated mean depth measurement. However, the deviation between the two extreme results was only 0.7 mm. This suggests that the reflectivity of the target surface is correlated to the intensity of the backscattered infrared signal measured by Kinect depth sensor.

The authors also found that, independently of the distance, Kinect v2 returned invalid depth readings when trying to measure a highly reflective fabric commonly used for IR-based optical motion capture. Corti et al. concluded that the effect of colour and material on accuracy is generally less than 1 mm and can often be neglected; however, the depth of reflective surfaces cannot be measured accurately using Kinect v2. Furthermore, in all their experiments, the standard deviation of the samples was consistently equal to SD = 1 mm, indicating that colours and materials have little effect on precision.
Lachat, Macher, Mittet, et al. (2015) found that depth measurements of a compact disk (i.e. a highly reflective surface) parallel to the camera may be overestimated by up to 6 cm. This result is in agreement with the findings reported by Corti et al., obtained measuring the distance of a highly reflective fabric. Furthermore, measurements of the black squares of a checkerboard (Lachat, Macher, Landes, et al. 2015) returned deviations up to 1.2 cm. These surfaces returned very low infrared readings, which caused an overestimation of the depth measurements.

Sarbolandi et al. (2015) positioned a planar checkerboard with varying levels of grey at 1 m of distance from Kinect. The maximum error, obtained as difference between the depth of the darkest area and of the white reference, was 3 mm. Consistently with results reported by Lachat et al., darker regions returned lower infrared intensities, which in turn resulted in larger depth measurements compared to brighter regions.

To quantify the so called amplitude-related distance error, Fursattel et al. (2015) captured and averaged 1000 depth maps of two different targets at 70 cm of distance: first, they attached a white sheet of paper on a flat black background; second, they adopted the same experimental setup, but featuring opposite colours. The aim of this test was to assess the effect of different levels of reflectivity on depth readings, which ideally should be uniform because the two colours lied on the same plane. The first experimental setup provided negligible difference (−1 mm) between the white centre and the black background, and an error of about 1 mm compared to the reference distance. With the second setup, Kinect overestimated the depth of the black central region by approximately 7 mm. These results agree with those reported by Lachat et al. and Sarbolandi et al., and confirm that very dark surfaces should be avoided as target for depth measurement using Kinect v2.

Steward et al. (2015) collected depth maps of a white Spectralon² target with 99% reflectivity and of a black Spectralon target with 5% reflectivity. Distances between Kinect and the target ranged from 1 to 4.5 m, and a step of 0.5 m was used. For each station, points lying on the target were used to calculate a least-squares plane fit. Then, the RMSE of the residuals of the least-square plane fitting were calculated, to evaluate how distance and reflectivity of the object affected the accuracy of depth measurements. For the white target, the RMSE ranged from 2 mm at 1 m to 3.5 mm at 4 m. For the black target, the lower bound was the same, but the RMSE monotonically increased to RMSE = 10 mm at 2.75 m and reached a maximum of RMSE = 23 mm at 4 m. The authors suggested that the optimal distance Kinect-target is between 1 and 2.5 m, because RMSE < 10 mm even using a black target.

Table 6. Summary of the results reported in the reviewed studies about the effect of material types and colours on the accuracy of Kinect v2 depth measurements. Abbreviations: NS = not specified, BG = background, BK = black, W = white, Y = yellow, R = red, B = blue, G = green.

<table>
<thead>
<tr>
<th>Study</th>
<th>Distance Kinect-target (m)</th>
<th>Material</th>
<th>Colour</th>
<th>Depth deviation (mm)</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Lachat, Macher, Mittet, et al. 2015)</td>
<td>NS</td>
<td>Compact disk</td>
<td>(Not reported)</td>
<td>60</td>
</tr>
<tr>
<td>(Lachat, Macher, Landes, et al. 2015)</td>
<td>NS</td>
<td>NS</td>
<td>Black–white</td>
<td>12</td>
</tr>
<tr>
<td>(Corti et al. 2015)</td>
<td>0.775</td>
<td>Cardboard, film, fabric</td>
<td>BK, W, Y, R, B, G</td>
<td>&lt; 1</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Wood–aluminium</td>
<td>NS</td>
<td>0.7</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Highly reflective fabric</td>
<td>NS</td>
<td>Invalid depth</td>
</tr>
</tbody>
</table>

² Spectralon is a fluoropolymer, which has the highest diffuse reflectance of any known material or coating over the ultraviolet, visible, and near-infrared regions of the spectrum. It is a registered trademark of Labsphere (NH, US).
In summary, the results of these studies indicate that black targets should be avoided for Kinect v2 depth measurements, because their poor reflectivity reduces the signal-to-noise ratio of backscattered IR light received by the depth sensor. At short distance (0.7 m), this can cause errors up to 7 mm compared to a coplanar white background. When black areas cannot be avoided, the recommended range of use for Kinect v2 is between 1 and 2.5 m because, within this interval, even black targets achieve RMSE < 10 mm. Reflective surfaces must also be avoided, because they cause either high inaccuracy (up to 60 mm) or invalid depth values. At 0.775 m of distance, non-reflective materials featuring a variety of colours provided negligible differences on accuracy (< 1 mm), suggesting that hue has little effect on depth measurement. This finding is relevant to the development of the novel Kinect coloured marker tracking (KCMT) system presented in Chapter 4 of this thesis.

### 3.2.7 Effect of motion

TOF cameras require several correlation images to reconstruct each depth map (Sarbolandi et al. 2015). Specifically, Kinect v2 requires 9 correlation images for each depth frame. The interval of time required to capture all the correlation images used to generate a single depth frame is called integration time, and is similar to the exposure time of regular cameras (Fursattel et al. 2015). During this time interval, multiple phase measurements are performed for each depth pixel. Then, the depth is calculated.

Moving objects affect the amount of reflected IR light received by a depth pixel during the integration time. Processing the acquired correlation images ignoring motion leads to erroneous distance values, which are more noticeable in proximity of object boundaries.

To simulate a moving target in a controllable way, Sarbolandi et al. (2015) mounted a turning Siemens star on a stepping motor and selected 9 different angular velocities ranging between 0 and 100 rpm. A Siemens star is a device commonly used to evaluate the resolution of optical instruments, printers and displays. It consists of alternated white and black “spokes”. In this experiment, white spokes were alternated with cut-out spokes, to allow visualizing the background. The diameter of the star was 730 mm and each spoke was 15° wide. The aim of this test was to evaluate the percentage of invalid pixels in correspondence of a peripheral arc spanning 90° of the wheel. Invalid pixels were defined as those pixels that cover an inhomogeneous region in terms of depth, thus returning incorrect depth measurements. Kinect automatically labels these pixels as invalid. Ideally, based on the geometry of the star, the resulting data should contain 50% foreground and 50% background pixels. Although in the static condition pixels were almost equally split, 11% of the total number of pixels was marked as invalid. With increasing angular speed, the number of foreground and background pixels linearly decreased and, consequently, invalid pixels linearly increased. Interestingly, there were always more foreground than background pixels, due to the shadowing effect of the spokes on the background. The authors also reported two plots of the depth measurements against the angular coordinate, at 0 and 60 rpm. Invalid points were generally located at the edges of the

<table>
<thead>
<tr>
<th>Study</th>
<th>Reflectivity</th>
<th>Background</th>
<th>RMSE</th>
</tr>
</thead>
<tbody>
<tr>
<td>(Sarbolandi et al. 2015)</td>
<td>1</td>
<td>NS</td>
<td>Black – white</td>
</tr>
<tr>
<td>(Fursattel et al. 2015)</td>
<td>0.7</td>
<td>Paper</td>
<td>White centre – black BG</td>
</tr>
<tr>
<td></td>
<td></td>
<td></td>
<td>Black centre – white BG</td>
</tr>
<tr>
<td>(Steward et al. 2015)</td>
<td>1 – 4</td>
<td>Spectralon, reflectivity</td>
<td>99%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Spectralon, reflectivity</td>
<td>5%</td>
</tr>
<tr>
<td></td>
<td></td>
<td>White</td>
<td>RMSE = 2 – 3.5</td>
</tr>
<tr>
<td></td>
<td></td>
<td>Black</td>
<td>RMSE = 2 – 23</td>
</tr>
</tbody>
</table>
spokes, i.e. in correspondence with the transitions between foreground and background. Although this effect was more evident at 60 rpm, it also affected measurements at 0 rpm. Furthermore, an additional edge effect was identified, causing overshooting (i.e. a few mm overestimation) of depth values at the edges of the foreground spokes.

Fursattel et al. (2015) found that Kinect v2 has a strong internal filter compared to other time-of-flight depth sensor, which drastically reduces measurement noise, even when the scene reflectivity is low. However, the downside of strong filtering is that it affects the temporal resolution of the depth map, reducing the accuracy of measurements of fast-moving objects.

Several motion compensation methods have been proposed, however all of them must be applied to the raw correlation images, which are not accessible in Kinect v2 SDK. Indeed, Kinect directly returns the depth map, and no access is provided to the internal data processing pipeline. Moreover, these correction algorithms are computationally very expensive and would noticeably affect the capture frame rate of the depth stream (Sarbolandi et al. 2015).

To visualize the effect of motion on Kinect-derived measurements, a participant was asked to run on a treadmill at 9 km/h (2.5 m/s). A ø38-mm spherical coloured marker\(^3\) was attached to the participant’s lateral malleolus. Kinect v2 was positioned on a 75-cm high tripod, at 1.2 m from the closer edge of the treadmill. The 3D point cloud (which is derived from the depth map) of the subject’s foot exhibited some artefacts along the direction of Kinect depth axis (Z) during fast motion (Figure 20). Furthermore, the data suggest that the direction of the artefacts may depend on the direction of motion and their magnitude may depend on speed (Figure 20B and Figure 20C).

![Figure 20. Screenshots of Kinect v2 point cloud. Motion affects depth readings, generating artefacts on the surface of the spherical marker and of all moving objects in general. A) Slow speed. B) Moving fast towards the negative X direction. C) Moving fast towards the positive X direction.](image)

The experiments reported in this section indicated that: 1) Kinect v2 depth measurements are affected by the speed of the target, i.e. the number of invalid pixels returned in a depth map is proportional to speed; 2) invalid depth measurements are located in correspondence with boundaries between foreground and background objects; 3) speed causes an overshooting effect, i.e. overestimation of depth at the edges of foreground objects; 4) speed also causes depth artefacts along the Z axis in correspondence with fast moving objects, and the sign of the error may depend on the direction of movement of the object. The effect of motion speed on the accuracy of Kinect v2 depth measurement is quantified in Chapter 9, as part of the analysis of agreement between the proposed coloured marker tracking methodology and a Vicon marker-based motion capture system.

\(^3\) For a description of these coloured spherical markers, see Chapter 4.
3.2.8 Optical lens distortion

When generating a point cloud from a depth map, Kinect v2 SDK applies a correction to account for the geometric distortion introduced by the lens of the depth camera. This correction is based on intrinsic calibration parameters stored in the SDK. According to previous studies, this factory calibration only accounts for the radial component of the optical distortion (Lachat, Macher, Mittet, et al. 2015; Corti et al. 2015), but not for the tangential component. Steward et al. (2015) recommended to perform an individual camera calibration for Kinect v2. This custom calibration process required the calculation of the intrinsic parameters of the camera by recording images of a checkerboard from different points of view.

The most popular computer vision libraries such as OpenCV (Bradski 2000), Emgu CV (Emgu Corporation) or MATLAB Computer Vision System Toolbox include a camera calibration toolbox which could be used to generate a custom calibration for Kinect v2 depth camera. However, for the purposes of this thesis, the measurements provided by the depth camera were combined with the colour information provided by the RGB camera of Kinect v2, based on a per-pixel correspondence defined by the Coordinate Mapper class of the SDK (see section 4.7 Marker detection algorithm for further details). The calibration of either just the depth camera or both the depth and the RGB cameras would have affected the relationship between the pixels of the two imaging sensors. For this reason, it was decided to rely on the factory calibration of Kinect v2, instead of performing a custom calibration to determine the full set of intrinsic parameters of the depth camera.

3.3 Discussion

The present chapter quantitatively evaluated the accuracy and precision of the Kinect v2 depth camera, based on findings from previous studies and results of this thesis. The aim of this review was to identify the sources of error affecting depth data and to quantify their individual effects. The results provided an estimate of the accuracy and precision anticipated for the data used as input by the novel coloured marker tracking methodology presented in Chapter 4.

The following sources of measurement error were identified: 1) internal temperature of the hardware; 2) imperfect generation of the modulated infrared light; 3) distance from the target; 4) optical lens distortion; 5) random electronic noise; 6) colour and reflectivity of the target; 7) motion speed of the target. Although different techniques were adopted to assess these metrological parameters, results were consistent across the reviewed studies. Temperature drift can be practically neutralized observing a warm-up time of 20 min. Similarly, the effect of target colour and reflectivity on accuracy can be minimized (below 1 mm) avoiding black or reflective surfaces, and using Kinect v2 at a range of distances from the target comprised between 1 and 2.5 m. The latter findings have been implemented in the data collection procedures presented in Chapters 8 and 9 of this thesis.

In contrast, some sources of error cannot be avoided. The centre of the sensor featured a systematic wiggling error, which was function of the distance and ranged from ±0.5 cm at 1 m, up to ±1.5 cm at 3 m. Its precision ranged from 1.2 mm at 0.5 m to 4 mm at 4 m of distance from the target.

Accuracy and precision were also a function of the radial coordinate across the sensor area: deviations ranged between −6 and 6 mm at a distance up to 1.25 m from the target. These residuals followed a radial trend and increased with the distance to the target, reaching up to tens of cm at the corners of the sensor at 3 m distance from the target. Similarly, precision
ranged from about 1 mm at 0.8 m, up to 10 mm at 3.75 m (at worst). Borders and corners achieved substantially worse results, with SD up to 40 mm.

The effect of motion on depth measurements hasn’t been thoroughly investigated previously. Some of the studies only performed a qualitative analysis. For this reason, this source of error will be quantitatively assessed in Chapter 9, as part of the analysis of agreement between the novel coloured marker tracking methodology presented in this thesis and a Vicon marker-based motion capture system.

The sources of error identified in this review are reported in Table 7, together with their impact on accuracy and precision, and possible workarounds to minimize their effect. Considering a distance of 2 m was selected between Kinect v2 and a subject, and assuming all the precautions to limit the depth measurement error (Table 7, right column) were observed, a systematic error component of ±10 mm and a random error component of ±5 mm can be anticipated for Kinect v2 depth measurements.

Table 7. Summary of the sources of error affecting Kinect v2 depth measurements, their impact on accuracy and precision, and possible workarounds to minimize their effect

<table>
<thead>
<tr>
<th>Source of error</th>
<th>Error amplitude (mm)</th>
<th>Solution</th>
</tr>
</thead>
<tbody>
<tr>
<td>Internal temperature</td>
<td>5 → ~0</td>
<td>20 min warm-up</td>
</tr>
<tr>
<td>Infrared light modulation</td>
<td>±5 at 1 m</td>
<td>Limit distance</td>
</tr>
<tr>
<td></td>
<td>±15 at 3 m</td>
<td></td>
</tr>
<tr>
<td>Distance from target</td>
<td>( \propto \text{distance}^2 )</td>
<td>Limit distance</td>
</tr>
<tr>
<td>Radial effect</td>
<td>&gt;15 at corners</td>
<td>Avoid corners of sensor</td>
</tr>
<tr>
<td>Electronic noise (centre precision)</td>
<td>1.2 at 0.5 m</td>
<td>Limit distance</td>
</tr>
<tr>
<td></td>
<td>4 at 4 m</td>
<td></td>
</tr>
<tr>
<td>Colour and reflectivity of target</td>
<td>60 → &lt;1</td>
<td>Avoid black or reflective surfaces</td>
</tr>
<tr>
<td>Motion speed of target</td>
<td>Not quantified yet</td>
<td>Limit target speed</td>
</tr>
</tbody>
</table>

Comparing the results reported in this review with those from other studies on the accuracy of Kinect v2 markerless tracking algorithm (see Chapter 9 of this thesis), it can be observed that – even considering the aforementioned sources of error – raw depth measurements remain one order of magnitude more accurate compared to joint coordinates provided by the pose
estimation algorithm, which uses those depth measurements as input. Indeed, Xu & McGorry (2015) compared the Kinect v2 markerless algorithm with a marker-based system and found the accuracy to be posture- and joint-dependent. For an upright standing posture, they found the average error across all participants and joint centres was 87±15 mm. The poorest agreement was at the feet, with an average deviation of 207±71 mm reported for the left foot. Different standing postures yielded even larger deviations. Wang et al. (2015) oriented Kinect v2 in different configurations relative to the participants during both sitting and standing tasks. Considering only standing exercises with Kinect facing participants (a best-case scenario), the mean error of the markerless algorithm for lower body joints ranged between 103 and 146 mm (SD ≈ 30 mm), with ankle and foot kinematics being most inaccurate. Notably, both studies looked at static tasks, and consequently those results were not affected by motion-related error.

This comparison suggests that the degree of inaccuracy observed in joint coordinates provided by the Kinect v2 pose estimation algorithm is largely due to the markerless algorithm itself, rather than due to the raw depth data used as input. This observation provided the major motivation of this thesis, i.e. to develop an alternative motion tracking system for Kinect v2 based on the raw depth measurements, rather than using the off-the-shelf pose estimation algorithm provided by Microsoft. This alternative approach, denoted as Kinect coloured marker tracking (KCMT) system, will be illustrated in Chapter 4 of this thesis.

3.4 References


CHAPTER 4
Tracking coloured markers using Microsoft Kinect v2 and computer vision techniques

4.1 Introduction
This chapter describes the algorithm developed as part of the proposed Kinect coloured marker tracking (KCMT) system. The technical foundations of this algorithm, and the hardware required to use this solution for motion capture are provided. The software application used to track spherical coloured markers in 3D was developed in collaboration with Mr Gino Coates and initially denoted as KinEdge. During the process of writing the validation paper reported in Chapter 9 of this thesis, the combination of the KinEdge software application, the related data capture protocol, and the required hardware were denoted as Kinect coloured marker tracking (KCMT) system. For consistency with that paper, this name was used to indicate also the KinEdge software application in the rest of this thesis.

4.2 Software architecture
The KCMT application was implemented using the .NET Framework (v4.7.1, Microsoft, Redmond, US), Kinect for Windows SDK 2.0 and Emgu CV computer vision library (v3.0.0, Emgu Corporation), which is a C# wrapper of OpenCV (Bradski 2000). Throughout this chapter, the OpenCV library is cited, since the documentation for this library is more detailed compared to that for Emgu CV. Kinect SDK provided access to frame objects, which contained multiple data types: infrared and depth images from the depth sensor (with size 512 × 424 px), and colour images from the RGB camera (1920 × 1080 px). It should be noted that all images provided by Kinect SDK were mirrored along the horizontal direction by default, due to the intended use of Kinect v2 as gaming device. Hence, they were reflected along the same direction before being used as input for the KCMT algorithm.

The proposed application was based on two asynchronous processes: 1) the capture pipeline temporarily stored frame objects from Kinect v2 into a queue, whose maximum size depended on the available random-access memory (RAM); 2) the marker detection pipeline loaded frames from the queue, detected coloured markers and returned their 3D coordinates. The marker detection pipeline performed a coarse marker localization on the RGB image; the result was then refined using the infrared image. This 2-step approach, described in the section 4.7 Marker detection algorithm, was adopted because the RGB images carried colour information required to identify markers, but was more affected by motion blur compared to the infrared images (see section 4.4.1 Factors influencing the selection of HSV tolerances).

Kinect v2 frames were large objects, continuously generated by the SDK during data capture; to ensure the smooth execution of the application, these frames had to be routinely cleared from memory after use. In .NET Framework, memory is automatically managed via Garbage Collection (GC). While smaller objects (classified as generation 0 and 1) are continuously cleared with no significant impact on performance, larger objects (classified as generation 2) are deallocated via...
blocking GCs. During these events, the execution of the program is suspended for a short time, which occasionally may reach up to a few seconds. While necessary to free up memory, blocking GCs significantly affected Kinect v2 capture rate.

If the application has special memory requirements, GC can be adjusted (Microsoft n.d.). Blocking GCs can be temporarily avoided (e.g. during capture), setting the GC policy to Low Latency. Nevertheless, it should be noted that collection of generation 2 objects is performed anyway when a certain threshold of occupied memory is reached, to maintain system stability. In the proposed algorithm, to ensure consistent capture frame rate, a GC was enforced just before recording a new trial and the GC policy was set to Low Latency. When capture was stopped by the user, the GC policy was reset to the default value (Interactive) and another GC was performed to dispose of the remaining large objects.

To speed up the execution of the tracking algorithm, the user could select a region of interest (ROI) in the RGB image before recording, large enough to contain the participant during the task. Pixels outside the ROI were ignored by the algorithm, thus reducing the processing time. If a custom ROI was not selected, the entire RGB frame was considered as ROI by default.

To locate coloured markers in the RGB images, some computer vision techniques were applied, namely image binarisation and contour detection. The following sections discuss the foundations of these techniques. Subsequently, the marker tracking algorithm developed for the KCMT system is described in detail.

4.3 Hue, saturation and value

One of the initial steps in the marker detection algorithm was a coarse localization of the coloured marker in the RGB image provided by Kinect v2. This localization was performed by means of a computer vision technique denoted as contour detection. Contours are curves joining all the adjacent points along a boundary having same colour or intensity. They are used for shape analysis and object detection and recognition from images. To maximize the accuracy of detection of contours, it is common practice to pass binary images as input to the contour detector. For example, in the OpenCV implementation, contours are detected as white objects on a black background. Therefore, objects of interest must be white, whereas the background must be black.

Different criteria can be used to binarise an image, depending on the aim of the analysis. Because the objective of the KCMT system was to track coloured markers, the most effective way to separate them from the background and from each other was to detect their hue. Hue is one of three parameters, the others being saturation and value, which constitute a coordinate system for representing colours. This system, denoted by the acronym HSV, is alternative to the RGB (red, green and blue) colour model. The RGB model is ideal for electronic devices such as displays and projectors, which produce colours by combining red, green and blue light in varying intensities. However, the relationship between the three colour components in the RGB space is not intuitive. The HSV colour model was created as a more convenient way for users to specify colours in software (Smith 1978). This model is also known as HSB (hue, saturation and brightness)\(^4\) and can be represented as a three-dimensional polar coordinate system (Figure 21).

\(^4\)A similar colour model, denoted as HSL (hue, saturation and lightness), was also developed at the same time, but will not be discussed because it is not relevant to this thesis.
The hue coordinate defines colours in terms such as red, yellow, blue-green, etc. In the polar coordinate system, it corresponds to the angular dimension, featuring red primary at 0°, green primary at 120°, blue primary at 240°, and then wrapping back to red at 360°. Saturation measures the departure of a hue from achromatic, i.e. white or grey. Value measures the departure of a hue from black.

Regardless of the system adopted, the colour components of each pixel of an image are stored in memory. The number of bits used to store each colour component of a single pixel is denoted as depth of an image. The depth defines how finely the colour coordinates can be expressed by an image. Kinect v2 colour images use the RGB model and have byte depth. More specifically, the depth is unsigned 8-bit integer (UInt8) (Microsoft n.d.). The range of this integral data type is from 0 to 255 (or $2^8 – 1$).

To convert Kinect images from RGB to HSV, the cvtColor() method from the OpenCV library was used. With this method, saturation and value are represented by integers in the interval from 0 to 255, requiring a byte for each component. To fit the hue domain (0°, 360°) into a byte, the calculated hue of each pixel is divided by 2 before being stored in memory, meaning the domain is halved and ranges from 0 to 180 (OpenCV n.d.).

After conversion to HSV, the image was ready to be binarised. To this end, an HSV filter (based on pre-selected hue, saturation and values ranges) was applied. The following section describes how these HSV ranges were obtained for each marker colour.

### 4.4 HSV filter

To create a binary image of a marker, a filter was applied to the HSV image (Figure 22). This filter contained a specific range for each of the three components: pixels with HSV coordinates within these ranges were replaced by 1 (corresponding to white), whereas the remaining pixels were replaced by 0 (corresponding to black).
A different HSV filter for each marker colour was defined by the user before starting the capture session, using a custom developed tool included in the KCMT software application and denoted as Marker setup tool. The HSV ranges for each filter were calculated using the following relation:

\[
\text{range} = \text{measurement} \pm \text{tolerance}
\]  

where measurement was a generic colour component (H, S or V) measured from an image of the marker, and tolerance was the pre-determined half-width of the interval for that colour component. Tolerances were coordinate-specific, and although they could be manually customized by the user, they were generally left to their default values. These default values were experimentally determined by the author of this thesis, based on many binarisation tests performed using different colours and under a variety of lighting conditions. Measurements, on the other hand, were obtained by the user before starting a capture session, using the aforementioned Marker setup tool. This tool visualized the RGB stream of Kinect v2 on screen. When the user double-clicked on a pixel (which had to belong to a coloured marker), the corresponding HSV coordinates were measured. Then, the appropriate HSV ranges were calculated using Eq. 4.1 and stored into an XML file, together with the corresponding marker names. The XML file could be reused for following capture sessions and adjusted when marker colours or illumination conditions changed.

The identification of optimal HSV tolerances was fundamental to ensure reliable marker detection from colour images. If HSV ranges were too wide, pixels belonging to other objects in the scene could have been binarised to white, causing detection of unwanted contours. On the other hand, if the HSV ranges were too narrow, some (or all) pixels belonging to the marker could have been binarised to black, affecting marker detection and causing gaps in its trajectory. The following section illustrates how such tolerances for the HSV ranges were empirically selected.

### 4.4.1 Factors influencing the selection of HSV tolerances

The selection of suitable HSV tolerances was complicated by several external factors influencing how the coloured surface of a marker was perceived by the RGB camera. These factors were: reflectivity, motion blur and illumination.

A highly reflective marker surface, due for example to a glossy paint, caused significant variations in the colour components (especially value) perceived by the RGB camera across the marker surface. This problem was solved adopting matte acrylic paints (Jo Sonja’s, Chroma Inc.). Fast movements caused motion blur in colour images, making the HSV components of the
affected pixels non-uniform and non-constant. In some camera systems, motion blur can be reduced by selecting a shorter exposure time (i.e. the interval of time during which the sensor is exposed to light for each frame). However, Kinect SDK 2.0 didn’t allow to modify such parameter for the RGB camera, because it was automatically adapted by the firmware based on the average brightness of the scene.

Illumination had a twofold effect on colours. Nonuniform illumination casted shadows on the spherical surface of the marker, i.e. one side appeared darker than another from the camera point of view. In addition, insufficient illumination caused low brightness in the scene. In this condition, the Kinect v2 firmware automatically increased the exposure time of the RGB camera above 30 ms, to increase the amount of light entering the image sensor\(^5\), and its gain above 1x, to amplify the signal and increase the apparent brightness of the image. However, both corrections had side effects, i.e. longer exposure time increased motion blur, while higher gain amplified noise in the image. These camera settings could not be modified via the SDK: the class ColorCameraSettings was provided only to get their current values via the class properties ExposureTime and Gain, not to set them.

To control the illumination during capture, two low-cost photographic lights (each featuring four 115-W compact fluorescent lamps) were positioned at the sides of Kinect and pointed towards the capture volume. Furthermore, to increase the brightness of the scene, white sheets were laid on the ground. With this setup, a uniform illumination, an exposure time of 19 ms and a gain of 1x (corresponding to no amplification) were achieved.

While reflections and illumination could be controlled with a careful choice of the capture conditions, motion blur was only partially reduced by the increased brightness of the scene, i.e. some blur effect was still present in colour images of moving objects. To detect markers during fast movements, HSV ranges had to cater for the variability of colour components due to motion blur. Therefore, optimal tolerance values for each HSV component were determined, compromising between two conflicting factors: maximizing marker detection in presence of motion blur, and excluding unwanted objects from the binary image.

### 4.4.2 Experimental determination of optimal HSV tolerances

The first step was to test whether all three colour components were equally affected by motion blur. Using ImageJ (NIH, US), blurred images of coloured markers (Figure 23) were converted from RGB to HSV and the three components were visualized separately (Figure 24). The hue appeared uniform across the blurred area of the fast-moving marker, meaning this component was not significantly influenced by motion blur. In contrast, saturation and value were nonuniform, indicating these two coordinates were more affected by motion blur compared to hue.

\(^5\) When the scene was particularly dark, Kinect v2 automatically reduced its capture frame rate from 30 to 15 fps, to allow for longer exposure time (up to 66 ms).
To quantitatively evaluate the variation of the HSV components between static and blurred markers, and within each blurred area of a marker, a set of images including either static or moving markers were analysed using ImageJ. First, each image was converted from RGB to HSV. Then, the marker area was manually selected using the freehand tool. For each colour component, mean, standard deviation, maximum and minimum were calculated within the selection.

Based on the results of this analysis, different tolerance values for the HSV ranges were selected and tested in the laboratory, simulating motion capture conditions at different speeds. The resulting binary images were visually evaluated based on two parameters: first, accuracy in the identification of the marker contour; second, number of undesired background pixels binarised to white. The tolerances of the HSV filter which provided the best binarisation results were $\pm 5$ out of 180 for hue, $\pm 50$ out of 255 for saturation and $\pm 75$ out of 255 for value. These values were included as default ranges in the Marker Setup Tool provided in the KCMT software application and were used in all experiments reported in this thesis.

### 4.4.3 Hue range for red primary

Since hue is the angular coordinate in the HSV polar system, the upper and lower limits of its domain are coincident, i.e. the hues at 0° and at 360° are equivalent and correspond to red primary. Because the hue domain in OpenCV is reduced by a factor of two for memory reasons, i.e. $0 \leq H \leq 180$ (see section 4.3 Hue, saturation and value), the same convention was adopted in the rest of this chapter.

The hue range for the HSV filter was initially calculated as (see section 4.4 HSV filter):

$$H_{\text{range}} = H_{\text{measured}} \pm H_{\text{tolerance}}.$$
However, when $H_{\text{measured}} < H_{\text{tolerance}}$ or $H_{\text{measured}} > 180 - H_{\text{tolerance}}$, the limits of the hue range fell outside the hue domain, i.e. below 0 or above 180 respectively. For example, when $H_{\text{measured}} = 3$:

$$H_{\text{range}} = 3 \pm 5 = [-2, 8].$$

Similarly, when $H_{\text{measured}} = 178$:

$$H_{\text{range}} = 178 \pm 5 = [173, 183].$$

Limits such as −2 and 183 were not valid, because outside the hue domain. Since pixels could not have a hue outside the domain, a filter based on such invalid hue limits was more restrictive compared to a filter based on a non-red hue, i.e. some red pixels were converted to black instead of white during the binarisation.

To solve this issue, the hue range was wrapped back in correspondence of 0 and 180. To this end, an algorithm was developed to split the hue range in two sub-ranges for red primary, obtaining a total hue interval the same size as that found for other colours, i.e. twice the tolerance. For example, the [−2, 8] range was transformed into two range checks over both [0, 8] and [178, 180]. Similarly, the [173, 183] range was transformed into two range checks over both [0, 3] and [173, 180].

The algorithm was based on 2 constants, representing the boundaries of the hue domain:

$$HUE_{\text{MIN}} = 0$$
$$HUE_{\text{MAX}} = 180$$

and on 4 functions, representing possible values for the lower and upper limits of the hue range:

$$H_{\text{lower}} = \begin{cases} H_{\text{measured}} - H_{\text{tolerance}} \\ H_{\text{measured}} - H_{\text{tolerance}} + HUE_{\text{MAX}} \end{cases}$$
$$H_{\text{upper}} = \begin{cases} H_{\text{measured}} + H_{\text{tolerance}} \\ H_{\text{measured}} + H_{\text{tolerance}} - HUE_{\text{MAX}} \end{cases}$$

First, a check was performed to verify that $H_{\text{measured}}$ was within the hue domain. Second, between the 2 possible values calculated for each limit of the hue range, those outside the domain were rejected. When $H_{\text{measured}} = H_{\text{tolerance}}$ or $H_{\text{measured}} = HUE_{\text{MAX}} - H_{\text{tolerance}}$, there was an ambiguity for $H_{\text{lower}}$ or $H_{\text{upper}}$ respectively, i.e. there were two values for either limit falling into the domain. In such cases, the ambiguity was resolved using the following criteria:

$$H_{\text{lower}} = \min(H_{\text{lower}})$$
$$H_{\text{upper}} = \max(H_{\text{upper}})$$

For example, if $H_{\text{measured}} = H_{\text{tolerance}} = 5$:

$$H_{\text{lower}} = \min(0, 180) = 0.$$  

Similarly, if $H_{\text{measured}} = HUE_{\text{MAX}} - H_{\text{tolerance}} = 175$:

$$H_{\text{upper}} = \max(180, 0) = 180.$$
Third, the hue range was determined according to the relative magnitudes of the two limits: when $H_{\text{lower}} < H_{\text{upper}}$, the hue range was $H_{\text{lower}} \leq H \leq H_{\text{upper}}$. Otherwise, two sub-ranges were returned: $H_{\text{UEMIN}} \leq H \leq H_{\text{upper}}$ and $H_{\text{lower}} \leq H \leq H_{\text{UEMAX}}$.

4.5 Choosing colours based on HSV ranges

A careful choice of colours was fundamental to improved marker detection using the proposed algorithm. The optimal tolerance for the hue range was identified as $\pm 5$. Therefore, measured marker hues had to differ by at least 10 to avoid obtaining overlapping ranges, otherwise inaccurate binarisation and contour detection could occur.

Saturation measures the departure of a hue from achromatic, i.e. white or grey. Because maximum diversity between all hues was desirable to uniquely identify each marker, maximum distance from a common colour such as white or grey was also beneficial for the same reason. Therefore, highly saturated paints were selected for all marker colours. Similarly, because value measures the departure of a hue from black, this component was maximized too. Moreover, matte paints were selected, to avoid reflections which would have affected the uniformity of the colours perceived by the camera (see section 4.4.1 Factors influencing the selection of HSV tolerances). After an extensive research among commercially available paints and manual testing, a brand of matte acrylic paints was identified to give the best results (Jo Sonja’s, Chroma Inc.). Although the catalogue of this line of paints was quite extensive, not all colours were suitable for marker tracking using Kinect v2 RGB camera.

Hue ranges of all colours tested in this study were plotted on the hue axis (Figure 25), to check for interferences between colours or empty intervals. Each hue was represented by a rectangle, whose left and right sides represented the lower and upper limit of the range, respectively. Solid rectangles represent the 7 colours selected for the KCMT markerset. Empty rectangles represent hues that couldn’t be tracked using Kinect v2.

Both Orange and GreenPthalo could not be used, the former because it was similar to the colour of the skin, the latter because it overlapped by more than 50% with BlueFluo. GreenLight and any violet hue could not be used, due to limitations in Kinect v2 colour camera. Indeed, although during static measurements GreenLight was located in an empty hue interval between Yellow and GreenFluo, during dynamic tests it was perceived as Yellow by Kinect v2 camera, causing interference between the two hues. Similarly, violet hues caused interference with BlueDark.

In contrast, although GreenFluo was partially overlapping with Green, the resulting interference wasn’t significant. To further reduce the effects of hue interference, the KCMT marker detection algorithm was programmed to choose the contour with most white pixels in each binary image (see section 4.7 Marker detection algorithm). In this way, isolated white pixels in the binary image due to suboptimal HSV filtering were ignored.

![Figure 25. Hue ranges selected for KCMT markers (solid rectangles) and other tested colours (empty rectangles)](image-url)
Care also was taken when selecting participants’ clothing. Bright and unsaturated t-shirts and shorts were used (i.e., white or grey), because they didn’t reduce the average brightness of the scene, nor interfere with saturated colours used for markers. Garments were also tight-fitting, to reduce marker occlusions.

4.6 Designing coloured markers
Markers were custom-made using \( \varnothing 38 \text{ mm} \) white polystyrene foam balls, commonly available at art and craft shops. Smaller markers were also tested, but they couldn’t be reliably tracked during dynamic tasks. Raw spheres were painted using the identified matte acrylic paints (Jo Sonja’s, Chroma Inc., Figure 26a).

![Figure 26. Toolkit for marker design: (a) coloured markers and corresponding paint tubes; (b) clear round surface savers were used as bases for coloured markers, to protect the thin layer of paint; (c) water-based construction glue used to attach the bases to the coloured markers.](image)

Markers were attached to the participants’ skin using double-sided sticky tape. To allow reusing markers and avoid damaging the thin layer of paint, a clear circular plastic base (\( \varnothing 19 \times 1 \text{ mm} \)) was attached to each marker, as interface with the sticky tape (Figure 26b). Most glues contain solvents that can melt polystyrene foam. To quickly and strongly hold markers and bases together without dissolving the foam, a thick water-based construction glue was used (Figure 26c). To improve adhesion, a thin spherical cap (about 2 mm in height) of the marker was cut, obtaining a flat surface on which attaching the base.

4.7 Marker detection algorithm
After establishing the technical foundations of colour detection and associated challenges, the proposed marker detection algorithm may be described in detail. A simplified flowchart with pictures illustrating the essential steps of the algorithm is reported in Figure 27, whereas a complete flowchart is reported in Figure 28.
4.7.1 Marker detection using colour, infrared and depth data

First, a smoothing median filter (5 × 5 kernel) was applied to the colour image within the ROI to increase colours uniformity (Figure 28). The smoothed ROI was converted from RGB to HSV colour space and then binarised: pixels with HSV levels within the ranges defined for the current marker to be tracked were set to white, while the rest were set to black. The binary ROI was then dilated using a 3 × 3 kernel, to fill small holes in white regions.

Extreme outer contours were detected in the binary ROI and approximated via polygons to reduce the number of points, using an accuracy of 0.005 times the length of their perimeter. Area and minimum enclosing circle (MEC) were computed for each approximated contour. A bounding box (BB) was obtained for each MEC and the white pixels contained in each BB were counted. Using a facility of Kinect SDK called Coordinate mapper (Microsoft 2014a), the position of each MEC centre was mapped from the RGB image to Kinect v2 local coordinate system (camera space), obtaining its X, Y and Z coordinates. Kinect reference frame is defined as follows: Z is the depth coordinate (positive outwards), Y is the vertical coordinate (positive upwards) and X is obtained via the right-hand rule.

To represent a valid marker candidate, each contour had to fulfil the following requirements:

\[
10 \leq \text{Area} \leq 1800 \text{ px}^2 \\
Z \leq 4 \text{ m}
\]

The ranges for Area and Z were determined empirically, to avoid confusion between markers and background objects with similar colours. Among all valid contours, the one with most white pixel was selected as candidate marker.

The BB of the candidate contour was mapped from the colour image to the infrared image and was denoted as infrared ROI (IRROI). To ensure that the IRROI fully contained the marker, it was inflated by a factor of 0.35 × min(BBwidth, BBheight). The IRROI (which contained a greyscale image) was compressed from ushort to byte format using linear interpolation. The interval from 10th to 90th percentile of pixel intensities was used as range for linear interpolation, to disregard extremely dark or bright pixels and have larger greyscale resolution between these limits. The IRROI was then smoothed using a Gaussian filter with 3 × 3 kernel.
To detect the circle corresponding to the marker, the Hough Circle Transform (HCT) was applied to the IRROI. The HCT is a feature extraction technique for detecting circular objects in a digital image. The OpenCV implementation of the HCT includes a preliminary edge detection step, based on a Canny filter. The threshold to be used by this filter was determined using Otsu’s algorithm (Otsu 1979). The following parameters were selected for the HCT: minimum radius = 3 pixels; maximum radius = minimum between width and height of the IRROI; accumulator threshold = 13; minimum distance = 100 pixels. The noise and low resolution of the greyscale image contained in the IRROI affected the results of the HCT. To improve the circle detection reliability, the HCT was applied recursively, varying the accumulator resolution from 1 to 6 by increments of 1. The roundest circle among those detected by each HCT iteration was added to a list of circles. The coordinates of the centre of the optimal circle were estimated as median values of the coordinates of the centres of all circles in the list. The centre of this optimal circle was mapped from the infrared image to camera space, obtaining the 3D position vector of the centre of the marker spherical surface. To obtain the coordinates of the inner centre of the marker, the position vector was extended by the known marker radius (see section 4.7.2 Marker centre correction).

The process described above was repeated for all marker colours defined in the XML file, before being repeated for the next frame. Since Kinect v2 has an irregular framerate (= 30 fps), frame timestamps could not be generated assuming a constant time interval between frames. Hence, marker coordinates obtained from a frame object were associated with the timestamp of the corresponding depth image. Three-dimensional coordinates were rounded to the 4th decimal place and exported to a .trc (track row column) file, which is compatible with OpenSim (Stanford, US) musculoskeletal modelling software and any spreadsheet application.
Figure 28. Flowchart of the developed marker detection algorithm, based on colour, infrared and depth data from Kinect v2
4.7.2 Marker centre correction

Since Kinect v2 is a depth camera, it measures the 3D coordinates of points belonging to the surface of the objects in front of it. However, with spherical markers, obtaining the coordinates of the inner centre is more desirable than measuring the coordinates of the centre of its surface facing Kinect. To this end, a correction was applied to marker coordinates provided by the KCMT system, based on the known marker radius $r$.

The following two approximations were made for this calculation: first, the optical aberration introduced by Kinect depth camera was ignored; second, the rectangular depth sensor was approximated by a point, meaning distortions due to its finite sizes were ignored. Under such approximations, it could be assumed that Kinect depth camera produced undistorted depth and infrared images of the objects in front of it. Therefore, a spherical marker projected onto the sensor plane appeared always as a disk, independently of its 3D position relative to the origin $O$ of Kinect coordinate system. Consequently, this 3D problem could be represented as a 2D problem (Figure 29).

![Figure 29](image)

*Figure 29. The centre "C" of the spherical marker was obtained extending the vector "OP" by the marker radius $r"*

Kinect coordinate system is represented by the axes $x$, $y$ and $z$ centred in $O$. The circle with centre $C$ and radius $r$ is the projection of a spherical marker onto the plane of Figure 29. The segment $\overline{AB}$ is the projection of the circle (i.e. the marker) onto an imaginary plane parallel to Kinect depth sensor. $P$ is the centre of $\overline{AB}$ and the point of tangency between $\overline{AB}$ and the circle. Note that the tangent line to a circle at a point is perpendicular to the radius to that point, consequently: $\overline{AB} \perp \overline{PC}$.

Based on the assumptions reported above, a spherical marker projected onto the sensor plane appears always as a disk, independently of its position with respect to $O$. Consequently, the projection $\overline{AB}$ of the circle is always perpendicular to the vector $\overline{OP}$ connecting Kinect origin to the centre of $\overline{AB}$, i.e. $\overline{AB} \perp \overline{OP}$.

Because $\overline{AB} \perp \overline{PC}$ and $\overline{AB} \perp \overline{OP}$, and considering that $\overline{OP}$ and $\overline{PC}$ have $P$ in common, $\overline{OP}$ and $\overline{PC}$ are collinear. $\overline{OP}$ is measured by Kinect via the proposed marker detection algorithm. Because $\overline{OP}$ and $\overline{PC}$ are collinear, and because the length of $\overline{PC}$ is the known radius $r$ of the marker, the position vector $\overline{OC}$ can be determined extending $\overline{OP}$ by the radius. The above relation can be described by the following equations:

$$\overline{OC} = \overline{OP} + \overline{PC}$$
Thus:

\[ \overrightarrow{OC} = \overrightarrow{OP} + \frac{\overrightarrow{OP}}{\|\overrightarrow{OP}\|} r. \]

Thus:

\[ \overrightarrow{OC} = \overrightarrow{OP} \left( 1 + \frac{r}{\|\overrightarrow{OP}\|} \right) \]

where \( \overrightarrow{OC} \) is the desired position vector of the marker centre \( C \) expressed in Kinect coordinate system. The latter quantity, i.e. the centre of the spherical coloured marker, is the outcome measurement expected from the proposed KCMT system.

4.8 Conclusion

In conclusion, a novel methodology to track coloured markers in 3D space was described in this chapter. This approach combined colour, infrared, depth and point cloud data streams provided by the affordable Microsoft Kinect v2, and used computer vision techniques to extract 3D trajectories of the centre of coloured spherical markers. The marker detection algorithm was discussed in detail, with focus on the challenges of binarising colour images to reliably isolate the markers of interest. Furthermore, the design of the custom-made coloured markers was illustrated, including description of the hardware selected to assemble optimal markers for this application.

Some limitations of this methodology ought to be considered: homogeneous illumination and sufficient brightness are required in the capture volume to reduce shadows (which may affect perception of marker colours by the Kinect RGB camera) and motion blur (which may reduce the tracking accuracy during fast motions). In addition, the maximum number of coloured markers that can be tracked at the same time is currently limited to 7, due to incapacity of the Kinect v2 RGB camera to discern a greater number of different hues.

While a 12-camera Vicon motion capture system similar to that used as gold standard in Chapters 2, 8 and 9 of this thesis costs approximately AU$ 250,000 (Thewlis et al. 2013), the proposed KCMT system was built using an AU$ 200 Kinect v2 sensor and freely available software libraries. Thanks to the affordability of Kinect v2 and of the hardware required to create custom coloured markers, the KCMT system may have application for on-site analysis of lower limb biomechanics in facilities such as schools, gyms and clinics, where more expensive and cumbersome multi-camera optical motion capture systems may not be convenient.

Validation studies for the proposed methodology are reported in Chapters 8 and 9 of this thesis, where this approach is compared against a multi-camera Vicon motion capture system, respectively using single-leg squat and treadmill locomotion as benchmark tasks.

4.9 References


OpenCV, cvtColor. Available at: http://docs.opencv.org/2.4/modules/imgproc/doc/miscellaneous_transformations.html#cvtcolor.


CHAPTER 5
Coordinate transformation algorithms

5.1 Abstract
Each motion capture system measures the coordinates of body markers in a specific reference frame. However, to compare measurements of the same participant concurrently recorded using two different motion capture systems, all coordinates must be expressed in the same reference frame. To this end, data expressed in one reference frame must be transformed to the other. The aim of the present chapter is to describe two different algorithms developed to transform coordinates of body markers between Kinect v2 and Vicon reference frames. Such methods were used in the validation studies of the novel Kinect coloured marker tracking (KCMT) system, reported in Chapter 8 and 9 of this thesis.

5.2 Introduction
The purpose of a motion capture system is to measure the coordinates of markers placed on a participant’s body. Such coordinates are expressed in a coordinate system (i.e. they are measured with respect to a 3D frame of reference), whose origin is located somewhere in the motion capture lab. Each motion capture system has its own frame of reference; some systems also offer the user a procedure to customise it before recording trials. When two (or more) systems are used to concurrently capture motion data, it is often necessary to transform all measurements to a common coordinate system, in order to correctly compare and evaluate their 3D marker trajectories. This coordinate transformation may be directed towards the reference frame of one of the motion capture systems, or towards a completely new reference frame. Furthermore, this operation can happen before exporting data from the motion capture software (i.e. in real-time), or after (i.e. during post-processing), depending on the specific system configuration and on the selected data analysis pipeline.

In this thesis, motion data were concurrently captured using two different methods: the novel Kinect coloured marker tracking (KCMT) system presented in Chapter 4, and a conventional Vicon multi-camera optical system. In the validation studies of KCMT reported in Chapters 8 and 9, there were two reasons to transform Kinect- and Vicon-derived coordinates. First, to perform inverse kinematics and inverse dynamics analyses using OpenSim (Stanford, US), based on marker trajectories measured by the two motion tracking systems and on ground reaction forces measured by a force plate; this study, in which participants performed single-leg squat (SLS) trials, is presented in Chapter 8 of this thesis. Second, to assess the agreement between 3D marker trajectories measured by the two methods; this study, in which participants walked and ran on a treadmill, is reported in Chapter 9 of this thesis. For simplicity, hereinafter the two studies will be denoted as Single-leg squat study and Treadmill study, based on their respective trials type.

Coordinate transformation is typically a 2-step process: first, the pose of one reference frame must be fully determined with respect to the other reference frame. This process involves finding the orientation of the coordinate axes and the position of the origin. Second, the coordinates expressed in one reference frame are transformed to the other reference frame: this is achieved by applying a rotation and a translation to the input coordinates.
The aim of the present chapter is to illustrate the methods used to transform coordinates in the two studies reported in this thesis. Hereinafter, the conventions, hardware and algorithms used to determine the pose of one reference frame with respect to the other are described. Furthermore, the equations used to transform marker coordinates from the original to the destination reference frame are presented. Lastly, the linear algebra concepts and algorithms used to derive the coordinate transformation equations are provided in section 5.4 Appendix.

5.3 Methods

5.3.1 Kinect v2 coordinate system

The Kinect v2 coordinate system, also referred to as camera space (Microsoft 2014a), has the origin located at the centre of the depth sensor and the axes oriented as follows: Z is normal to the depth sensor plane and directed outwards, along the optical axis of the depth camera; Y is directed upwards; X follows the right-hand rule (Figure 30). It should be noted that the exact position of the origin with respect to the external case of Kinect was not provided by Microsoft, and may be measured only by disassembling the device.

Because Kinect is based on a depth sensor, its primary output is represented by the depth – i.e. the Z coordinate – of the objects in its field of view. The X and Y coordinates of each 3D point are calculated by the Kinect SDK, based on geometrical equations and on the Z coordinate of each corresponding depth pixel. Therefore, it is reasonable to hypothesise that different levels of accuracy exist for measurements along the three coordinate axes. One of the aims of the Treadmill study was to test this hypothesis and quantify the accuracy based on a Bland-Altman analysis. For this reason, it was decided to transform Vicon coordinates to Kinect space, so that the agreement results could be reported in the latter reference frame. Using this approach, specific conclusions could be drawn for each axis of the Kinect v2 coordinate system.

5.3.2 Vicon coordinate system

The methodology developed by Vicon to calibrate its multicamera systems is denoted as DynaCal. This calibration is usually repeated at the beginning of each motion capture session and is composed of two distinct phases (Woolard 1999; Vicon n.d.; Vicon 2007). The first phase is a dynamic calibration, which requires the user to wave a wand throughout the whole capture volume. This step allows the system to calculate the positions and orientations of all the cameras relative to each other, and results in accurate 3D reconstructions of the 2D marker trajectories seen by each camera. However, the resulting 3D trajectories need to be measured relative to a user-defined coordinate system in which, for example, the vertical axis is the direction of gravity and the other two axes are in the floor plane (Woolard 1999). To this end, a second phase denoted as static calibration is performed, in which the user puts a static calibration object on the ground. The Vicon software application measures the position and orientation of this object and aligns the global coordinate system to it.
To be visible by the Vicon system, both the wand and the static calibration object have markers attached in known positions. Indeed, both the dynamic and static stages of the system calibration process depend on the Vicon application software knowing the dimensions and relative marker positions of these calibration objects (Vicon 2007).

**Ergocal**

Over time, Vicon has developed different static calibration objects. The *Ergocal static calibration object* (Figure 31) is commonly used in biomechanics because it allows the user to align the Vicon coordinate system to a corner of a force plate. In the SLS study, to position the origin of the Vicon coordinate system near the centre of the capture volume, the Ergocal was aligned to one of the corners of the 2nd force plate (Figure 32).

![Figure 31](image1.png)

*Figure 31. The coordinate system of the Vicon Ergocal static calibration object is centred on its inferior 90° corner. Source (Vicon 2007).*

![Figure 32](image2.png)

*Figure 32. The Ergocal used in the motion analysis laboratory was aligned to one of the corners of the 2nd force plate.*

The Ergocal features four ø9.5-mm markers attached on its top surface; these can be replaced by ø14- or ø25-mm markers, as long as they are all the same size (Vicon 2007). The coordinate axes are aligned with the edges of the object, and the origin corresponds to its inferior 90° corner. The Ergocal is provided with 2 alignment plates, used to accurately align its edges to those of a force plate. The object is also provided with adjustment screws and spirit levels, to ensure it is level on the floor. It should be noted that the contact points between the Ergocal and the floor were represented by 3 conic tips, sharp enough to pierce through the carpet and thus ensure maximum stability of the object on the metal surface of the force plate.
Vicon static calibration algorithm

All Vicon static calibration objects have 4 markers (Woolard 1999). Three of these markers are in a straight line, while the fourth, denoted as singleton, is away from this line. Two of the three markers in the straight line are significantly closer together to one another than to the third, making it possible to spatially identify all four markers uniquely.

The static calibration algorithm performs the following operations (Figure 33): first, it records 20 samples (frames) of the static calibration object (Woolard 1999; Vicon n.d.). The object must be in view of at least three cameras (Vicon 2007). Second, the known coordinates of the Ergocal markers expressed in its local coordinate system are read from a marker definition file (.vsk in Vicon Nexus software (Vicon 2007)), and the reconstructed points are matched uniquely to them (Woolard 1999). Third, a line of best fit is found through the three aligned 3D points. Very commonly, although not essentially, this line is defined in the .vsk file to be in the direction of one of the desired co-ordinate system axes. Fourth, the vector which passes through the singleton and is perpendicular to the best-fit line through the three aligned points is then found. If the alignment of the three points is set parallel to an axis, this perpendicular vector must lie in the plane of the other two axes. Again, very commonly but not essentially, this vector is defined in the .vsk file to be in the direction of one of the two remaining co-ordinate system axes. At this point, the algorithm uses the right-hand rule to determine the third axis (Vicon 2007). Lastly, the coordinate system origin is moved until the three coordinates of the singleton defined in the .vsk file match those measured by the system (Woolard 1999). By the end of this step, both the orientation and position of the desired coordinate system are matched to the camera parameters and the calibration is complete.

Figure 33. A flowchart of the Vicon static calibration algorithm (Woolard 1999; Vicon n.d.; Vicon 2007)
The above plain-language description of the static calibration algorithm is based on information provided by Vicon in their official manuals. Below, the algorithm is illustrated in terms of linear algebra relations and sketches. These mathematical steps were not provided by Vicon but were deducted as part of the present thesis to align different coordinate systems. It was assumed that the 4 Ergocal markers were positioned in an L-shape, whose arms were parallel to the short edges of the object and to the two horizontal axes of its local coordinate system defined in the .vsk file. This assumption was verified based on measurements taken on the Ergocal using a digital caliper. Although not essential (Woolard 1999), this condition simplifies the following demonstration.

Two coordinate systems were defined (Figure 34). \((O - xyz)_1\) is the Vicon coordinate system before the static calibration, but after the dynamic calibration, when the relative poses of all cameras have been determined. The pose of this system is unknown to the user and not required for calculations. \((O - xyz)_2\) is the Vicon coordinate system after the static calibration. This system corresponds to the local coordinate system of the Ergocal when in position for the static calibration phase.

The orientation of \((O - xyz)_2\) with respect to \((O - xyz)_1\) was determined as follows: first, a line was fitted through the 3 aligned markers (see section 5.4.4 Line of best fit for 3 or more 3D points). From its direction, the unit vector \(x_2\) was obtained. Second, the shortest vector perpendicular to this line and passing through the singleton marker was determined (see section 5.4.5 Shortest vector perpendicular to a line and passing through a point). From this second direction, the corresponding unit vector \(y_2\) was obtained. Third, the unit vector \(z_2\) was determined via the right-hand rule.

In the rest of this chapter, all vectors indicated by a lower-case bold italic letter (e.g. \(v\)) were intended as column vectors, unless otherwise specified.

The vectors \(x_2, y_2\) and \(z_2\) represented the columns of the rotation matrix \(R_2\), describing the orientation of \((O - xyz)_2\) with respect to \((O - xyz)_1\):

\[
R_2 = \begin{bmatrix}
\vdots & \vdots & \vdots \\
x_2 & y_2 & z_2 \\
\vdots & \vdots & \vdots \\
\end{bmatrix}^{3x3}
\]

Figure 34. Determination of the rotation matrix of \((O - xyz)_2\) with respect to \((O - xyz)_1\)
To determine the position vector $d_{12}$ of $O_2$ measured in $(O - xyz)_1$, the position of the singleton marker in both coordinate systems was used (Figure 35), based on the following relation:

$$s_1 = R_2 s_2 + d_{12}$$

which can be re-written as:

$$d_{12} = s_1 - R_2 s_2.$$ 

After calculating $R_2$ and $d_{12}$, the pose of $(O - xyz)_2$ with respect to $(O - xyz)_1$ was fully determined. These parameters are used by the Vicon system to transform the position $r$ of a generic marker $P$ from $(O - xyz)_1$ to $(O - xyz)_2$ (Figure 36). The known marker position $r_1$ can be expressed as follows:

$$r_1 = R_2 r_2 + d_{12}$$

where $r_2$ is the unknown position of the same maker in the calibrated system.
Considering the properties of the rotation matrices \( R_2^T R_2 = I \), the following relations can be derived:

\[
R_2^T r_1 = R_2^T R_2 r_2 + R_2^T d_{12}
\]

\[
r_2 = R_2^T r_1 - R_2^T d_{12}
\]

\[
r_2 = R_2^T (r_1 - d_{12})
\]

where:

- \( r_1 \) is a generic marker tracked by Vicon and whose coordinates are expressed in the uncalibrated coordinate system, and \( r_2 \) is the same marker, transformed to the calibrated reference frame.

The algorithm described above was used as a model to implement the coordinate transformation procedures both for the Single-leg squat study and the Treadmill study. These procedures are discussed in detail in the following sections.

5.3.3 Coordinate transformation in the Single-leg squat study

To perform inverse dynamics analyses, marker trajectories and ground reaction forces are required as input data. The latter are initially expressed in the coordinate system of the force plates, which is generally different from the reference frame of the motion capture system (e.g. Vicon). However, when a motion capture system is installed into a laboratory, the parameters required to transform from force plates to Vicon coordinate system are determined and stored into the motion capture software application (e.g. Vicon Nexus). Consequently, measurements from force plates are always exported using the same coordinate system as the marker trajectories.

In the Single-leg squat study, inverse dynamics analyses were performed using data recorded with both the KCMT system and Vicon. In this study, Kinect-derived coordinates were transformed to Vicon space. This approach was preferred compared to the opposite (i.e. transforming from Vicon to Kinect) for two reasons: firstly, in this way there was no need to
transform the ground reaction forces, as they were already expressed in the Vicon coordinate system. Secondly, to have consistent data visualization between the virtual workspace of Vicon Nexus software, and the transformed Kinect marker trajectories.

To determine the pose of the Vicon coordinate system in Kinect space, an L-frame featuring 4 coloured markers was built (Figure 37). This L-frame mimicked the Vicon Ergocal calibration object. A MATLAB (R2016a, MathWorks, Natick, MA, US) algorithm similar to the Vicon static calibration was implemented, to be used in conjunction with the L-frame.

Pose estimation hardware: L-frame

The L-frame was built using a bar of dressed pine, with rectangular section of 64 x 19 mm. The bar was cut in two parts of 50 and 40 cm of length respectively, which were connected by flat, L-shaped metal plates and nails. To obtain a 90° angle, a metal square was used as guide while connecting the two parts (Figure 37). Two rectangular metal plates were attached to the outer edges of the L-frame, to ensure accurate alignment to the force plate, and thus to the Vicon coordinate system. The arms of the L-frame were parallel to the horizontal axes of the Vicon coordinate system: specifically, the long arm was parallel to the X-axis and the short arm was parallel to the Y-axis (Figure 38).

Figure 37. The custom L-frame was made of wood and its two arms were assembled using nails and metal plates. A square (left picture) was used to obtain a 90° angle. In this figure, top view (left) and bottom view (right) are reported.

Figure 38. (Left) The L-frame positioned on the corner of the 2nd force plate of the motion analysis laboratory. Four coloured calibration markers were attached. (Right) The arms of the L-frame were parallel to Vicon horizontal axes, and the inferior corner corresponded to the origin.

Four coloured markers (⌀20 mm) were attached in known positions of the L-frame coordinate system. Due to human error during the assembling process, these theoretical positions were not

---

6 The coordinates described in this subsection were stored in a function denoted as calibration_local_coordinates.m.
fully achieved. The actual positions were then measured using a digital caliper, and included as constants into the coordinate transformation algorithm, to compensate for such error (Table 8).

<table>
<thead>
<tr>
<th>Marker</th>
<th>Theoretical coordinates</th>
<th>Actual coordinates</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>x (m)</td>
<td>y (m)</td>
</tr>
<tr>
<td>GreenFluo</td>
<td>0.020</td>
<td>0.020</td>
</tr>
<tr>
<td>BlueFluo</td>
<td>0.350</td>
<td>0.020</td>
</tr>
<tr>
<td>Red</td>
<td>0.450</td>
<td>0.020</td>
</tr>
<tr>
<td>Yellow</td>
<td>0.020</td>
<td>0.400</td>
</tr>
</tbody>
</table>

The reported coordinates represented the centres of the 4 calibration markers. The 37-mm measured for the z-coordinate were due, from bottom-up, to the carpet lying on top of the force plate (5 mm), the head of the nails (1 mm), the metal plate joints (1 mm), the wooden bar (19 mm), the transparent marker supports (1 mm) and the marker radius (10 mm). In contrast with the Ergocal, the L-frame had no sharp tips as points of contact with the floor. Therefore, the L-frame rested on the carpet, instead of being in direct contact with the force plate. For this reason, the thickness of the carpet was taken into account in the z-coordinate of the calibration markers.

As with the Ergocal, three markers of the L-frame were aligned and parallel to Vicon x-axis. Furthermore, two markers were aligned and parallel to the y-axis. This alignment was not required for a pose estimation algorithm based on Vicon static calibration, but was included for future use with alternative algorithms. The first step of the pose estimation procedure consisted of recording a static trial using the proposed KCMT system: during this trial, the L-frame was positioned on the 2nd force plate and within Kinect field of view. From this trial, the coordinates of the coloured calibration markers in Kinect space were obtained. These were used to determine the pose of the Vicon reference frame in Kinect space, using the algorithm illustrated in the next section. After this step, Kinect could not be moved for the duration of the motion capture session; otherwise, a new calibration trial had to be performed.

**Pose estimation algorithm**

To determine the pose of the L-Frame in Kinect space, an algorithm based on Vicon static calibration was implemented. The Vicon X-axis was determined as the line of best fit through the 3 aligned markers of the L-frame (see section 5.4.4 Line of best fit for 3 or more 3D points). The Y-axis was found as the shortest vector perpendicular to this line, passing through the singleton marker (see section 5.4.5 Shortest vector perpendicular to a line and passing through a point). The Z-axis was found via the right-hand rule, as the cross product of X and Y.

Once the 3 vectors representing the directions of Vicon axes were determined, they were normalized to obtain the unit vectors $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$. Then, they were arranged as column vectors, to obtain the rotation matrix $\mathbf{R}_v$ of the Vicon coordinate system expressed in Kinect space:

---

7 The function containing this algorithm was denoted as `pose_estimation_vicon_in_kinect.m`. 

99
Because \( \mathbf{v}_1, \mathbf{v}_2, \) and \( \mathbf{v}_3 \) were orthogonal unit vectors, this matrix was orthogonal\(^8\). However, to ensure accuracy of the results, the following checks were performed on this matrix by the algorithm:

- it was verified that \( \text{rank}(\mathbf{R}_v) = 3 \), to ensure the three column vectors were independent;
- it was verified that \( \det(\mathbf{R}_v) = 1 \), to ensure that \( \mathbf{R}_v \) was a rotation matrix\(^9\);
- it was verified that all column vectors \( \mathbf{v} \) were unit vectors, allowing a tolerance equal to \( \pm \) the machine precision (\( \varepsilon = 2 \cdot 10^{-16} \)):
  \[
  1 - \varepsilon \leq \| \mathbf{v} \| \leq 1 + \varepsilon;
  \]
- it was verified that the orthonormality error was of the same order of magnitude of the machine precision. This check was based on the property of orthogonal matrices \( (\mathbf{R}_v^T \mathbf{R}_v = \mathbf{I}) \), and the error was calculated using the Frobenius norm\(^10\):
  \[
  \text{err} = \| \mathbf{I} - \mathbf{R}_v^T \mathbf{R}_v \|_{\text{Fro}} \leq 10 \cdot \varepsilon.
  \]

To determine the position vector \( \mathbf{d}_{kv} \) of Vicon origin in Kinect space, the centroid of the 3 aligned markers measured in the L-frame space was matched with the position of the same point in Kinect space. The centroid of the 3 aligned markers was used as a virtual marker instead of the singleton marker adopted in the Vicon static calibration for two reasons: first, to spread the inevitable measurement error across multiple markers, instead of just one. Second, because the three aligned points were located at shorter distance from Kinect, compared to the singleton. Indeed, Kinect accuracy decreases when distance from the sensor increases (see Chapter 3).

The actual coordinates of the 3 aligned markers expressed in the L-frame reference system (Table 8) were arranged as columns of the matrix \( \mathbf{A}_v \). The L-frame coordinate system was aligned to the Vicon coordinate system when the L-frame was in position for calibration:

\[
\mathbf{A}_v = \begin{bmatrix}
  x_1 & x_2 & x_3 \\
  y_1 & y_2 & y_3 \\
  z_1 & z_2 & z_3
\end{bmatrix}
\]

The centroid of such points was calculated as the mean of \( \mathbf{A}_v \) along the row direction:

\[
\mathbf{c}_v = \begin{bmatrix}
  \bar{x} \\
  \bar{y} \\
  \bar{z}
\end{bmatrix}
\]

\(^8\) For a definition of orthogonal matrix, see section 5.4.1 Linear algebra.
\(^9\) For a definition of rotation matrix, see section 5.4.1 Linear algebra.
\(^10\) See section Norms of a matrix in the Appendix.
Similarly, the coordinates of the three aligned calibration markers measured by Kinect were arranged as columns of the matrix $A_k$:

$$
A_k = \begin{bmatrix}
X_1 & X_2 & X_3 \\
Y_1 & Y_2 & Y_3 \\
Z_1 & Z_2 & Z_3 
\end{bmatrix}
$$

The centroid of such points was given by the mean of $A_k$ along the row direction:

$$
c_k = \begin{bmatrix}
\bar{X} \\
\bar{Y} \\
\bar{Z}
\end{bmatrix}
$$

The position vector $d_{kv}$ of Vicon origin measured in Kinect space was determined using the following relation between the position of the centroid $C$ expressed in the two coordinate systems (Figure 39):

$$
d_{kv} = c_k - R_{cv}c_v
$$

At this stage, the pose of the Vicon coordinate system in Kinect space was fully determined. These parameters were used to transform coordinates from Kinect to Vicon space, using the method described in the next section.

Coordinate transformation from Kinect to Vicon space

Once the pose of Vicon frame in Kinect space was known, marker trajectories were transformed from Kinect to Vicon space. The position vector of a generic point $P$ in $\mathbb{R}^3$ was denoted as $r_k$ and $r_v$ when expressed in Kinect and Vicon coordinate systems, respectively (Figure 40).
The following relation held:

\[ r_k = R_v r_v + d_{kv} \]

where \( R_v \) and \( d_{kv} \) represented the known orientation and position of the Vicon reference frame in Kinect space. The previous relation was re-arranged as:

\[ R_v r_v = r_k - d_{kv}. \]

Then, using the property of orthogonal matrices \( (R_v^T R_v = I) \), the following relation was obtained:

\[ r_v = R_v^T (r_k - d_{kv}). \]

The known position of the generic point \( P \) expressed in the Kinect coordinate system \( (r_k) \) was transformed to the Vicon space \( (r_v) \).

**Transformation to OpenSim coordinate system**

OpenSim (Stanford, US) is an open-source software application, used in the SLS study to perform inverse kinematics and inverse dynamics analyses. This computer program uses a so-called *model coordinate system*. When the musculoskeletal model is in its default position (Figure 41), the global OpenSim axes are directed as follows (Lund & Hicks n.d.):

- the Y axis is directed upwards;
- the X axis is directed along the postero-anterior direction;
- the Z axis is directed along the medio-lateral direction and follows the right-hand rule.
OpenSim documentation states that it is the user’s responsibility to transform motion capture data from the laboratory coordinate system (i.e. the Vicon space in this thesis) to the OpenSim reference frame. It also states that any arbitrary coordinate system may be defined.

Using the approach described in the previous section, Kinect-derived marker trajectories were initially transformed to the Vicon reference frame. Then, they were further rotated to match the OpenSim axes convention described above (see function `ViconToOpenSimRotation.m`). The rotation from Vicon to OpenSim frame was composed of two consecutive 90° rotations (Figure 42): the first rotation was around the X coordinate axis, the second rotation was around the Y axis. The resulting overall rotation matrix was obtained by pre-pending the second rotation to the first:

\[
R_{V\to OS} = R_y^{-90°} R_x^{-90°}.
\]

Vicon-derived marker trajectories and ground reaction forces were also transformed to the OpenSim reference frame. However, this transformation was carried out as part of the conversion from .c3d, which is the file format used by Vicon for motion data, to .trc and .mot,
which are the file formats used by OpenSim for marker trajectories and ground reaction forces respectively. This transformation was performed via the *MATLAB OpenSim Toolbox v2* (Lichtwark et al. 2014) (see function `btk_c3d2trc.m`). In this case, the rotation was obtained reordering the axes from XYZ to YZX. This operation was equivalent to multiplying the coordinates by the rotation matrix $R_{V\text{to}OS}$ reported above.

As a result of these transformations, the musculoskeletal model animated with data collected during single-leg squat trials was displayed in the correct upright position (Figure 43).

![Figure 43. After the coordinate transformation from the Vicon to the OpenSim coordinate system, the model animated with SLS data was displayed in the correct upright position.](image)

### 5.3.4 Coordinate transformation in the Treadmill Study

In the Single-leg squat study, the pose of the Vicon reference frame was estimated in the Kinect space. In contrast, in the Treadmill study, the Kinect pose was estimated in the Vicon space. This section illustrates the methodology used in the latter case. It should be noted that the pose estimation algorithms implemented in the two cases are very similar, based on the fact that they were both inspired by Vicon static calibration.

**Pose estimation hardware: reflective markers and 3D-printed array**

To determine the orientation of the Kinect reference frame in Vicon space ($R_K$), 4 reflective markers were attached on the top surface of Kinect (Figure 44), forming an L-shape. The 3 aligned markers were parallel to the X axis, whereas the singleton marker was offset towards the positive Z direction.
The estimation of the orientation based on the 3D positions of manually-placed markers is affected by human error. A number of strategies were implemented to minimize this error. First, the edges of the device were used as reference to position the markers. Second, although 3 points are sufficient to determine the 3D pose of an object, 4 markers were used in this study to minimize the pose estimation error. Third, the distance between these markers was maximized as much as possible, considering the limited area provided by the Kinect top surface. Fourth, relatively small reflective markers were used (⌀9 mm).

Because the exact location of the Kinect origin within the external case of the device was not known (see sub-section 5.3.1 Kinect v2 coordinate system), another point was required to determine the position vector $d_{vk}$ of Kinect origin in Vicon space. The coordinates of this point had to be measurable by both systems. For this reason, a 3-marker array was 3D-printed and positioned in proximity of the centre of the capture volume (Figure 45). This array was symmetrical, featuring two ⌀9-mm reflective markers at its extremities and a ⌀38-mm coloured marker at the midpoint between them. The centre of this array could be tracked by both Vicon and the KCMT system, and was used in the following pose estimation algorithm to determine the position vector $d_{vk}$.

**Pose estimation algorithm**

The steps of the pose estimation algorithm were as follows: first, the x-axis was determined as the line of best fit through the 3 aligned markers (see section 5.4.4 Line of best fit for 3 or more 3D points). Second, the z-axis was determined as the shortest vector perpendicular to this line,
passing through the singleton marker (see section 5.4.5 Shortest vector perpendicular to a line and passing through a point). Third, the y-axis was obtained as the cross product of z and x, according to the right-hand rule.

Once the 3 vectors representing the directions of Kinect axes were determined, they were normalized to obtain the unit vectors $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_3$. Then, they were arranged as column vectors, to obtain the rotation matrix $R_k$ of Kinect coordinate system expressed in Vicon space:

$$R_k = \begin{bmatrix}
\vdots & \vdots & \vdots \\
\mathbf{v}_1 & \mathbf{v}_2 & \mathbf{v}_3 \\
\vdots & \vdots & \vdots
\end{bmatrix}$$

Because $\mathbf{v}_1, \mathbf{v}_2$ and $\mathbf{v}_3$ were orthogonal unit vectors, this matrix was orthogonal\textsuperscript{12}. However, the following checks were programmatically performed anyway on this matrix:

- it was verified that $\text{rank}(R_k) = 3$, to ensure the three column vectors were independent;
- it was verified that $\det(R_k) = 1$, to ensure that $R_k$ was a rotation matrix\textsuperscript{13};
- it was verified that all column vectors $\mathbf{v}$ were unit vectors, allowing a tolerance equal to the machine precision ($\varepsilon = 2 \cdot 10^{-16}$):

$$1 - \varepsilon \leq \|\mathbf{v}\| \leq 1 + \varepsilon;$$

- it was verified that the orthonormality error was of the same order of magnitude of the machine precision. This check was based on the property of orthogonal matrices $(R_k^T R_k = I)$, and the error was calculated using the Frobenius norm\textsuperscript{14}:

$$\text{err} = \|I - R_k^T R_k\|_{\text{Fro}} \leq 10 \cdot \varepsilon.$$  

To determine the position vector $d_{vk}$, the 3D-printed marker array was positioned in proximity of the centre of the capture volume. The centre of this array was denoted as $P$ (Figure 46).

\textsuperscript{12} For a definition of orthogonal matrix, see section 5.4.1 Linear algebra.

\textsuperscript{13} For a definition of rotation matrix, see section 5.4.1 Linear algebra.

\textsuperscript{14} See section Norms of a matrix in the Appendix.
Considering the position vector $P$ expressed in Vicon ($r_v$) and in Kinect space ($r_k$), the following relation was used (Figure 46):

$$d_{vk} + R_k r_k = r_v.$$  

In this equation, all variables were known except $d_{vk}$; therefore, the relation was re-arranged to find the unknown vector:

$$d_{vk} = r_v - R_k r_k.$$  

The advantage of the method presented in this section compared to the one used in the SLS study was twofold: the accuracy of the pose estimation was greater due to the use of Vicon instead of Kinect-derived measurements; in addition, the present method didn’t require the use of a custom-made L-frame. However, this method had also some limitations: the small area provided by the Kinect top surface limited the distance between the 4 reflective markers. Therefore, even a small error in their coordinates could have had significant impact on the accuracy of the pose estimation and, consequently, of the coordinate transformation. Furthermore, because the position of Kinect origin in Kinect space was not known, an extra step involving the use of a 3D-printed marker array was required to determine the position vector $d_{vk}$.

Coordinate transformation from Vicon to Kinect space

Once the pose of the Kinect reference frame in Vicon space was fully determined, marker trajectories measured by the Vicon system were transformed to Kinect space\(^\text{15}\). The position vector of a generic point $P$ in $\mathbb{R}^3$ was denoted as $r_k$ and $r_v$ when expressed in Kinect and Vicon coordinate systems, respectively (Figure 46).

The following relation was used:

$$r_v = R_k r_k + d_{vk}$$

\(^{15}\text{It should be noted that no ground reaction forces were measured in the Treadmill study.}\)
where \( R_k \) and \( d_{vk} \) represented the known orientation and position of Kinect reference frame in Vicon space. The previous relation was re-arranged as:

\[
R_k r_k = r_v - d_{vk}.
\]

Using the property of orthogonal matrices \( (R_k^T R_k = I) \), the following relation was obtained:

\[
r_k = R_k^T (r_v - d_{vk}).
\]

The known position of the generic point \( P \) expressed in Vicon coordinate system \( (r_v) \) was transformed to Kinect space \( (r_k) \).

5.3.5 Transforming multiple markers at the same time

The last relation reported in the previous section transforms the coordinates of a single point from Vicon to Kinect space, knowing the pose of the Kinect coordinate system in Vicon space. However, typically multiple points must be transformed at each instant of time. One option was to write the previous transformation within two nested for-loops: first, looping across all available points (i.e. markers) in the inner loop; second, looping across all instants of time (i.e. frames) in the outer loop.

A better alternative, used in this thesis, was to remove the loop across all markers, taking advantage of MATLAB vectorization capabilities (MATLAB 2017). This approach made the code significantly shorter (and therefore less error prone) and improved its computational performance.

For this demonstration, it was assumed that the objective was to transform all Vicon-derived marker coordinates to the Kinect space, as discussed in the previous section. Assuming the markerset was composed of \( n \) points, all their coordinates measured by Vicon at a generic frame were initially stored in a .trc file, where they were concatenated in a single, long row vector \( r_v \):

\[
r_v = [r_{v1x} \ r_{v1y} \ r_{v1z} \ r_{v2x} \ r_{v2y} \ r_{v2z} \ ... \ r_{vnx} \ r_{vny} \ r_{vnz}].
\]

Using MATLAB \textit{reshape} function, this row vector was re-arranged as a \( 3 \times n \) matrix \( P_v \), where each column contained the coordinates of a marker:

\[
P_v = \begin{bmatrix}
\vdots & \cdots & \vdots \\
r_{v1} & \cdots & r_{vn} \\
\vdots & \cdots & \vdots 
\end{bmatrix}_{3 \times n}.
\]

The transformation of a single point reported in the previous section was modified, to transform all markers at a generic frame in one step:

\[
P_k = R_k^T (P_v - d_{vk})
\]

where:

\[
P_k = \begin{bmatrix}
\vdots & \cdots & \vdots \\
r_{k1} & \cdots & r_{kn} \\
\vdots & \cdots & \vdots 
\end{bmatrix}_{3 \times n}
\]

\footnote{The function containing this algorithm was denoted as \textit{transform_vicon_coords_to_kinect_with_refl_markers.m}.}
was a $3 \times n$ matrix whose columns represented the unknown coordinates of the $n$ markers transformed to the Kinect space at a generic frame; $R_k$ was the known $3 \times 3$ rotation matrix representing the orientation of Kinect reference frame in Vicon space; and:

$$P_v - d_vk = \begin{bmatrix} \vdots & \cdots & \vdots \\ r_{v1} - d_{vk} & \cdots & r_{vn} - d_{vk} \\ \vdots & \cdots & \vdots \end{bmatrix}_{3 \times n}$$

was a $3 \times n$ matrix whose columns contained the known differences between the coordinates of the $n$ markers in Vicon space at the generic frame, and the position vector of Kinect reference frame in Vicon space ($d_vk$).

Lastly, the matrix $P_k$ containing the transformed points was re-arranged back to a single row vector $r_k$, using the reshape function:

$$r_k = [r_{k1x} \quad r_{k1y} \quad r_{k1z}, \quad r_{k2x} \quad r_{k2y} \quad r_{k2z}; \quad \ldots \quad r_{knx} \quad r_{kny} \quad r_{knz}]$$

so that the resulting coordinates could be stored back into a .trc file. This process was repeated for each frame via a single for-loop. With this approach based on MATLAB vectorization capabilities, a second (nested) for-loop across the $n$ markers was avoided. The same principle was applied to the opposite transformation, i.e. from Kinect to Vicon coordinate system, in the SLS study\textsuperscript{17}.

\textsuperscript{17} The function containing this algorithm was denoted as \texttt{transform_kinect_coordinates_to_vicon.m}
5.4 Appendix
5.4.1 Linear algebra
Definitions (Savov n.d.):

**Linear transformation**: product of a matrix and a column vector, producing a transformed column vector, which can have different dimension from the input vector.

**Vector space**: a set of vectors and all linear combinations of these vectors. For example, the vector space \( S = \text{span}(v_1, v_2) \) consists of all vectors of the form \( v = \alpha v_1 + \beta v_2 \), where \( \alpha \) and \( \beta \) are real numbers.

**Columns space of a matrix**: set of vectors that can be produced as linear combinations of the columns of the matrix.

**Row space of a matrix**: set of vectors that can be produced as linear combinations of the rows of the matrix.

**Range**: column space of a matrix.

**Basis**: any set of vectors that can be used as a coordinate system for a vector space. A basis for a \( n \)-dimensional vector space \( S \) is any set of \( n \) linearly independent vectors that are part of \( S \).

**Dimension of a vector space**: number of vectors in a basis for that vector space.

**Rank**: dimension of the vector space spanned by the columns of a matrix. This is the same as the dimension of the space spanned by its rows.

**Matrix inverse** \( (A^{-1}) \): undoes the effect of a matrix \( A \).

**Determinant**: a special way to combine the entries of a matrix that serves to check if a matrix is invertible or not.

**Invertible matrix**: if \( \det(A) \neq 0 \), then \( A \) is invertible. \( A \) corresponds to a linear transformation which maps the \( n \)-dimensional input to an \( n \)-dimensional output, such that there exists an inverse transformation \( A^{-1} \) that can faithfully undo the effects of \( A \).

**Singular matrix**: if "det" (\( C \)) = 0 then \( C \) is not invertible and is defined singular. There is no \( C^{-1} \) that can undo the effects of \( C \).

**Orthonormal basis**: basis of a space whose vectors are all unit vectors and orthogonal to each other.

**Orthogonal matrix** \( (Q) \): a square matrix with real entries whose columns and rows are orthogonal unit vectors (i.e. orthonormal vectors). For this matrix the following relation holds: \( Q^T Q = QQ^T = I \), where \( I \) is the identity matrix. Equivalently, \( Q^T = Q^{-1} \). NOTICE: these matrices are called orthogonal, but their vectors must be always orthonormal (orthogonal and normalized). The lack of normality of \( Q \) would change the norm of the input vector when \( Q \) is applied to it.
Rotation matrix: a square matrix with real entries. More specifically, an orthogonal matrix with determinant 1.

If "det" ($A$) = ±1, the matrix is orthogonal.

If "det" ($A$) = 1, the orthogonal matrix $A$ is a rotation matrix. This is the type of matrix of interest for the present thesis.

If "det" ($A$) = −1, this orthogonal matrix $A$ is a reflection matrix, which is not useful for the purposes of this thesis.

5.4.2 Norms of a matrix
Given a generic $A_{m \times n}$ matrix, there are several norms that can be calculated from it (Lyche n.d.; Lange 2010; Stensby n.d.):

$L_1$ norm: the maximum column sum of absolute values of $A$. The absolute values down each column are added, and then the largest result is selected:

$$\|A\|_1 = \max_{1 < j < n} \left( \sum_{i=1}^{m} |a_{ij}| \right)$$

$L_\infty$ norm: the maximum row sum of absolute values of $A$. The absolute values along each row are added, and then the largest result is selected:

$$\|A\|_\infty = \max_{1 < i < m} \left( \sum_{j=1}^{n} |a_{ij}| \right)$$

Frobenius norm: this type of matrix norm has the useful property of being invariant under rotations. It can be defined in various ways:

$$\|A\|_F = \sqrt{\sum_{i=1}^{m} \sum_{j=1}^{n} |a_{ij}|^2} = \sqrt{\text{trace}(A^T A)} = \sqrt{\sum_{i=1}^{\min(m,n)} \sigma_i^2}$$

where $\sigma_i$ are the singular values of $A$.

$L_2$ norm: also called the spectral norm, it is the largest singular value of $A$:

$$\|A\|_2 = \sigma_{\max}(A)$$

5.4.3 Singular value decomposition
The singular value decomposition (SVD) is a factorization of a matrix $M \in \mathbb{R}_{m \times n}$ in the form:

$$M_{m \times n} = U_{m \times m} \Sigma_{m \times n} V_{n \times n}^T$$

where:

$U$ is an orthogonal matrix, $\Sigma$ is a diagonal matrix whose elements are the singular values of $M$ (usually arranged in descending order) and $V$ is an orthogonal matrix. Since $U$ and $V$ are orthogonal, they (and their transposes) can be regarded as orthonormal bases.

The $m$ columns of $U$ and the $n$ columns of $V$ are called the left-singular vectors and right-singular vectors of $M$ respectively.

In the special, yet common case when $M$ is a rotation matrix, $U$ and $V$ can be viewed as rotation matrices as well, while $\Sigma$ can be regarded as a scaling matrix. Thus, the expression $U \Sigma V^T$ can be intuitively interpreted as a composition of a rotation, a scaling and another rotation.
It is very uncommon for the full SVD to be required. It is often sufficient, faster and more economical for storage to compute a reduced version of the SVD. There are different types of reduced SVD, but the most widely used is the thin SVD (also known as economy size or just reduced SVD). The thin SVD is defined as follows:

\[ M_{m \times n} = U_{m \times n} \Sigma_{n \times n} V_{n \times n}^T \]

where only the \( n \) column vectors of \( U \) corresponding to the row vectors of \( V^T \) are calculated. This is particularly convenient if \( n \ll m \).

### 5.4.4 Line of best fit for 3 or more 3D points\(^ {18} \)

Given a set of \( n \) 3D points, we want to determine the line of best fit for these points (see (User85109 n.d.) for implementation and (George n.d.) only for theoretical reference). Hence, we define a \( n \times 3 \) matrix containing the 3D coordinates of the points on which the line must be fitted:

\[ P = \begin{bmatrix}
    x_1 & y_1 & z_1 \\
    x_2 & y_2 & z_2 \\
    \vdots & \vdots & \vdots \\
    x_n & y_n & z_n 
\end{bmatrix}_{n \times 3} \]

The coordinates of the centroid of the point cloud are calculated as mean of the columns of \( P \):

\[ v_0 = \begin{bmatrix}
    \bar{x} \\
    \bar{y} \\
    \bar{z}
\end{bmatrix} \]

Then we subtract \( v_0^T \) to each row vector of \( P \) and store the results in \( A_{n \times 3} \):

\[ A = \begin{bmatrix}
    x_1 - \bar{x} & y_1 - \bar{y} & z_1 - \bar{z} \\
    x_2 - \bar{x} & y_2 - \bar{y} & z_2 - \bar{z} \\
    \vdots & \vdots & \vdots \\
    x_n - \bar{x} & y_n - \bar{y} & z_n - \bar{z} 
\end{bmatrix}_{n \times 3} \]

Now we calculate the SVD (see section 5.4.3 Singular value decomposition) of \( A \):

\[ A = U \Sigma V \]

The right-singular vector which corresponds to the largest singular value of \( A \) represents the direction of the line of best fit. Since the MATLAB implementation of the SVD sorts the singular values in decreasing order, the vector we are interested in is the first column of \( V \):

\[ v = V(:,1) \]

The resulting line of best fit is:

\[ v(t) = tv + v_0 \]

Any point \( v(t) \) on this line can be obtained changing the scalar \( t \).

To quantify the accuracy of fit, the distances between the fitted line and the input points in \( P \) can be calculated (see (Onlinemschool.com n.d.; Tzeng 2011) for a geometric proof and (Weisstein n.d.) for an analytical proof). Given a generic point \( p_i \) from the input matrix \( P \):

\(^{18}\) The function containing this algorithm was denoted as `FitLine3D.m`. 

112
\[ p_i = P(i,:)^T \]

The distance between this point and the line of best fit is:

\[ d_i = \frac{|(v_0 - p_i) \times v|}{|v|} \]

Once the distance is calculated for all points in \( P \) and the results are stored in an array \( d \), the mean and standard deviation of this array can be calculated to quantify the accuracy of fit of the line.

5.4.5 Shortest vector perpendicular to a line and passing through a point\(^{19}\)

Given a line passing through the points \( A \) and \( C \), we want to find the shortest perpendicular vector from that line to another point \( B \). It should be noted that there are infinite vectors which are perpendicular to a line and pass through another point outside this line: indeed, to be perpendicular, the vector does not need to intersect the first line. To univocally define the perpendicular vector, this must also be the shortest.

Let us define \( P \) the unknown intersection point between the line \( AC \) and the perpendicular line through \( B \) (Figure 47). The equation of the line through \( AC \) is:

\[ P = A + \gamma (C - A) \]

where \( \gamma \) is an unknown scalar.

---

\(^{19}\) The function containing this algorithm was denoted as `perpendicular_line_one_point.m`. The deduction of this algorithm was found online (Benproductions1 n.d.), but included an error in the second last equation, which causes the vector to have opposite direction. The author of this thesis submitted a request to correct this equation. The deduction reported in this thesis and the corresponding MATLAB function are correct.
Because the dot product between two perpendicular vectors is zero, the unknown vector \( V = B - P \) perpendicular to \( AC \) can be obtained from the following relation:

\[
(B - P) \cdot (C - A) = 0
\]

At this point, the first equation can be substituted into the second, obtaining:

\[
\left( B - (A + \gamma (C - A)) \right) \cdot (C - A) = 0
\]

\[
\left( B - A - \gamma (C - A) \right) \cdot (C - A)
\]

Performing the dot-product one obtains:

\[
(B \cdot x - A \cdot x - \gamma (C \cdot x - A \cdot x))(C \cdot x - A \cdot x) + \\
+ (B \cdot y - A \cdot y - \gamma (C \cdot y - A \cdot y))(C \cdot y - A \cdot y) + \\
+ (B \cdot z - A \cdot z - \gamma (C \cdot z - A \cdot z))(C \cdot z - A \cdot z) = 0
\]

Substituting these two new variables:

\[
D = C - A \\
O = B - A
\]

and solving for \( \gamma \), the following relation can be obtained:

\[
(O \cdot x - \gamma D \cdot x)D \cdot x + (O \cdot y - \gamma D \cdot y)D \cdot y + (O \cdot z - \gamma D \cdot z)D \cdot z = 0
\]

\[
O \cdot x \cdot D \cdot x - \gamma D \cdot x^2 + O \cdot y \cdot D \cdot y - \gamma D \cdot y^2 + O \cdot z \cdot D \cdot z - \gamma D \cdot z^2 = 0
\]

\[
\gamma D \cdot x^2 + \gamma D \cdot y^2 + \gamma D \cdot z^2 = O \cdot x \cdot D \cdot x + O \cdot y \cdot D \cdot y + O \cdot z \cdot D \cdot z
\]

\[
\gamma (D \cdot D) = O \cdot D
\]

\[
\gamma = \frac{O \cdot D}{D \cdot D}
\]

Once \( \gamma \) is known, one can determine both the closest point \( P \) on the line and the unknown vector \( V \) as:

\[
P = A + \gamma D \\
V = B - P
\]

### 5.5 References

Benproductions1, How to find the vector that passes through a point and is perpendicular to another vector. Available at: http://answers.unity3d.com/questions/535822/how-to-find-the-vector-that-passes-through-a-point.html.


Lund, K. & Hicks, J., Coordinates and utilities. Available at: http://simtk-
confluence.stanford.edu:8080/display/OpenSim/Coordinates+and+Utilities.

Lyche, T., Matrix norms. , pp.1–27.


Onlinemschool.com, Distance from a point to a line - 3-Dimensional. Available at: http://onlinemschool.com/math/library/analyticgeometry/p_line/.


Stensby, J., Chapter 4: Matrix Norms. ReCALL, pp.1–16.


Vicon, 2007. Vicon MX hardware - System reference - Revision 1.6,

Vicon, Vicon user manual - Preparation.


CHAPTER 6
A novel iterative synchronization algorithm based on cross-covariance

6.1 Abstract
This chapter discusses the problem of synchronizing paired motion capture trials; that is, data recorded using two different motion capture systems (e.g. Kinect v2 and Vicon) concurrently. To this end, a novel iterative algorithm was developed, using cross-covariance to determine the lag between two corresponding trials. A motion trial is generally composed of multiple signals. Depending on the trial type, these signals can be either X, Y and Z coordinates of reflective marker trajectories, or joint angles.

The proposed algorithm computed a tentative shift between each pair of signals constituting the trials to be synchronized. The tentative shift that minimized the SD of the differences between measurements from the two trials was selected as the final time lag to be used to synchronize them.

In this chapter, the proposed algorithm is described in detail, including the input and output variables, the implemented mathematical principles and the underlying assumptions. This algorithm was then employed as part of the validation studies of the novel marker tracking methodology presented in Chapter 4. Results and discussion of these studies are reported in Chapters 8 and 9 of this thesis.

6.2 Introduction
To evaluate the agreement between measurements provided by two different human motion capture systems, their respective signals must be synchronized. These signals are typically represented by 3D coordinates (X, Y, and Z) of reflective markers attached to the participant’s body, or by joint angles, and are stored in motion trial files.

Different approaches can be adopted to synchronize human motion trial data concurrently recorded by two motion capture systems. These approaches can be based on hardware, software or a combination of both. A hardware approach is to establish a network connection between the two computers running the two motion capture systems. Once the two machines have joined the network, their internal clocks can be synchronized to a common absolute time. Timestamps provided by the two synchronized clocks can be attached to the recorded motion trials and used to temporally align them. A combined approach is to use a single computer, powerful enough to concurrently run the software applications controlling both motion capture systems. In this case, both systems share the same clock, and timestamps can again be used for synchronization. Lastly, a software-based synchronization can be achieved using signal post-processing techniques, based on the maximization of an objective function such as cross-correlation or cross-covariance.

At the time of data collection for the validation studies reported in Chapters 8 and 9 of this thesis, the Kinect-based software application presented in Chapter 4 was at an early stage of development. For this reason, the implementation of networking features wasn’t a priority in
the development process, making the first synchronization approach not applicable. Furthermore, because the applications controlling the Kinect and Vicon systems were highly demanding in terms of computational power and bandwidth, executing both at the same time on a single computer may have affected the frequency of data collection. For this reason, the second approach was also deemed not suitable. Consequently, the third method of synchronization (i.e. signal post-processing) was selected for this study. This chapter presents a novel synchronization algorithm, developed to temporally align Vicon- and Kinect-derived motion capture trials. This algorithm was applied to synchronize trials used in the Bland-Altman analyses of agreement of single-leg squats (Chapter 8 of this thesis) and of treadmill walk and run (Chapter 9).

It should be noted that MATLAB (R2016a, MathWorks, Natick, MA, US) features an algorithm called *alignsignals*. As reported in MATLAB documentation, this algorithm uses the estimated delay $D$ to delay the earliest signal such that the two signals have the same starting point. However, this built-in MATLAB function has two important limitations. First, the delay in the earliest signal is achieved by prepending $D$ zeros to the signal itself. Since the purpose of this study is to use synchronized signals as input for a Bland-Altman analysis of agreement, zeros cannot be added, because they would affect the agreement between the two measurement systems. This MATLAB function has also an option to preserve the original length of the delayed signal, truncating its last $D$ samples. This option simply truncates the end of the signal by the same number of samples added at the start, and doesn’t solve the issue of having extra zeros at the start of the signal. Second, *alignsignals* uses cross-correlation to estimate the delay: this approach was found to be ineffective for the type of signals used in the present study, because the estimation of the delay was often highly inaccurate (see subsection 6.3.2 Cross-covariance).

The proposed synchronization algorithm has the following benefits: first, it doesn’t add any additional values (e.g. zeros) to the signal in order to achieve a shift. Second, it truncates both paired trials to the same length, with minimum loss of information. Third, it achieves highly accurate synchronization results based on an iterative approach using cross-covariance. Fourth, it is valid for two different types of motion data, i.e. 3D marker coordinates (X, Y and Z) and joint angles. In the following sections, the underlying assumptions and the mathematical principles supporting the proposed synchronization algorithm were described, as well as its input and output variables.

### 6.3 Methods

#### 6.3.1 Assumptions

The proposed algorithm was designed to synchronize time series from two different human motion capture systems, such as marker trajectories or joint angles. In this context, the following assumptions were made about such motion data. First, paired signals to be synchronized represented the same physical variable, for example *the X coordinate of the marker attached to the lateral malleolus*, or *the knee flexion angle*. Second, signals had same sampling rate, which could be either native or achieved via interpolation. Third, if the two signals had different lengths (expressed as number of samples, or frames), the shortest signal was *complete*, i.e. it contained data pertaining to the entire movement to be analysed. Consequently, after synchronization, the longer signal could be truncated to match the short one, without significant loss of information.
6.3.2 Cross-covariance

To understand cross-covariance, it is worth introducing other similar functions first, namely auto-correlation, auto-covariance and cross-correlation. Given a sample sequence $x(n)$, the auto-correlation (ACR) function is the average product of the sequence $x(n)$ with a time shifted version of itself (Hinton 2002):

$$ACR(m) = E\{x(n) \cdot x(n + m)\}$$

where $E$ denotes mathematical expectation and $m$ is the shift. Similarly, auto-covariance (ACV) is the average product of the two versions of the sequence, but with the mean value, $\bar{x}(n)$, removed:

$$ACV(m) = E\{[x(n) - \bar{x}(n)] \cdot [x(n + m) - \bar{x}(n)]\}.$$ 

In the case of a random variable having zero mean, the ACR and ACV functions are identical.

When these functions are calculated for different shift values $m$, it is possible to plot their trend against the shift itself. It is understandable that when the shift is zero, positive peaks of the signal align with positive peaks, and negative peaks with negative peaks, giving a large positive product. Therefore, we expect the ACR function to be large and positive for $m = 0$. On the other hand, introducing a time shift $m \neq 0$, i.e. sliding one version of the signal with respect to the other, the positive and negative peaks begin to lose their alignment, thus reducing the ACR.

With random sequences, i.e. those where successive values may be assumed as being independent, ACR returns a large central peak at $m = 0$, with small non-zero values either side. Indeed, at zero time shift, positive and negative values align with themselves giving a large positive ACR. However, for any other time shift, a given sample is just as likely to align with one of the opposite sign, so the cross-product averages out to zero. In other words, there is no correlation between adjacent samples. This type of sequence is more commonly referred to as white noise (Mancini 2002).

Cross-correlation (CCR) is essentially the same process as ACR, but instead of comparing a sequence with a time shifted version of itself, it compares two different sequences. The CCR of two sequences $x(n)$ and $y(n)$ is defined as:

$$CCR(m) = E\{x(n) \cdot y(n + m)\}.$$ 

Cross-covariance (CCV) is the same as CCR, except that the mean values of the two sequences are removed:

$$CCV(m) = E\{[x(n) - \bar{x}(n)] \cdot [y(n + m) - \bar{y}(n)]\}.$$ 

CCR and CCV can be used to determine timing differences between two sequences. For example, if $x(n)$ and $y(n)$ are identical white noise sequences which differ only in the time origin, then their CCR (or CCV) will be zero for all values of $m$, except the one which corresponds to the timing difference. Once this shift value is known, it can be used to synchronize the two input signals.
The shift which provides the maximum CCR (or CCV) is the estimate of the point in time when the signals are best aligned. Therefore, this shift is determined by the argument of the maximum\(^{20}\) of the CCR (or CCV)\(^{21}\):

\[
s = \text{arg max}(\text{CCR}(x, y))
\]

\[
s = \text{arg max}(\text{CCV}(x, y))
\]  \hfill (6.1)

Both CCR and CCV are available as functions in MATLAB and are named `xcorr` and `xcov`, respectively. These functions apply different lags to the second signal (\(y\)) with respect to the first signal (\(x\)), and for each lag calculate the corresponding CCR or CCV, respectively:

\[
[\text{CCR}, \text{lags}] = \text{xcorr}(x, y)
\]

\[
[\text{CCV}, \text{lags}] = \text{xcov}(x, y).
\]

The lag which maximizes the function can be imagined as the number of positions by which the second signal (\(y\)) must be shifted to be synchronized with the first signal (\(x\)). Therefore, hereinafter this lag will be indicated as \(\text{shift}\).

To visualize the concepts explained above, an example provided by the MATLAB documentation on cross-covariance can be used (MathWorks 2018). Let \(s_1\) be a random signal and \(s_2\) a copy of it, but delayed by 50 samples with respect to \(s_1\) (Figure 48). To obtain this delay, \(s_1\) was circularly shifted\(^{22}\) forwards by 50 samples, and stored as \(s_2\). The delay introduced in \(s_2\) was represented by a red vertical line.

---

\(^{20}\) In mathematics, the argument of the maximum (or \(\text{arg max}\)) is the point of the domain of a function at which the function is maximized.

\(^{21}\) Because CCR and CCV are two different functions, the respective resulting shifts may differ.

\(^{22}\) When an array is circularly shifted using MATLAB function \textit{circshift}, all its elements are moved forwards by as many positions as the requested shift \(s\). The \(s\) number of elements overflowing at the end of the array are prepended to the array, so that the output array has the same length as the input one. If the requested shift \(s\) is negative, all elements are shifted backwards, and the \(s\) overflowing elements at the start of the array are appended to its end.
The delay, that in real-life scenarios is unknown, can be determined by calculating the CCV between the two signals (Figure 49). As expected, the maximum of the CCV is found in correspondence of −50, i.e. s2 must be shifted backwards by 50 samples to be aligned with s1.

In the present thesis, CCV was preferred to CCR based on experimental findings. The synchronization algorithm was tested twice on treadmill gait trials captured using the Kinect and Vicon systems, with the only difference of using CCR the first time and CCV the second. Data were captured according to the protocol described in Chapter 9 of this thesis. Out of 9 pairs of signals, CCV returned the best estimate of the shift (i.e. 55 samples) 5 times, versus 4 times returned by CCR (Table 9). Furthermore, when the best shift wasn’t identified, CCV still provided
a more accurate result compared to CCR, as demonstrated by the mean and SD of the shifts (54±2 samples for CCV, vs. 119±174 samples for CCR). Comparing these synchronization results, it was evident that CCV provided more accurate and precise lag estimation across multiple pairs of signals.

Table 9. Shifts calculated using CCR and CCV from treadmill gait data. CCV provided more accurate and precise lag estimation across all pairs of signals. Starred values indicate the best estimate of the shift. All shifts are expressed in number of samples.

<table>
<thead>
<tr>
<th>Signal</th>
<th>Shift from CCR</th>
<th>Shift from CCV</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hip X</td>
<td>-20</td>
<td>52</td>
</tr>
<tr>
<td>Hip Y</td>
<td>422</td>
<td>56</td>
</tr>
<tr>
<td>Hip Z</td>
<td>-20</td>
<td>52</td>
</tr>
<tr>
<td>Knee X</td>
<td>55*</td>
<td>55*</td>
</tr>
<tr>
<td>Knee Y</td>
<td>421</td>
<td>55*</td>
</tr>
<tr>
<td>Knee Z</td>
<td>48</td>
<td>49</td>
</tr>
<tr>
<td>Ankle X</td>
<td>55*</td>
<td>55*</td>
</tr>
<tr>
<td>Ankle Y</td>
<td>55*</td>
<td>55*</td>
</tr>
<tr>
<td>Ankle Z</td>
<td>55*</td>
<td>55*</td>
</tr>
<tr>
<td>Mean</td>
<td>119</td>
<td>54</td>
</tr>
<tr>
<td>SD</td>
<td>174</td>
<td>2</td>
</tr>
</tbody>
</table>

6.3.3 Input variables

The proposed algorithm was implemented in MATLAB and required several input arguments. First, the two motion capture trials to be synchronized. Second, a list of the names of the signals to be used for synchronizing the two trials. In this study, all signals available in the input motion trials were used for synchronization. However, in some instances, specific signals may need to be excluded, for example when they are not available in both motion trials to be synchronized, or when they are known to be affected by errors (such as temporary marker occlusion).

The third input variable was the maximum lag allowed for calculating the CCV, expressed as a positive integer number of frames. In MATLAB xcov function, when a maximum lag is not specified, the range in which the CCV is calculated equals 2N−1, where N is the greater of the lengths of the two input signals. In contrast, when a maximum lag is specified, the CCV is calculated in the interval [−maximum lag, maximum lag], where the boundaries are included. The typical delay affecting paired trials in this study was due to human error when clicking the start button on the two different computers controlling the two motion capture systems. Typically, this delay was in the order of tenths of a second. Based on this consideration, and to include a safety margin, the maximum lag was set to 2 s, resulting in an interval of [−2, 2] s for the calculation of the CCV. Limiting the maximum lag had two advantages: first, it improved the reliability of the algorithm, because unlikely solutions corresponding to very large shifts were automatically excluded; second, it minimized the processing time, because the number of CCV computations was substantially reduced.

As output, the algorithm returned the two synchronized trials, the best estimate of the shift, and the result of the objective function minimized to find this best estimate (see section 6.3.5 Algorithm).
6.3.4 Interpolation of input data

To align two motion trials, the proposed synchronization algorithm shifted one trial by an integer number of frames with respect to the other. The native motion trials framerates were 120 fps for Vicon-derived data, and 30 fps for Kinect-derived data. If Kinect-data is up-sampled to match the Vicon framerate, the synchronization resolution would be 1 s / 120 frames = 0.008 s. Even if the best estimate of the shift is found by the synchronization algorithm, a remaining lag of half the resolution (0.004 s) would still have a significant impact on the results of the Bland-Altman analysis of agreement, particularly with high-speed movements such as running or landing.

To address this problem and find the optimal target framerate, a number of Bland-Altman analyses of agreement were carried out using a set of paired Kinect- and Vicon-derived trials. These trials, which included treadmill walking and running, were recorded using a single participant and utilising the protocol described in Chapter 9 of this thesis. For each Bland-Altman analysis, the target framerate was progressively doubled, ranging from 120 fps (the native Vicon framerate) up to 960 fps. Trials were up-sampled using spline interpolation in MATLAB. For each framerate, the limits of agreement (LOA) between each pair of signals were calculated. On average it was found that a framerate of 240 fps corresponded to substantially narrower LOA compared to 120 fps, in the order of magnitude of 1 cm. A framerate of 480 fps further narrowed the LOA by about 1 mm compared to 240 fps. A further increase to 960 fps didn’t correspond to a significant improvement in the agreement, but increased the computational cost due to the greater number of samples to be analysed and stored in memory.

Based on these observations, all trials analysed in the agreement studies reported in Chapters 8 and 9 of this thesis were up-sampled to 480 fps using spline interpolation, before being passed as input arguments to the proposed synchronization algorithm. The finer shifting resolution provided by this higher sampling rate improved the synchronization (and consequently the agreement) results, compared to using the native sampling rate of the Vicon system. Furthermore, the interpolation of Kinect data to a constant framerate resolved the intrinsic limitation of Kinect recording data at a variable frequency, which is only approximately equal to 30 fps.

6.3.5 Algorithm

Given two paired trials, the objectives of the proposed algorithm were to accurately synchronize them, and to truncate them at the same length causing minimum loss of information (see section 6.5 Appendix for the complete MATLAB code). The algorithm started with the identification of the two input trials as short (denoted as $f$) and long (denoted as $g$), based on their number of frames (Figure 50). A variable denoted as $AverageSD$ was created and provisionally set to infinite, for later use. CCV was calculated between the first pair of signals composing the two trials. A candidate shift was determined in correspondence of the maximum of the CCV, according to Eq. 6.1. Using this candidate shift, the two trials were tentatively synchronized and truncated, using the custom developed $SyncAndTruncate$ function.
Two variables influenced the behaviour of the SyncAndTruncate function. These variables were the relative length of the first signal with respect to the second, and the amplitude of the shift. The possible combinations of these variables and the corresponding behaviours of this function are described in detail hereinafter.

Let $L_1$ and $L_2$ be the lengths of the two input trials and $s$ be the unknown shift to be applied to the second signal in order to align it with the first. The following relations represent all possible combinations between the lengths of the trials and the sign of the shift:
\[
\begin{cases}
L_1 < L_2 \\
L_1 = L_2 \\
L_1 > L_2 
\end{cases}
\]
\[
\begin{cases}
s < 0 \\
s = 0 \\
s > 0 
\end{cases}
\]

(6.2)

corresponding to a total of \(3^2 = 9\) possible cases to be solved by the SyncAndTruncate function. To reduce the number of cases, the two input trials were already identified as short \(L_f\) and long \(L_g\) outside this function, based on their relative lengths. In this way, 3 out of 9 cases were eliminated because \(L_f\) could not be longer than \(L_g\), and Eq. 6.2 were rewritten as:

\[
\begin{cases}
L_f < L_g \\
L_f = L_g 
\end{cases}
\]
\[
\begin{cases}
s < 0 \\
s = 0 \\
s > 0 
\end{cases}
\]

The remaining 6 cases were solved via one of the 3 following options (Figure 51). It should be noted that the three cases reported below assumed that \(L_f \leq L_g\), therefore they included two cases each:

1. **Negative shift** \((s < 0)\): \(g\) was circularly shifted backwards by \(|s|\) positions. Then both trials were truncated at the end in correspondence of:

   \[
   \text{newLength} = \min(L_f, L_g - |s|)
   \]

   samples from the start.

2. **Positive shift** \((s > 0)\): instead of shifting \(g\) forwards, \(f\) was circularly shifted backwards by \(s\) positions, to avoid truncation at the start of the signals. Then, both signals were truncated at the end in correspondence of:

   \[
   \text{newLength} = L_f - s
   \]

   samples from the start.

3. **Zero shift** \((s = 0)\): signals were already aligned. If \(L_f < L_g\), \(g\) was truncated at the end in correspondence of:

   \[
   \text{newLength} = L_f
   \]

   samples from the start.
After synchronization and truncation using the candidate shift, an array of differences was calculated for each pair of signals composing the trials. For each array of differences, the SD was determined. Then, the mean of all these SD was computed.

This entire process, representing the first iteration of the proposed algorithm, was repeated for all pairs of signals. For each iteration, a candidate shift and a candidate mean SD were calculated. At the end of all iterations, the candidate shift that generated the smallest candidate mean SD was selected as the final shift to be used to synchronize and truncate the two trials.

The mean of the SD of the differences between signals was selected as objective function to be minimized because the Bland-Altman limits of agreement are calculated as mean \( \pm \) SD of the differences between measurements (Bland & Altman 1986). Agreement is reduced by poor synchronization, since measurements captured at different times by the two systems are compared. However, a temporal misalignment between the signals doesn’t affect the mean of the differences (also known as bias); it only affects the SD of the differences. For this reason, the SD of the differences between paired signals was selected as the objective function to be minimized in order to achieve optimal synchronization.

6.4 Conclusion

In conclusion, an iterative synchronization algorithm based on cross-covariance was developed. The proposed solution presents some benefits compared to the MATLAB in-built algorithm denoted as **alignsignals**: first, it doesn’t add any additional values (e.g. zeros) to the signal in order to achieve a shift. Second, it truncates both paired trials to the same length, with minimum loss of information. Third, it achieves highly accurate synchronization results based on an iterative approach using cross-covariance. Fourth, it is valid for two different types of motion data, i.e. 3D marker coordinates (X, Y and Z) and joint angles. This algorithm was employed as
part of the data analysis pipelines used in the agreement studies discussed in Chapters 8 and 9 of this thesis.
6.5 Appendix

Given the complexity of the synchronization algorithm described in section 6.3.5 Algorithm, its MATLAB code was reported below. It should be noted that the function denoted as SyncAndTruncate in the thesis was originally denoted as ShiftAndTrim in the code.

```matlab
function [s1_sync, s2_sync, shift, mean_sd] = SyncStructs(s1, s2, ...
    fields_to_sync_1, varargin)
% This function synchronizes and trims to the same length two mocap trials
% using cross-covariance and returns the shift value between the trials.
% The optimal shift which ensures the best synchronization is calculated
% iteratively, minimizing the mean of the standard deviations of the
% differences between pairs of signals.
% © October 2014 Alessandro Timmi, edited in September 2016.
% Input:
% s1, s2 = two structs containing as fields the signals to be synced. One
% of the fields in each struct must contain the timestamps and be
% called "time".
% fields_to_sync_1 = cell array with the name(s) of the field(s) to be used
% for synchronization. For .mot files, they must be joint angles
% (e.g. {'knee_angle_r'}). For .trc files, they must be marker names
% (e.g. {'RASI'}). If multiple markers are defined, synchronization
% will be based on all of them.
% fields_to_sync_2 = (optional parameter, default = fields_to_sync_1) if
% the two structs have fields with different labels, we need to
% specify the field(s) for sync in the second struct correspondent
% to that in the first struct. The length of the two cell arrays must
% be the same.
% coords_to_sync = (optional parameter, default = 'X') cell array with
% coordinates to be used for synchronization. If multiple coordinates
% are defined, synchronization will be based on all of them. For .mot
% files, the default value can be used, because each joint angle is
% a single coordinate.
% maxlag = (optional parameter, default = []) maximum lag, expressed in
% number of frames. If specified, the returned cross-correlation
% sequence is calculated only in the range [-maxlag maxlag], instead
% of across the entire length of the longer signal.
% Output:
% s1_sync, s2_sync = the two structs, synced and trimmed at the same
% length.
% shift = If shift > 0, s1 was moved backwards. If shift < 0, s2 was moved
% backwards.
% mean_sd = mean of the standard deviations of the differences between
% pairs of signals. Used as objective function to be minimized via
% synchronization. Also useful to monitor the results of
% synchronization of a set of trials.
%
% Default input values:
default.fields_to_sync_2 = fields_to_sync_1;
default.coords_to_sync = {'X'};
default.maxlag = [];

% Setup input parser:
p = inputParser;
p.CaseSensitive = false;

% Add required input arguments:
addRequired(p, 's1', @isstruct)
addRequired(p, 's2', @isstruct)
addRequired(p, 'fields_to_sync_1', ...
    @(x) validateattributes(x, {'cell'}, {'vector', 'nonempty'}));
```

127
% Add optional input arguments in the form of name-value pairs:
addParameter(p, 'fields_to_sync_2', default.fields_to_sync_2,...
    @(x) validateattributes(x, {'cell'}, {'vector', 'nonempty'}));
addParameter(p, 'coords_to_sync', default.coords_to_sync,...
    @(x) validateattributes(x, {'cell'}, {'vector', 'nonempty'}));
addParameter(p, 'maxlag', default.maxlag,...
    @(x) validateattributes(x, {'numeric'}, {'scalar', 'nonnegative',
    'integer'}));

% Parse input arguments:
parse(p, s1, s2, fields_to_sync_1, varargin{:});

% Copy input arguments into more memorable variables:
s1 = p.Results.s1;
s2 = p.Results.s2;
fields_to_sync_1 = p.Results.fields_to_sync_1;
fields_to_sync_2 = p.Results.fields_to_sync_2;
coords_to_sync = p.Results.coords_to_sync;
maxlag = p.Results.maxlag;

clear varargin default p

%%
fprintf('Starting sync and trim procedure...\n');

% Check that fields_to_sync_1 and fields_to_sync_2 have same length:
if ~isequal(length(fields_to_sync_1), length(fields_to_sync_2))
    error('fields_to_sync_1 and fields_to_sync_2 have different length.\n')
end

% NOTE: In min() function, if there are several identical minimum values,
% the index of the first one found is returned (which in this case is 1).
[frames_short, shortest] = min([length(s1.time), length(s2.time)]);

% Detect the short and long trials in terms of number of frames:
if shortest==1
    s_short = s1;
s_long = s2;
    fields_to_sync_short = fields_to_sync_1;
    fields_to_sync_long = fields_to_sync_2;
elseif shortest==2
    s_short = s2;
s_long = s1;
    fields_to_sync_short = fields_to_sync_2;
    fields_to_sync_long = fields_to_sync_1;
else
    error('Problem while detecting the shortest signal');
end

% We need the length of the long dataset too:
frames_long = length(s_long.time);

% Convert coordinates to sync from char to corresponding numbers:
coords_to_sync_num = coord2num(coords_to_sync);

% Synchronization of the long dataset relative to the short:
fprintf('Synchronization based on the following fields:\n')
for i=1:length(fields_to_sync_1)
    fprintf('\t"%s" and "%s"\n', fields_to_sync_1{i}, fields_to_sync_2{i});
fprintf('and on the following columns:
')
for i=1:length(coords_to_sync_num)
    fprintf('	%d
', coords_to_sync_num(i))
end

% If not passed as input argument, set maxlag to its default value
% reported in Matlab implementation.
% We need to do this, otherwise we should write the cross-covariance
% command twice (with and without maxlag as optional argument).
% According to Matlab help for the xcov() function, the default value for
% maxlag is equal to N - 1, where N is the length of the longer signal.
% In this way, the lag range is 2N - 1:
if isempty(maxlag)
    maxlag = frames_long - 1;
end

% Pre-set this variable which will be used below:
mean_sd = Inf;

fprintf('List of calculated shifts:
')
for i = 1:length(fields_to_sync_1)
    for j = 1:length(coords_to_sync)
        % Cross-covariance for current pair of signals:
        [cc, lags] = xcov(...
            s_short.markers.(fields_to_sync_short{i})(:, coords_to_sync_num{j}),...
            s_long.markers.(fields_to_sync_long{i})(:, coords_to_sync_num{j}),...
            maxlag);
        % Find index corresponding to maximum cross-covariance:
        [~, max_cc_index] = max(cc);
        % Get the lag corresponding to max cross-covariance. This will be
        % the SHIFT for the current pair of signals (temporary shift):
        shift_temp = lags(max_cc_index);
        fprintf('Shift from %s %s: %d.
', fields_to_sync_short{i}, ...
            coords_to_sync_num(j), shift_temp);
        % Sync and trim the two structs using the temporary shift value,
        % returning them in the order in which they were input:
        [s1_sync_temp, s2_sync_temp] = ShiftAndTrim(...
            s_short, s_long, shift_temp, ...
            frames_short, frames_long, shortest);
        % To determine if this is the “best” shift, we need
        % to verify if it minimizes the standard deviation (SD) of the
        % differences between pairs of signals. Indeed, a pure sync error
        % affects only the SD, not the mean (or constant bias) of the
        % differences between two signals.
        % Therefore, we calculate the SD of the differences for each pair
        % of signals to be synced.
        % Preallocate an array for the SD of each pair of signals:
        sd_collection = zeros(length(fields_to_sync_1), length(coords_to_sync));
        for k = 1:length(fields_to_sync_1)
            % Remember that std() operates along the columns, returning a row
            % vector.
            sd_collection(k,:) = std(s1_sync_temp.markers.(fields_to_sync_1{k}) - ...
                s2_sync_temp.markers.(fields_to_sync_2{k}));
        end
        % Calculate the mean of all the SD...
mean.sd_temp = mean(sd_collection(:));
% ... and compare it to the reference one:
if mean.sd_temp < mean.sd
% If it is smaller, it becomes the new reference:
mean.sd = mean.sd_temp;
% and the correspondent temporary shift is stored:
shift = shift_temp;
end
end
end

% Sync and trim the two trials using the best shift found. Now we also get
% as output the shift with the correct sign, taking into account whether or
% not the two trials were swapped due to their lengths:
[s1_sync, s2_sync, shift] = ShiftAndTrim(s_short, s_long,...
    shift, frames_short, frames_long, shortest);
end

function [s1_sync, s2_sync, shift] = ShiftAndTrim(f, g, s, Lf, Lg, shortest)
if s < 0
% CircShift g backwards:
g.markers = structfun(@(y) ( circshift(y, s, 1) ),
    g.markers, 'UniformOutput', false);

% Determine new trial length:
newLength = min(Lf, Lg - abs(s));

% Trim both trials at the new length from the start:
f.markers = structfun(@(y) ( y(1:newLength, :) ),
    f.markers, 'UniformOutput', false);
g.markers = structfun(@(y) ( y(1:newLength, :) ),
    g.markers, 'UniformOutput', false);

f.time = f.time(1:newLength);
g.time = g.time(1:newLength);
elseif s > 0
% CircShift f backwards:
f.markers = structfun(@(y) ( circshift(y, -s, 1) ),
    f.markers, 'UniformOutput', false);

% Trim both trials at Lf - s from the start:
newLength = Lf - s;

f.markers = structfun(@(y) ( y(1:newLength, :) ),
    f.markers, 'UniformOutput', false);
g.markers = structfun(@(y) ( y(1:newLength, :) ),
    g.markers, 'UniformOutput', false);

f.time = f.time(1:newLength);
g.time = g.time(1:newLength);
elseif s == 0
fprintf('Signals are already synced (shift = 0)')
if Lf < Lg
    fprintf('.

')
% Trim g at Lf from the start:
newLength = Lf;
end
g.markers = structfun(@(y) (y(1:newLength, :)), ... 
g.markers, 'UniformOutput', false);

g.time = g.time(1:newLength);

else
    fprintf(' and have same length.

end

else
    error('Error during the detection of the shift')
end

% Return the structs in the same order as they were input and the shift
% with the same sign:
if shortest == 1
    s1_sync = f;
    s2_sync = g;
    shift = s;
else
    % Trials were swapped, swap them back:
    s1_sync = g;
    s2_sync = f;
    % Since we have just swapped the trials, we also need to change the
    % sign of the shift before returning it:
    shift = -s;
end

% Update structs info for .trc files:
% This "if" is just a way to check whether this is a .trc or .mot struct,
% because .mot files don't need this step.
if isfield(s1_sync, 'NumMarkers')
    s1_sync = UpdateTrcStructInfo(s1_sync);
    s2_sync = UpdateTrcStructInfo(s2_sync);
end

6.6 References

methods of clinical measurement. The Lancet, 327(8476), pp.307–310. Available at:


p.279–280. Available at:
http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:op+amps+for+everyone#0.

MathWorks, 2018. Cross-covariance. Matlab Documentation. Available at:
[Accessed October 14, 2018].
CHAPTER 7
Analysis of agreement between two measurement methods

7.1 Introduction
Motion capture systems can measure the instantaneous 3D position of discrete points on the body. One of the aims of this thesis was to develop a novel motion capture system, which was more affordable and portable compared to multi-camera optical motion capture systems but had sufficient accuracy to be used for clinical biomechanical applications. When a new method of measurement is introduced, it is necessary to evaluate the agreement between the novel method and the established gold standard, in order to verify if the two methods can be used interchangeably for a specific purpose. Given a measurement by the new method (which can be less expensive, less invasive or generally more convenient than the gold standard one), the objective of agreement studies is to assess how far this might be from a measurement by the established method. Although the problem of assessing agreement between measurement methods is quite common in clinical practice, in the last few decades there has been much controversy concerning the choice of statistical technique to be used. As a result, there is considerable confusion surrounding the meaning and interpretation of results from such studies investigating measurement error (Muller & Buttner 1994; Bartlett & Frost 2008).

In 1986, Bland and Altman (Bland & Altman 1986) proposed a simple yet effective way to assess the agreement between two measurement methods. Their methodology is based on the evaluation of the bias between the two systems, calculated as the mean of the differences between the measurements, and the 95% limits of agreement (LOA), calculated as the bias ± 1.96 standard deviations of the differences between the measurements. If the differences are normally distributed, approximately 95% of them lie between the LOA. The bias and the LOA are visualized in a diagram now known as the Bland-Altman plot.

Bland and Altman’s 1986 paper was published in The Lancet and is the 29th most cited study over all fields according to Nature (Van Noorden et al. 2014), featuring 27,918 citations as of 22 April 2017 according to Web of Science. Interestingly, the ideas discussed in this paper were already presented in a previous article published in 1983 on The Statistician (Altman & Bland 1983). The 1986 paper was only an attempt to publicise the approach to a medical audience (Bland & Altman 2012; Bland 2014). The success of the 1986 paper led to continuous enquiries by researchers on how to apply the proposed statistical technique to more complex cases where the original simple method did not apply. The authors addressed many of these questions in their subsequent papers which have also been highly cited (Bland & Altman 1990; Bland & Altman 1995a; Bland & Altman 1995b; Bland & Altman 1999; Bland & Altman 2003; Bland & Altman 2007; Bland & Altman 2012), indicating the usefulness and success of the Bland-Altman technique for comparing novel measurement methods with more established ones.

Despite the effectiveness of the Bland-Altman approach, alternative statistical techniques are still commonly used for evaluating the agreement between two measurement systems. The most commonly used methods include, but are not limited to the following: correlation coefficient, Student’s t test, intraclass correlation, linear regression, concordance correlation
coefficient, root mean square deviation (Bland & Altman 1995b; White & van den Broek 2004; Costa-Santos et al. 2010). Each of these alternative approaches was originally intended for different applications. Consequently, all these approaches present substantial limitations when used for the analysis of agreement between methods of measurement.

The objective of this chapter was two-fold: first, to review the statistical techniques more commonly used for method agreement studies, illustrating why the Bland-Altman approach is the most suitable technique for assessing the agreement between a novel motion capture system and an established one. Second, to describe the implementation of the Bland-Altman method used in this thesis for the validation of the proposed motion capture system.

7.1.1 Statistical techniques commonly used for assessing agreement
Bland-Altman analysis of agreement
The critical question addressed by the Bland-Altman analysis is: “Can we replace the old method of measurement by the new method and treat the resulting observations in the same way?” (Bland & Altman 1995b). In other words, the aim of this analysis is to determine whether two methods agree sufficiently well for them to be used interchangeably. Numerically, this question can be expressed as follows: given a measurement by the new method, how strongly might this deviate from a measurement by the old method (Bland & Altman 1995a; Bland & Altman 1995b)?

The first step in the analysis of agreement is to calculate the differences between corresponding measurements by the two methods. Then the mean of these differences, $\bar{d}$, and their standard deviation, $s_d$, are calculated. If the differences are from an approximately Normal distribution, and if they are independent of the magnitude of the measurement, 95% of them are expected to lie within $\bar{d} - 1.96 s_d$ and $\bar{d} + 1.96 s_d$, which are called the 95% limits of agreement (LOA). These limits include the differences between single measurements on the same subject by the two methods with probability 95%.

To check if the differences are normally distributed, a histogram can be constructed and visually inspected. It should be noted that the measurements themselves do not have to follow a Normal distribution, and often they will not (Bland & Altman 1986). To verify if $\bar{d}$ and $s_d$ are constant throughout the range of measurement, the authors suggested to report a scatter diagram of the difference against the average of the two measurements (Bland & Altman 2003). The LOA and $\bar{d}$ are then added to the scatter plot, and about 95% of the points should lie within the limits. This diagram is known as Bland-Altman plot.

The mean of the differences $\bar{d}$ is also known as bias and represents the systematic difference between the two methods. The bias indicates whether on average one method tends to underestimate or overestimate measurements relative to the measurements of the second method (Bartlett & Frost 2008). Standard deviation, used to calculate the LOA, measures random fluctuations around the mean (Bland & Altman 1995a). LOA are ultimately just estimates and, as with any type of estimate, it is essential to report their precision calculating their 95% confidence intervals (CI, see subsection Confidence intervals in section 7.1.2).

Pearson correlation
The Pearson or product-moment correlation coefficient, $r$, also known as interclass correlation or sample correlation coefficient, is the most commonly used measure of correlation. It is employed to assess the strength of the association between two variables (Gordon 2014) and answers the question: how well are the two methods of measurement related?
Suppose we have \( n \) pairs of observations on two variables \( X \) and \( Y \): \((x_1, y_1), \ldots, (x_n, y_n)\). One may define \( s_x \) to be the standard deviation of the sample on \( X \) and \( s_y \) to be the standard deviation of the sample on \( Y \). The sample correlation coefficient \( r \) is given by:

\[
r = \frac{1}{1 - \frac{1}{n}} \sum_{i=1}^{n} \left( \frac{x_i - \bar{x}}{s_x} \right) \left( \frac{y_i - \bar{y}}{s_y} \right).
\]

The correlation coefficient \( r \) ranges between \(-1 \leq r \leq 1\). Only when there is an exact linear relationship between the two datasets, \( r = \pm 1 \). A value of \( r = 0 \) indicates no linear relation between the values of the two variables. However, a very small correlation does not mean that two variables are not associated via another type of relation. Furthermore, the correlation does not depend on the units of the two variables.

Pearson correlation should not be used for agreement studies for several reasons. First, \( r \) measures the strength of a relation between two variables, not the agreement between them. There is perfect agreement only if the points in a scatter plot of one method against the other lie along the line of equality (i.e. diagonal), but there is perfect correlation if the points lie along any straight line (Bland & Altman 1986; Bland & Altman 1995b). For example, let us say we have two methods of measurement \( X \) and \( Y \), whose correlation is 0.86. If we have a third set of measurements \( Z \), obtained by adding 2 to \( X \), this will consistently overestimate the true value by 2 units. However, the correlation between \( Y \) and \( Z \) is the same as its correlation with \( X \), 0.86, because correlation ignores any systematic bias between the two variables (Bland & Altman 2003). Indeed, a strong correlation allows nearly perfect prediction of one measure from the other, but the actual agreement may be non-existent (Muller & Buttner 1994).

Second, correlation depends on the range of the true quantity in the sample. If this is wide, the correlation will be greater than if it is narrow (Bland & Altman 1986; Bland & Altman 2003). For example, suppose the quantity to be measured has between-subject variance, \( \sigma_T^2 \), and the two measurement methods A and B have within-subject errors, independent of each other and the true value, with variance \( \sigma_A^2 \) and \( \sigma_B^2 \). Then the correlation between methods is given by (Bland & Altman 1995b):

\[
r = \frac{\sigma_T^2}{\sqrt{(\sigma_T^2 + \sigma_A^2)(\sigma_T^2 + \sigma_B^2)}}
\]

The larger \( \sigma_T^2 \) is compared to \( \sigma_A^2 \) and \( \sigma_B^2 \), the larger will be \( r \). In other words, the observed correlation coefficient is strongly dependent on the sample chosen. A high correlation for any two methods designed to measure the same property could just be a sign that a widespread sample was chosen (Giavarina 2015), i.e. one with high variability. Because researchers usually try to compare the two methods over the whole range of values commonly encountered, a high correlation is almost guaranteed (Bland & Altman 1986). An implication of this fact is that significant correlation can result from a single outlying value of the paired observations obtained from the two methods, a problem that could be easily identified via a scatterplot of the data (Gordon 2014).

Student’s t test

The comparison of two methods of measurement by a Student’s t test is another suboptimal approach to agreement studies. It tests the null hypothesis that the mean difference between the methods is zero in the population. It thus assesses whether there is evidence that the two methods agree on average, but gives no information about how well they agree for individuals.
(Bland & Altman 1995b). Before illustrating this limitation, it should be noted that the $t$ value in a one-sample $t$ test is calculated as:

$$t = \frac{\bar{x} - \mu_0}{s/\sqrt{n}}$$  \hspace{1cm} (7.1)

where $\bar{x}$ is the sample mean, $\mu_0$ is the hypothesized mean of the population, $s$ is the sample standard deviation and $n$ is the sample size. In an agreement study, the sample is composed of the differences between the two methods, $\mu_0$ is equal to zero and the denominator of Eq. 7.1 represents the standard error (SE) of the sample. Furthermore, it should be noted that the $P$-value indicates the probability of observing a result at least as extreme as that obtained, given that the null hypothesis is true (Gordon 2014). If the $P$-value is small, this is evidence against the null hypothesis; if the $P$-value is large, there is no reason to doubt the null hypothesis. Since the $P$-value is calculated from the probability density function using the $t$-value as input, the larger is the absolute value of the $t$ value, the smaller is the corresponding $P$-value (Figure 52).

![Two-tailed test](image)

Figure 52. The larger the absolute value of $t$, the smaller the $p$-value

If there is systematic bias between the methods, the numerator of the $t$ value tends to be high, consequently the $P$-value tends to be low, hence the $t$ test tends to be significant. However, in this case, the $t$ test is less likely to be significant if the agreement is poor, because the standard deviation of the differences, and hence the standard error of the mean difference, will be high\(^{23}\). In contrast, if the two methods agree on average (i.e. there is no systematic bias between them), the $P$-value tends to be high. In this case, the $t$ test will be non-significant regardless of the agreement for individuals. In conclusion, the method will lead to the wrong conclusion whether agreement is good or poor. All the limitations of the $t$ test apply also to the analysis of variance (ANOVA), which is sometimes used when more than two methods are evaluated (Bland & Altman 1995b).

### Intraclass correlation

Intraclass correlation (ICC) was developed at the end of the 19th century to measure concordance of items in genetics. Thereafter, ICC gained entry first into psychology and then into medicine in general (Muller & Buttner 1994).

ICC is defined as the ratio of the variance between subjects, $\sigma_B^2$, to the total variance, $\sigma_T^2$. These variances are derived from ANOVA (Costa-Santos et al. 2010):

$$ICC = \frac{\sigma_B^2}{\sigma_T^2}$$  \hspace{1cm} (7.2)

\(^{23}\) Indeed, a large $s$ would decrease the $t$-value (Eq. 7.1) and consequently increase the $P$-value.
ICC was intended to examine questions of, for example, concordance of a genetic variable in twins, or to assess the reliability of a psychological questionnaire by means of repeating similar questions about the same subject. ICC allows tests against zero, and, therefore, this method is adequate for study settings where the main interest lies in assessing whether the concordance is greater than by chance. In methods agreement studies, the aim is to judge how much two instruments differ, and it is of little interest whether they concur more than by chance (Muller & Buttner 1994).

Although the ICC avoids the problem of linear relationship being mistaken for agreement (Bland & Altman 1990), it has important limitations affecting its capability to assess the agreement between two methods of measurement. First, ICC ignores the order of the two variables. Indeed, given two variables X and Y, the ICC is the average correlation across all possible orderings of paired measurements into X and Y. Using the ICC and ignoring the ordering, the two methods of measurement would be treated as a random sample from a population of methods. However, in agreement studies, there is a very clear ordering, the two variables being the two methods. (Bland & Altman 1990; Bland & Altman 1995b).

Second, this technique assumes that the measurement error is the same for both methods, whereas the objective is to find this result from the method comparison study (Bland & Altman 1990).

Third, similarly to Pearson correlation, ICC depends on the range of the measurements in the sample: the more variable the subjects (larger $\sigma^2$ in Eq. 7.2), the greater will be the ICC. Therefore, the ICC is higher when applied to a heterogeneous population (high variance) than when applied to a very homogeneous population (low variance) (Bland & Altman 1990; Muller & Buttner 1994; Bland & Altman 1995b; Costa-Santos et al. 2010). Some authors considered this characteristic of ICC an advantage, because it would make the discordance relative to the magnitude of measurements, however it prevents comparisons across populations. Consequently, ICC values have no absolute meaning, and the cut off value of 0.75 often reproduced to signify good agreement is not justified (Bland & Altman 1990; Muller & Buttner 1994; Costa-Santos et al. 2010). If studies report only ICC, readers can only make use of the estimate if the population in which the reader intends to use such measurements has equal heterogeneity (Bartlett & Frost 2008).

Fourth, as a dimensionless quantity, it is arguably quite difficult to interpret, and deciding what value constitutes sufficiently high agreement is often made in a subjective fashion (Bartlett & Frost 2008). Indeed, ICC is not related to the scale of measurement or to the size of error which might be clinically allowable (Bland & Altman 1990).

Fifth, ICC ranges from 0 (no agreement) to 1 (perfect agreement); however, it may also be negative. How such negative ICC values should be interpreted remains unclear (Costa-Santos et al. 2010), and the suggestion to redefine them as zero does not really solve the problem (Muller & Buttner 1994).

Sixth, it is impossible to summarize the agreement adequately using a single number, because in this way there is not separation between systematic and random error (Bland & Altman 1990).

The ICC may be used when two or more new measurement methods are to be compared with a gold standard. The ICC approach, resulting in a single measure, is the easiest method to judge which of the new methods yields the best accuracy (Muller & Buttner 1994). The ICC may also
have some merits in the unusual case where the two methods being compared do not measure in the same units, but it should not be used otherwise (Bland & Altman 1995b).

Regression and the coefficient of determination

Given two variables within a group of participants, regression predicts the value of one variable for any given value of the other variable. This approach (particularly the linear regression) has been often incorrectly adopted in agreement studies, even though their purpose is different from predicting one variable from the other. It is often thought that, as the data in a scatterplot should cluster around the line of equality for good agreement, the regression line should be similar to the line of equality, i.e. its slope should be 1 (Bland & Altman 1994b). This is not so, because regression attempts to predict the observed Y from the observed X, not the true Y from the true X. Measurement errors in X reduce the slope of the line and so raise the lower end of the line and lower the upper end, so that the intercept is increased above zero. This is the effect of a well-known statistical phenomenon called regression towards the mean (Bland & Altman 1994a). Assuming Y is the novel measurement system and X is the established one, this result can be misinterpreted saying that the new method overestimates the measure when the true value is low and underestimates it when the true value is high. In fact, when we regress measurements by one method on measurements by another, we expect that the slope will be less than one and the intercept greater than zero. Therefore, a slope < 1.0 and an intercept > 0.0 do not tell us anything (Bland & Altman 2003).

Considering a linear regression model, the slope and intercept of the regression line cannot indicate if two methods agree closely or are widely scattered about the line. The standard errors of the slope and intercept indicate the precision of the estimates, but have not direct bearing on the between-method agreement. Regression methods may be used to calibrate one method against the other, using the new method to predict the old. This may be particularly relevant when the two methods use different units. However, this is not the purpose of a method comparison study (Bland & Altman 1995b).

The coefficient of determination $R^2$ (or $r^2$, also known as goodness-of-fit) quantifies how well the regression model performs, in terms of explaining the variation in the response variable Y, as a function of the explanatory variables, the xs:

$$R^2 = \frac{SS_{reg}}{SS_{tot}} = 1 - \left( \frac{SS_{res}}{SS_{tot}} \right).$$

The SS (sums of squares) in these expressions come from the ANOVA. The total SS ($SS_{tot}$) represents the overall variation in the response variable Y. The residual SS ($SS_{res}$) represents the variation from the fitted line. $R^2$ lies between 0 and 1, where 0 indicates that the model explains none of the variability of the response data around its mean, and 1 indicates that the model explains all the variability of the response data around its mean.

Usually, a linear regression study is performed together with correlation measurement (Giavarina 2015). When there is just one explanatory variable, x, and a simple linear regression model, $r = \sqrt{R^2}$ is in fact the Pearson’s correlation between Y and x (Gordon 2014). Many examples of regression analysis applied to agreement studies are available in scientific literature. However, when the agreement analysis is conducted on a wide range of measurements, correlation and linear regression are not particularly informative, and could be misleading (Giavarina 2015).
Concordance correlation coefficient (CCC)

In 1989, Lin proposed a concordance correlation coefficient (CCC) (Lin 1989). This coefficient is obtained from Pearson’s $r$ correlation coefficient, which assesses the closeness of the data to the line of best fit, modified by taking into account how far the line of best fit is from the 45° line through the origin. The CCC does not fulfil the definition of an ICC, however simulation studies as well as theoretical considerations showed that this approach behaves quite similarly to an ICC (Muller & Buttner 1994). For this reason, it is not recommended for agreement studies, as it shares many of the issues already described for the ICC. For example, it is strongly influenced by the variance of the population in which it is assessed (Costa-Santos et al. 2010).

Root-mean-square deviation

The root-mean-square deviation (RMSD) or root-mean-square error (RMSE) is a quantity measuring deviation of a random variable from some standard or accepted value. Its value is determined by:

$$ RMSD = \sqrt{\frac{1}{n} \sum_{i=1}^{n} (x_i - a_i)^2} $$

where $x_i$ represents the set of $n$ observations and $a_i$ is the corresponding set of accepted values (Deakin & Kildea 1999).

Compared to the alternative statistical approaches, the RMSD has the advantage of being on the same scale as the data. However, it is a single number and, as such, it includes components of both variance and bias. Consequently, this technique is not capable of separating the error in its two components, the systematic part and the random part.

Advantages of the Bland-Altman analysis of agreement

For agreement studies, the BA analysis has several advantages compared to other statistical techniques. First, it doesn’t depend on the range of the sample. Second, it separates the systematic and random components of error between the two methods, represented by the bias and the LOA respectively.

Third, since the LOA have the same scale as the measured quantity, deciding what magnitude of error is acceptable becomes a clinical decision, not a statistical one. The question is whether the agreement is good enough for a particular purpose, not whether it conforms to some absolute, arbitrary criterion. Methods which may agree well enough for one purpose may not agree well enough for another.

Fourth, plotting the differences against the mean allows identification of outliers and examination for trends, either visually or by means of linear regression analysis (Bland & Altman 1990; Muller & Buttner 1994). Bland and Altman have recommended different types of data transformations, to deal with cases in which relationships between difference and mean are present (Bland & Altman 1986; Bland & Altman 1996; Bland & Altman 1999).

Fifth, being a more visual approach compared to other statistical techniques, the Bland-Altman analysis of agreement can be easily explained to non-statisticians (Muller & Buttner 1994). All these features are not concurrently available in any of the other statistical techniques. Therefore, such alternative methods are not recommended for agreement studies, because their results can be misleading.
7.1.2 Performing a Bland-Altman analysis of agreement

Why plotting difference against standard method is misleading

When comparing one measurement technique with another, it is necessary to verify if the difference between the measurements by the two methods is related to the magnitude of the measurement. For the mean and SD of the differences to be meaningful estimates, they must be reasonably constant throughout the range of measurement. Bland and Altman proposed to check this assumption graphically, plotting the differences against the average of the measurements by the two methods (Bland & Altman 1995a). Typical departures from this assumption are: 1) an increase in variability, displayed as an increment in the scatter of the differences as the magnitude of the measurement increases; 2) a trend in the bias, i.e. a tendency for the mean difference to rise or fall with increasing magnitude. Either would show that the methods do not agree equally through the range of measurement.

It is sometimes argued that when one method may be considered as a gold standard, it may be more accurate than the other method and so the difference should be plotted against the gold standard. However, a plot of the difference between the new and the standard method against the standard measurement will always show a relation, whether there is a true association between difference and magnitude or not (Bland & Altman 1995a).

In the following demonstration by Bland and Altman, the standard measurement is denoted by $S$, the new or test measurement by $T$, their variances by $\sigma_S^2$ and $\sigma_T^2$ respectively, and their correlation by $\rho$. If the study includes a wide range of measurements, and unless the two methods of measurement have very poor agreement, the two variances are expected to be similar and $\rho$ to be fairly large, at least 0.7. The possibility of a relation appearing in the Bland-Altman plot can be examined from the expected correlation coefficient between difference and average, which can be shown to be\(^{24}\):

$$\text{Corr}[T - S, (T + S)/2] = \frac{\sigma_T^2 - \sigma_S^2}{\sqrt{[(\sigma_T^2 + \sigma_S^2)^2 - 4\sigma_T^2\sigma_S^2]}}$$

This is zero if the variances are equal, and will be small unless there is a marked difference in the variability between subjects for the two methods. If there is a trend in the difference with increasing magnitude of the measurement, the variances will be different. Thus, there will be a non-zero correlation between difference and average, and the Bland-Altman plot will show the trend.

The expected correlation between difference and standard measurement is:

$$\text{Corr}[T - S, S] = \frac{\rho\sigma_T - \sigma_S}{\sqrt{[\sigma_T^2 + \sigma_S^2 - 2\rho\sigma_T\sigma_S]}}$$

This correlation will usually be negative. In particular, if there is no difference between the variances of the two methods and so no relation between difference and magnitude, the plot of difference against standard will still show a correlation. In this case, the formula reduces to:

\(^{24}\) Note the difference between the correlation of the two variables ($\rho$) and the correlation of difference and average of the two variables.
\[ \text{Corr}[T - S, S] = -\frac{\sqrt{1 - \rho^2}}{2} \]

This spurious correlation will be small when the methods being compared are themselves highly correlated, and will increase as the correlation between the two methods themselves falls.

The expected correlation between difference and test measurement is:

\[ \text{Corr}[T - S, T] = \frac{\sigma_T - \rho \sigma_S}{\sqrt{\sigma_T^2 + \sigma_S^2 - 2 \rho \sigma_T \sigma_S}} \]

This correlation will usually be positive. Thus, in the absence of a genuine association between difference and magnitude, the plot of difference against test measurement will suggest a positive relation, whereas the plot of difference against standard will suggest a negative one. This shows that both plots are liable to be very misleading and any relation found liable to be an artefact of the method of analysis. A plot of the difference against the average of the standard and new measurements is unlikely to mislead in this way (Bland & Altman 1995a).

**Confidence intervals**

The bias and LOA are only estimates of the values which apply to the whole population. A different sample would provide different estimates (Bland & Altman 1986). The precision of the estimates depends on the amount of observed data, i.e. on the sample size. To see how precise the estimates are, the 95% confidence intervals (CI) must be calculated. LOA should never be reported without CI (Hamilton & Stamey 2007).

For a 95% CI, if many samples of a given size are collected and the CI computed, in the long run about 95% of these intervals would contain the true mean. In the same way that statistical tests can be one or two-sided, CI can be one or two-sided. A two-sided CI brackets the population parameter from above and below. A one-sided CI brackets the population parameter either from above or below and provides an upper or lower bound to its magnitude (NIST/SEMATECH 2012a).

In the Bland-Altman analysis, the two-sided 95% CI of the estimate must be used. Provided the differences follow a distribution which is approximately Normal, they can be calculated as:

\[ \text{CI} = \text{estimate} \pm t_{1-\alpha/2,n-1} \cdot \text{SE} \]  \hspace{1cm} (7.3)

where \( \alpha \) is the desired significance level (0.05 in this case), \( n \) is the sample size, \( t \) is the \( 100(1 - \alpha/2) \) percentile of the \( t \)-distribution with \( n - 1 \) degrees of freedom, and \( \text{SE} \) is the standard error of the sample. The difference \( 1 - \alpha \) is defined as confidence coefficient and is equal to 0.95 in this case. The \( \text{SE} \) of the bias is \( \sqrt{s^2/n} \), whereas the \( \text{SE} \) of the LOA is approximately \( \sqrt{3s^2/n} \). Since the derivation of the latter \( \text{SE} \) is not straightforward, it is reported in section 7.4.1 Confidence intervals of the LOA.

From Eq. 7.3, the greater the number of samples \( n \) used for the evaluation of the difference between the methods, the narrower the CIs. That is, one way to obtain more precise estimates is to increase the sample size. In contrast, the larger the sample standard deviation, the larger the CI, i.e. noisy data generate wider intervals (NIST/SEMATECH 2012b).
Multiple measurements per individual

The study design described in Bland & Altman (1986) involves one measurement by each of the two methods on each participant, under the assumption of independent observations. Two observations are independent if the occurrence of one observation provides no information about the occurrence of the other observation (NEDARC 2016). A simple example is measuring the height of children in a sample at a single point in time: these should be unrelated observations. However, measuring one child’s height over time, the observations would be dependent because the height at each time point would affect the height at future time points.

Since it is often valuable to record multiple observations by each method on each individual, Bland & Altman addressed this more complex case in subsequent publications (Bland & Altman 1999; Bland & Altman 2007). In this case, the differences between the two methods of measurement include: 1) a between-individual component; 2) a within-individual component; 3) an autocorrelation component, if consecutive measurements on the same individual are not independent.

If all differences are treated as if measured on a different participant while in fact they aren’t, the structure of the data is ignored and there exists the risk of underestimating the interval between the LOA (Bland & Altman 2007; Hamilton & Lewis 2010). Specifically, the performance of the independent LOA worsens as the ratio of participant-to-participant vs. within-participant variability increases. This is troubling as generally the difference between measurements recorded on separate patients far exceeds the difference between measurements recorded on the same patient. Furthermore, the coverage of the independent LOA worsens as the number of observations per patient increases. Finally, increasing the number of patients in the trial does not entirely counterbalance the effect of the repeated measurements on the independent LOA. Thus, the use of the independent LOA in a repeated measures setting is ill-advised, even in large trials (Hamilton & Lewis 2010).

There are two possible scenarios to consider in case of replicated data (Bland & Altman 1999; Bland & Altman 2007): first, the true value doesn’t change; second, the true value changes.

The latter case is relevant for the analysis of agreement between Kinect- and Vicon-derived measurements, and is illustrated in this section, using data from (Bland & Altman 1999) as example. The first step is to verify that the agreement is the same over the range of measurement, doing a scatter plot of the difference against the average of the two methods and reporting the zero line (Figure 53). In this case, there is a positive bias, but no obvious variation in agreement across the range of measurements.

25 It is still unclear how much autocorrelation influences the estimate of the variance within subjects. This autocorrelation is something that Bland and Altman have commented on, suggesting it is worthy of further investigation (Bland & Altman 2007).

26 It should be noted that the same data were also presented in a subsequent paper by the same authors (Bland & Altman 2007), however with some errors in Table 1, Figure 3 and Figure 5.
In the case when the true value changes, the limits of agreement can be estimated using a technique based on components of variance. The model is the same of the one-way ANOVA: the observed difference is the sum of the mean difference (bias), a random between-subjects effect (heterogeneity) and a random error within the subject.

In accordance with the assumptions of the one-way ANOVA, the within-subject variance must be constant and observations within the subject must be independent. The first assumption can be verified by plotting the within-subject standard deviation of the differences against the within-subject average measurement, and visually checking that they are unrelated (Figure 54).

The difference for each pair of measurements $j$ on subject $i$ can be modelled as:

$$D_{ij} = B_i + I_i + W_{ij}$$
where $B$ is the constant bias, $I_i$ the interaction term between subjects and methods, and $W_{ij}$ the random error within the subject for that pair of observations.

The variance of $D_{ij}$ is then:

$$\sigma_d^2 = \sigma_{d_i}^2 + \sigma_{dw}^2$$

The between-subject variance ($\sigma_{d_i}^2$) and the within-subject variance ($\sigma_{dw}^2$) can be estimated by performing a one-way ANOVA. Using the difference in matched pairs of measurements as response variable and subjects as grouping variable, the within-groups mean square ($MS_w$) and the between-subjects mean square ($MS_b$) can be obtained.

The within-subject variance as is estimated as:

$$\sigma_{dw}^2 = MS_w$$

and the between-subject variance (heterogeneity) as $28$:

$$\sigma_{di}^2 = \frac{MS_b - MS_w}{(\sum m_i)^2 - m \sum_i^2}$$

where $i = 1, ..., n$ are the subjects and $m_i$ are the number of observations per subject.

The sum of these estimates provides $\sigma_d^2$. The standard deviation is the square root of this variance. The bias is the mean of all differences (Bland & Altman 2007)$^{29}$:

$$B = \text{mean}(D)$$

The 95% limits of agreement are estimated as:

$$LOA = B \pm 1.96 \sigma_d$$

The bias and LOA calculated using data from Bland & Altman (1999) are reported in Figure 55.

---

$27$ Also called error means square or residual mean square.

$28$ The equation of $\sigma_{di}^2$ is reported incorrectly in (Bland & Altman 1999), whereas the correct version is reported in (Bland & Altman 2007).

$29$ In (Bland & Altman 1999), the bias is calculated using an equivalent formula:

$$\text{bias} = \frac{\sum m_i \bar{d}_i}{\sum m_i}$$

where $\bar{d}_i$ is the mean difference for subject $i$. 

143
For the ANOVA, it was assumed that the repeated differences for a single subject were independent. This might be a rather strong assumption, that can be visually checked by plotting observed differences against the order in which the measurements were performed. Such a plot is shown in Figure 56, assuming that data from Bland & Altman (1999) were supplied in temporal order. There appear to be autocorrelation for some subjects, visualized as a trend among the differences with same number: as previously mentioned, whether this influences the estimate of the variance within subjects is unclear and further research in this area is needed (Bland & Altman 2007).

Figure 55. Bland-Altman plot with bias and limits of agreement, based on data from Bland & Altman (1999)

Figure 56. Scatter plot of differences between methods against the order in which the measurements were made (points are represented by the subject number). Note that in Bland & Altman (2007), the values of points labelled 1 and 4 were incorrectly swapped along the vertical axis.
Myles & Cui (2007) proposed an alternative method, using a random effects model instead of the ANOVA to estimate the within-subject variance. In general, the random effects model is an extension of the ANOVA, which can adjust for many more covariates; however, the description of this approach is beyond the scope of this thesis.

Confidence intervals with multiple measurements per individual
When multiple observations are recorded within each subject, the differences between the two measurement systems cannot be assumed independent as in the case of single measurement per subject. The risks of calculating the 95% CI of the LOA using the original (independence-based) bounds derived in Bland & Altman (1986) have been reported by Hamilton & Lewis (2010). Consequently, the calculation of the CI must be adapted to this more general case.

In their paper, Bland and Altman provided an approach to calculate CI for the case when the true value is constant (Bland & Altman 1999). However, no equations were given to calculate the CI when the true value changes. The authors only advised to review a publication by Burdick and Graybill (Burdick & Graybill 1992) on how to calculate CI for combinations of components of variance. In a subsequent paper, Zou proposed to calculate the CI with multiple measurements per individual using the delta method (Zou 2011). This approach was used in the agreement study reported in Chapter 9 of this thesis. The section 7.2 Statistical methods reports the analytical derivation of the LOA and of their 95% CI when multiple observations per individual are available and the true value varies.

Best practices when performing a Bland-Altman analysis of agreement
Based on the guidelines published by Bland and Altman in their papers, and analysing a large number of agreement studies found in literature, Santos (2011) reported a list of best practices to be adopted when carrying out a Bland-Altman analysis of agreement. A summary of this list, combined with advice from other studies, is reported in this section and can be summarised as follows:

1. A scatter plot of the difference between the methods against their mean must be reported, ensuring there is no relation between the difference and the mean.
2. The assumption that differences are likely to follow a normal distribution must be verified by looking at their histogram.
3. The formula of the LOA must be reported, and the LOA must be plotted on top of the scatter diagram. In case of multiple measurements per subjects, the modified LOA presented in Bland & Altman (1999) and Bland & Altman (2007) must be used.
4. The CI must be calculated and reported using the appropriate relation based on the study design.
5. It should be stated that since the differences are normally distributed, 95% of differences will lie between the limits of agreement
6. The outcome should be interpreted, answering questions such as, for example: are the differences significative? Is the new method of measurement appropriate for the intended use?
7.2 Statistical methods
In the following sections, the statistical methods utilised for data analyses in the present thesis are discussed in detail.

7.2.1 Sample size calculation
The Bland-Altman analysis of agreement has been widely employed to compare a novel clinical measurement device against an established one. Validation studies of such devices often involve human participants. A typical requirement for obtaining approval by a research ethics committee for studies involving human subjects is to calculate the sample size (i.e. the minimum number of participants required for a statistically meaningful results). Using the minimum number of subjects is especially important when dealing with clinical (i.e. non-healthy) patients.

The validation study presented in Chapter 9 of this thesis featured a more complex design compared to the cases discussed by Bland and Altman, even in their follow-up papers (Bland & Altman 1999; Bland & Altman 2007). In this study, participants ambulated on a treadmill at 4 different speeds. Three-dimensional coordinates of 3 different markers attached to the lower limbs were continuously captured during an interval of 15 s per subject. This data collection protocol presented multiple measurements per subject while the true quantity (i.e. marker position) varied. Furthermore, the outcome variable was affected by autocorrelation (i.e. the repeated differences for a single subject were not independent). The effects of both repeated measurements and autocorrelation were modelled in the proposed statistical analysis, and the 95% CI were calculated accordingly (see subsection Multiple measurements per individual in section 7.1.2), using the delta method (Zou 2011).

However, because no equation was provided by Bland and Altman (or other authors) for the calculation of the sample size with more complex study designs, the sample size in this thesis was calculated using the original method suggested by Bland (2004) for agreement studies with simpler design. This method is based on selecting a reasonable value for the 95% CI of the LOA between Kinect- and Vicon-derived measurements. Since this approach does not consider the complexity of the present study design (i.e. multiple measurements per subject and autocorrelation), a conservative (i.e. small) CI value was selected. The CI of the LOA was calculated as:

$$CI = \pm 1.96 \sqrt{3/n} \cdot s$$  

(7.4)

where $s$ is the standard deviation of the differences between Kinect v2 and Vicon measurements and $n$ is the sample size for Kinect v2 data. Previous studies on the characterization of Kinect v2 depth map (see Chapter 3) estimated a standard deviation $s \approx 5$ mm for the depth measurements, therefore this value was used in Eq. 7.4 (Lachat, Macher, Mittet, et al. 2015; Lachat, Macher, Landes, et al. 2015; Corti et al. 2015).

Kinect has a sampling frequency of 30 fps. Considering a duration of 15 seconds per trial, 1 trial per subject, and a total of 20 subjects, the following number of samples (i.e. frames) $n$ was obtained:

$$n = 15 \times 20 = 300$$

30 The difference between Kinect- and Vicon-derived measurements.
31 It should be noted that Kinect v2 and Vicon initially had different sample sizes, due to their different capture framerates (30 versus 120 fps). This difference was eliminated by down-sampling Vicon data during data analysis.
\[ n = 30 \text{ fps} \cdot 15 \text{ s trial} \cdot 1 \frac{\text{trials}}{\text{subject}} \cdot 20 \text{ subjects} = 9,000 \text{ samples} \]

which gave the following CI:

\[ \text{CI} = \pm 1.96 \sqrt{3/9,000 \cdot 5} = \pm 0.18 \text{ mm}. \]

Considering that the expected accuracy of Kinect v2 is in the order of 5 mm, the calculated CI is quite conservative, meaning that a sample size of 20 subjects ensures sufficient significance for the proposed validation study.

7.2.2 LOA with multiple observations per individual

Let \( D_{it} \) be the \( t^{th} \) difference (KCMT – Vicon) for the \( i^{th} \) individual. A model that takes into account the structure of the data is:

\[ D_{it} = \mu_D + S_i + E_{it} \]

where \( \mu_D \) is the overall mean, \( S_i \) is a random between-subject effect for subject \( i \), and \( E_{it} \) is a random within-subject error term.

It is assumed that \( S_i \) and \( E_{it} \) are independent from each other, have mean zero and variance \( \sigma_S^2 \) and \( \sigma_E^2 \) respectively. To allow for autocorrelation, an AR(1) (auto-regression order 1) model was fitted: this model assumes that the correlation between \( E_{it} \) and \( E_{i,t-1} \) (i.e., a certain measurement and the measurement recorded at the previous instant) is equal to \( \rho \), which was estimated.

Since \( D_{it} \) has mean \( \mu_D \) and variance \( \sigma_S^2 + \sigma_E^2 \), the limits of agreement (LOA) can be calculated as:

\[ \text{LOA} = \hat{\mu}_D \pm 1.96 \sqrt{\hat{\sigma}_S^2 + \hat{\sigma}_E^2} \]

where the ^ notation denotes sample estimates of population parameters. Using another suitable notation:

\[ \text{LOA} = \bar{D} \pm 1.96 \sqrt{S_S^2 + S_E^2} \]

where \( S_S^2 \) is the between-subject variation and \( S_E^2 \) is the within-subject variation. Using software that can fit a random effect model with auto-correlation within subjects, it is possible to obtain estimates of bias and LOA. In the present study, GenStat (VSNi, UK) was used. It should be noted that to fit these models, it is necessary to have the same number of frames per subject, i.e. data must be balanced. For linear mixed-effects (LME) models, see book by Pinheiro & Bates (2000).

7.2.3 Confidence intervals of the LOA with multiple observations per individual

To evaluate the precision of the LOA with multiple observations per individual, their approximate 95% CI must be calculated. The estimates of the LOA are defined as:

\[ \text{LOA} = \bar{D} \pm 1.96 \sqrt{\hat{\phi}}, \]

where \( \hat{\phi} = S_S^2 + S_E^2. \bar{D}, S_S^2 \) and \( S_E^2 \) are all mutually independent, so:
\[
\text{var} \left( \bar{D} \pm 1.96 \sqrt{\hat{\phi}} \right) = \text{var}(\bar{D}) + 1.96^2 \text{var} \sqrt{\hat{\phi}}.
\]

Because of the independence of \(S_S^2\) and \(S_E^2\),

\[
\text{var}(\hat{\phi}) = \text{var}(S_S^2) + \text{var}(S_E^2).
\]

The so-called delta method for approximating the variance of a function of a random variable is based on a Taylor series expansion. If the random variable \(Y\) has mean \(\mu_Y\) and variance \(\sigma_Y^2\), and \(\psi(Y)\) is a function of \(Y\), then:

\[
\text{var}[\psi(Y)] \approx [\psi'(\mu_Y)]^2 \sigma_Y^2.
\]

If \(\psi(Y) = \sqrt{Y} = Y^{1/2}\), it follows that:

\[
\text{var}[Y^{1/2}] \approx \left[\frac{\mu_Y^{1/2}}{2}\right]^2 \sigma_Y^2.
\]

Applying this to the limits of agreement problem, we have that:

\[
\text{var} \left( \bar{D} \pm 1.96 \sqrt{\hat{\phi}} \right) \approx \text{var}(\bar{D}) + 1.96^2 \left[\frac{\mu_{\hat{\phi}}^{1/2}}{2}\right]^2 \text{var}(\hat{\phi}) =
\]

\[
= \text{var}(\bar{D}) + 1.96^2 \left[\frac{\mu_{\hat{\phi}}^{1/2}}{2}\right]^2 \left(\text{var}(S_S^2) + \text{var}(S_E^2)\right) =
\]

\[
= \text{var}(\bar{D}) + 1.96^2 \left[\text{var}(S_S^2) + \text{var}(S_E^2)\right] \frac{4\mu_{\hat{\phi}}}{S_S^2 + S_E^2}.
\]

Eq. 7.5

Of course, none of the true values in this equation is available, only their estimates are. The expected value of \(\hat{\phi}, \mu_{\hat{\phi}} = \phi\), so \(\hat{\phi}\) is used as the estimate of \(\mu_{\hat{\phi}}\).

The software that fits the models provides standard errors for \(\bar{D}\) and for the two components of variance, \(S_S^2\) and \(S_E^2\). Substituting these into Eq. 7.5:

\[
\text{var} \left( \bar{D} \pm 1.96 \sqrt{\hat{\phi}} \right) \approx \text{[SE}(\bar{D})]^2 + 1.96^2 \left[\frac{[\text{SE}(S_S^2)]^2 + [\text{SE}(S_E^2)]^2}{4(S_S^2 + S_E^2)}\right].
\]

It follows from this that the margin of error (MOE)\(^{32}\) for the 95% CI for the LOA is approximately:

\[
\text{MOE} = 1.96 \cdot \sqrt{[\text{SE}(\bar{D})]^2 + 1.96^2 \left[\frac{[\text{SE}(S_S^2)]^2 + [\text{SE}(S_E^2)]^2}{4(S_S^2 + S_E^2)}\right]}.
\]

Eq. 7.6

7.3 Conclusion

In the present chapter, different approaches to the analysis of agreement between a novel and an established measurement system were discussed. In 1986, Bland and Altman have addressed
this problem presenting a simple yet effective statistical technique, now widely known as the Bland-Altman analysis of agreement. More than 30 years later, alternative methods are still incorrectly applied to determine the agreement between two measurement systems. The most commonly used methods were presented above, together with their pitfalls when applied to agreement studies.

Compared to other approaches, the Bland-Altman analysis of agreement has several advantages: i) it doesn’t depend on the range of the sample, ii) it separates the systematic and the random component of the error, iii) its results are expressed in the same scale as the measured quantity, iv) the Bland-Altman plot allows the researcher to easily identify outliers and trends in the data, and v) this technique can be easily explained to non-statisticians. For these reasons, the Bland-Altman analysis of agreement was selected for use in the present thesis.

To cater for the study design presented in Chapter 9 of this thesis, a variant of the original Bland-Altman (1986) method was also presented in this chapter. Specifically, this approach accounts for the effects of multiple measurements per individual and of autocorrelation between consecutive measurements. The following chapters (8 and 9) present two examples of use of the Bland-Altman technique (original and modified, respectively), used to assess the agreement between the novel KCMT system (at two different stages of development) and an established Vicon motion capture system.

7.4 Appendix
7.4.1 Confidence intervals of the LOA
Given the following definition of the LOA:

\[
\text{LOA} = \bar{d} \pm 2s
\]

it is possible to calculate their 95% CI. Because \( \bar{d} \) and \( s \) are statistically independent, the variance of the LOA can be calculated as:

\[
\text{var}(\bar{d} + 2s) = \text{var}(\bar{d}) + 4 \text{var}(s) = \frac{\sigma^2}{n} + 4 \text{var}(s)
\]

For a random sample from a Normal distribution:

\[
\text{var}(s^2) = \frac{2\sigma^4}{n - 1}
\]

If the random variable \( Y \) has mean \( \mu \) and variance \( \phi^2 \), then:

\[
\text{var}[\psi(Y)] \approx [\psi'(\mu)]^2 \phi^2
\]

Let’s define \( Y = s^2 \). Then \( \mu(Y) = \mu(s^2) = \sigma^2 \) and:

\[
\text{var}(Y) = \text{var}(s^2) = \frac{2\sigma^4}{n - 1}
\]

Hence:

\[
\text{var}(s) = \text{var}(\sqrt{Y}) = \text{var} \left( Y^{\frac{1}{2}} \right) \approx \\
\approx \left( \frac{1}{2} (\sigma^2)^{\frac{1}{2}} \cdot \frac{1}{2} \right)^2 \cdot \text{var}(Y) =
\]

149
\[
\frac{1}{4\sigma^2} \cdot 2\sigma^4 = \frac{\sigma^2}{2(n-1)} \approx \frac{\sigma^2}{2n}
\]

Therefore:

\[
\text{var}(s) \approx \frac{\sigma^2}{2n} \quad (7.8)
\]

Substituting Eq. 7.8 into Eq. 7.7, we obtain:

\[
\text{var}(\bar{d} + 2s) \approx \frac{\sigma^2}{n} + \frac{2\sigma^2}{n} = \frac{3\sigma^2}{n}
\]

Hence:

\[
\text{SE}(\bar{d} + 2s) = \sqrt{\frac{3\sigma^2}{n}}
\]

which is the result asserted by Bland and Altman for the SE of the LOA (Bland & Altman 1986).

### 7.5 References


Statistical Methods in Medical Research, 8(2), pp.135–160. Available at: http://smm.sagepub.com/content/8/2/135.short.


CHAPTER 8
Agreement between KCMT and Vicon for measuring joint angles and moments in the sagittal plane during single-leg squat

8.1 Abstract
Increased risk of ACL injury can be estimated based on analysis of lower limb joint angles and moments measured during a single-leg squat (SLS) task. However, expensive motion capture systems are required to measure these variables. To make this analysis accessible in a clinical context, a novel low-cost tracking methodology was developed. This methodology was denoted as the *Kinect coloured marker tracking* (KCMT) system. The objective of the study reported in this chapter was threefold: i) to assess the accuracy of sagittal joint angles and moments of the lower limbs, calculated using marker trajectories measured by KCMT; ii) to verify if KCMT-derived sagittal joint angles were more accurate compared to those obtained in previous studies from the proprietary Kinect v2 markerless pose estimation algorithm; iii) to verify if KCMT-derived sagittal joint angles were sufficiently accurate to allow discerning individuals at risk of ACL injury from those not at risk.

Eleven healthy participants were asked to perform SLS trials in barefoot condition, while standing on a force platform. Marker trajectories were simultaneously recorded using KCMT and a Vicon motion capture system, used as gold standard. Hip, knee and ankle joint angles from the two systems were calculated via inverse kinematics and joint moments were determined via inverse dynamics. A Bland-Altman analysis of agreement between KCMT- and Vicon-derived variables was carried out for each joint angle and joint moment measured on the sagittal plane.

The limits of agreement (LOA) of joint angles were −16°, 13° for hip flexion, −12°, 0° for knee flexion and −12°, 9° for ankle flexion. The LOA of joint moments were −15, 23 N m for hip flexion, −6, 19 N m for knee flexion and −12, 5 N m for ankle flexion. Biases were −2°, −6° and −2° for hip, knee and ankle flexion angle, and 4, 6 and −3 N m for hip, knee and ankle flexion moment, respectively.

The results of this work indicate that marker trajectories measured by KCMT can be used in a musculoskeletal modelling workflow and that the agreement with Vicon is joint-dependent. Comparison with previous studies from literature suggests that KCMT-derived kinematic measurements are more accurate than those obtained from the Kinect v2 markerless pose estimation algorithm. However, based on the results of the Bland-Altman analysis, further work is required for the novel methodology to replace conventional marker-based motion capture systems for the identification of ACL injury risk.
8.2 Introduction

Different functional tests involving the lower limbs can be used to assess the risk of ACL injury in an individual. As discussed in Chapter 1 of this thesis, two of these tasks, the drop vertical jump (DVJ) and the single-leg squat (SLS), have been previously studied using conventional marker-based optical motion capture systems and analysed by means of musculoskeletal modelling software. However, laboratory-grade motion capture systems are not always viable due to their high cost. The screening of large cohort of young participants could be more easily conducted on-site (e.g. in schools or sports clubs), rather than in a specialised laboratory, where each child is typically accompanied by his/her parents. On-site motion analysis requires a portable and affordable motion capture system, accurate enough to distinguish between healthy and high-risk patterns of lower limb biomechanical variables. To provide a solution to this problem, a novel low-cost motion capture system was developed in this thesis. This system was based on Kinect for Windows v2 (Microsoft, Redmond, US) and was denoted as Kinect coloured marker tracking (KCMT) system (see Chapter 4).

The aim of the present study was to quantify the accuracy of kinematic and dynamic variables derived from coloured marker trajectories measured by KCMT. A Vicon marker-based motion analysis system was used as gold standard. The SLS was selected as task for this study for two reasons: first, because it was previously used to predict the ACL injury risk (Zeller et al. 2003; Pantano et al. 2005; Willson et al. 2006; Yamazaki et al. 2009); second, because it involves lower motion speed compared to more challenging tasks such as landing, hence allowing the use of a device – Kinect v2 – with lower capture frame rate (30 fps) compared to conventional motion capture systems.

The objective of this study was threefold: i) to assess the accuracy of sagittal joint angles and moments of the lower limbs, calculated using marker trajectories measured by KCMT; ii) to verify if KCMT-derived sagittal joint angles were more accurate compared to those obtained in previous studies from the proprietary Kinect v2 markerless pose estimation algorithm; iii) to verify if KCMT-derived sagittal joint angles were accurate enough to allow discerning individuals at risk of ACL injury from those not at risk.

8.3 Methods

8.3.1 Participants

In this study, participants were recruited from a large group of young female individuals who were already participating in a larger study about the effect of lower limb dominance on ACL injury risk (Sayer et al. 2016). Data collection for the larger study (denoted as the ACL study) involved assessment of 103 healthy female participants over a period of 12 months between 2015 and 2016. The inclusion criteria were: (i) aged 7-25 years old; (ii) participating in weekly physical activity; and (iii) healthy weight (i.e., body mass index <30 kg/m²). Exclusion criteria for participation were: i) body mass index (BMI) greater than 25, ii) history of lower limb surgery or medically diagnosed ACL and/or meniscal injury, iii) other medical condition that currently affects the ability to perform running and jumping tasks and iv) use of orthotic devices (medically prescribed or otherwise) in footwear to correct foot posture (i.e., flat foot) during gait. The study was approved by the Human Research Ethics Committee at The University of Melbourne (ID 1442604) and participants provided written informed consent.

While all participants were recorded using a conventional marker-based motion analysis system, 75 of them were also concurrently recorded using the novel KCMT system. Because the
development of the KCMT software was carried out during the period of data collection of the ACL study, the KCMT markerset and algorithms were modified several times during this time span. To ensure maximum consistency in the analysed dataset, only SLS trials captured using the final KCMT software version and markerset (v1.6) were considered in the present study, meaning that only 14 participants were included. Of these 14 participants, one was excluded because left-leg dominant, to simplify the processing pipeline. Some of the remaining trials presented abnormal noise in the trajectories of the Metatarsus marker tracked by KCMT, probably due to colour similarity between this marker and the carpet covering the force platform. Considering all these exclusion criteria, a total of 31 trials from 11 female participants (11±2 years, 38±14 kg, 146±12 cm, all right-leg dominant) was included in this study.

8.3.2 Experimental protocol

Trials were performed in the motion capture laboratory of the Centre for Health, Exercise & Sports Medicine (CHESM) of The University of Melbourne. Participants were asked to perform a SLS in barefoot condition, standing with the dominant leg on a force plate, and completing the task within a 5-s period. To ensure appropriate depth of the SLS, a plinth was placed behind participants. Motion data were simultaneously recorded using a Vicon motion capture system (Vicon Motion Systems Ltd, Oxford, UK) and KCMT. Start and stop commands were verbally provided to the participants by a trained member of the research group, who was also in charge of controlling the Vicon workstation.

The Vicon system was composed of 12 cameras (8 MX F20 and 4 MX F40) sampling at 120 Hz and a synchronized force plate (ORS-6-2000, AMTI Watertown, MA, U.S.A.) sampling at 2400 Hz. Vicon data were collected, labelled and cleaned using Vicon Nexus v1.8.5 software. Vicon markerset, denoted as Schache-CHESM v0.92F, was composed of 46 reflective markers (⌀14 mm), 8 of which were only used in static trials for model calibration (see Figure 57 and Table 10).

Figure 57. Vicon markerset, denoted as Schache-CHESM v0.92F. Note: this image is for reference only; trials analysed in this study were collected in barefoot condition.
The KCMT system was based on the software application described in Chapter 4, a Kinect v2 sensor (Microsoft, Redmond, US) sampling at 30 Hz, and custom-made coloured markers (⌀38 mm). Because these markers were identified by their colour, substantially different hues were selected to avoid confusion between them, thus limiting their maximum number to 7 (see Chapter 4 for details). To compensate for this limited markerset, 4 virtual joint centres provided by the Kinect v2 markerless pose estimation algorithm were included in the KCMT v1.6 markerset. These points were denoted as SpineBase, ShoulderLeft, ShoulderRight and SpineShoulder (Figure 58 and Table 11).

Kinect v2 was positioned in front of the participant, and the distance was set to ensure that the participant was always fitting within the field of view (FOV) of Kinect, even when performing a drop vertical jump (DVJ) task. Indeed, although not analysed in this study, the DVJ task was part of the ACL study and was captured using the KCMT as well.

The Vicon markerset was established before the start of data collection for the ACL study, whereas the KCMT software and markerset were still under development during data collection. Due to the constraint of predefined positions of the Vicon markerset, the body locations available to attach KCMT coloured markers were limited.
Table 10. Vicon Schache-CHESM v0.92F markerset was composed of 46 reflective markers, 8 of which (indicated by a *) were only used in static trials

<table>
<thead>
<tr>
<th>Marker name</th>
<th>Anatomical position</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Trunk</strong></td>
<td></td>
</tr>
<tr>
<td>MAN</td>
<td>Jugular notch</td>
</tr>
<tr>
<td>T2</td>
<td>2\textsuperscript{nd} thoracic vertebrae</td>
</tr>
<tr>
<td>T10</td>
<td>10\textsuperscript{th} thoracic vertebrae</td>
</tr>
<tr>
<td><strong>Pelvis</strong></td>
<td></td>
</tr>
<tr>
<td>LASI/RASI</td>
<td>Anterior superior iliac spine</td>
</tr>
<tr>
<td>SACR</td>
<td>Midpoint of both PSIS</td>
</tr>
<tr>
<td><strong>Thigh</strong></td>
<td></td>
</tr>
<tr>
<td>LTHAP/RTHAP</td>
<td>Proximal anterior thigh</td>
</tr>
<tr>
<td>LTHLP/RTHLP</td>
<td>Proximal lateral thigh</td>
</tr>
<tr>
<td>LTHAD/RTHAD</td>
<td>Distal anterior thigh</td>
</tr>
<tr>
<td>LTHLD/RTHLD</td>
<td>Distal lateral thigh</td>
</tr>
<tr>
<td>LPAT/RPAT</td>
<td>Patella</td>
</tr>
<tr>
<td>LLEPI/RLEPI</td>
<td>Lateral epicondyle of knee</td>
</tr>
<tr>
<td>LMEPI/RMEPI*</td>
<td>Medial epicondyle of knee</td>
</tr>
<tr>
<td><strong>Tibia</strong></td>
<td></td>
</tr>
<tr>
<td>LTIAP/RTIAP</td>
<td>Proximal anterior tibia (1/3 from lateral epicondyle to lateral malleolus)</td>
</tr>
<tr>
<td>LTLAT/RTLAT</td>
<td>Lateral tibia (1/2 from lateral epicondyle to lateral malleolus)</td>
</tr>
<tr>
<td>LTIAD/RTIAD</td>
<td>Distal anterior tibia (2/3 from lateral epicondyle to lateral malleolus)</td>
</tr>
<tr>
<td>LLMAL/RLMAL</td>
<td>Lateral malleolus</td>
</tr>
<tr>
<td>LMMAL/RMMAL*</td>
<td>Medial malleolus</td>
</tr>
<tr>
<td><strong>Foot</strong></td>
<td></td>
</tr>
<tr>
<td>LMFL/RMFL</td>
<td>Mid foot lateral</td>
</tr>
<tr>
<td>LMFS/RMFS</td>
<td>Mid foot superior</td>
</tr>
<tr>
<td>LTOE2/RTOE2</td>
<td>Big toe nail</td>
</tr>
<tr>
<td>LHEEL/RHEEL</td>
<td>Distal calcaneus</td>
</tr>
<tr>
<td>LHEEL2/RHEEL2*</td>
<td>Proximal calcaneus</td>
</tr>
<tr>
<td>LTOE/RTOE*</td>
<td>3\textsuperscript{rd} metatarsophalangeal joint</td>
</tr>
<tr>
<td>LMT5/RMT5</td>
<td>5\textsuperscript{th} metatarsophalangeal joint</td>
</tr>
<tr>
<td>LMT1/RMT1</td>
<td>1\textsuperscript{st} metatarsophalangeal joint</td>
</tr>
</tbody>
</table>
Figure 58. A fully marked-up participant, featuring the markerset v1.6 tracked by KCMT. The crosses indicate the body positions of the 4 virtual joint centres tracked by the Kinect v2 pose estimation algorithm and used as part of the KCMT markerset in this study. Part of Vicon reflective markerset is also visible.

Table 11. KCMT markerset v1.6 included 4 virtual joints centres tracked by Kinect v2 pose estimation algorithm and 7 coloured markers

<table>
<thead>
<tr>
<th>Marker name</th>
<th>Type/colour</th>
<th>Anatomical position</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShoulderLeft</td>
<td>Virtual joint centre</td>
<td>Left shoulder joint</td>
</tr>
<tr>
<td>SpineShoulder</td>
<td>Virtual joint centre</td>
<td>Midpoint between the shoulders</td>
</tr>
<tr>
<td>ShoulderRight</td>
<td>Virtual joint centre</td>
<td>Right shoulder joint</td>
</tr>
<tr>
<td>SpineBase</td>
<td>Virtual joint centre</td>
<td>Midpoint between the hips</td>
</tr>
<tr>
<td>ASISLeft</td>
<td>Magenta</td>
<td>Left anterior superior iliac spine</td>
</tr>
<tr>
<td>ASISRight</td>
<td>Green</td>
<td>Right anterior superior iliac spine</td>
</tr>
<tr>
<td>Thigh</td>
<td>Dark blue</td>
<td>Midpoint of the thigh, anterior aspect</td>
</tr>
<tr>
<td>KneeLat</td>
<td>Yellow</td>
<td>Lateral epicondyle of the femur</td>
</tr>
<tr>
<td>Tibia</td>
<td>Red</td>
<td>3/4th of tibia length, anterior aspect</td>
</tr>
<tr>
<td>Malleolus</td>
<td>Fluorescent blue</td>
<td>Medial aspect of malleolus</td>
</tr>
<tr>
<td>Metatarsus</td>
<td>Fluorescent green</td>
<td>Distal midpoint between 1st and 5th metatarsus</td>
</tr>
</tbody>
</table>

Figure 59 shows a comparison between Vicon and KCMT markersets attached to the respective OpenSim (Delp et al. 2007) models. It should be noted that the model used for the KCMT included the shoulder joints, as opposed to the model used for Vicon. Because only the right leg was tracked by KCMT, the left leg and the arms were hidden in the corresponding model.
To perform inverse dynamics (ID) combining KCMT-derived joint angles and force platform measurements, a coordinate transformation from KCMT to Vicon coordinate system was required (see Chapter 5). For this reason, a calibration trial was recorded before each capture session using KCMT. This trial contained the coordinates of 4 coloured markers attached to a custom-made rigid L-frame and aligned with Vicon X and Y axes. Furthermore, to scale the generic musculoskeletal models to match the subject-specific anthropometric measurements, a static trial was recorded for each participant using both motion capture systems.

8.3.3 Data analysis
Data analysis was carried out by means of a custom MATLAB pipeline (R2016a, MathWorks, Natick, MA, US). All functionalities provided by OpenSim and mentioned in the following sections were accessed via the OpenSim v.3.3 API (application programming interface) for MATLAB.

Pre-processing
Marker trajectories and ground reaction forces (GRF) recorded by the Vicon system were exported to .c3d file format using Vicon Nexus software. Trials were then converted to OpenSim file formats (.trc for marker trajectories, .mot for GRF) using a customized version of MATLAB OpenSim Tools v2 (Lichtwark et al. 2014). Marker trajectories were filtered using a zero-phase, 4th order Butterworth filter, with a low-pass cut-off frequency of 12 Hz.

KCMT marker trajectories were directly exported to .trc file format and gaps were filled using spline interpolation. To remove unwanted spikes, a median filter with span 5 was applied, followed by the same Butterworth filter described above. Then, KCMT data were rigidly transformed to Vicon coordinate system, using the algorithms described in Chapter 5 and the

---

33 Gaps in coloured marker trajectories corresponded to 0.1% of the captured frames.
calibration trial containing coordinates of the L-frame. A further rotation was then applied to both Vicon and KCMT data, to transform coordinates from Vicon reference system (featuring Z as vertical axis) to OpenSim reference system (featuring Y as vertical axis, see Chapter 5 for more details).

OpenSim models

The model used to analyse Vicon data (denoted as gait_2392_simbody_Timmi) was a modified version of the standard OpenSim Gait2392 model. Some parameters of this model were customized to improve the scaling and inverse kinematics (IK) results of this study. The ranges of motion (ROMs) of lower limb joint coordinates – which were originally unrestricted – were limited by enabling their “clamped” attribute in the XML file of the model. Moreover, their default ROMs were reduced based on a trial and error approach to avoid incongruent (i.e. non-physiological) solutions to the IK problem (Table 12). Furthermore, the metatarsophalangeal (MTP) joint was locked at 0°. It is interesting to note that the original OpenSim model featured different ROMs between left and right side for some joint coordinates.

Table 12. Joint coordinates of the OpenSim Gait2392 model, customized for use with Vicon data. All coordinates were clamped and the corresponding ROMs were reduced in almost all cases, to avoid incongruent solutions to the IK problem.

<table>
<thead>
<tr>
<th>Joint coordinate</th>
<th>Locked</th>
<th>Clamped</th>
<th>Original ROM</th>
<th>Modified ROM</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Hip flexion</td>
<td>False</td>
<td>True</td>
<td>−120°, 120°</td>
<td>−60°, 120°</td>
<td></td>
</tr>
<tr>
<td>Hip adduction</td>
<td>False</td>
<td>True</td>
<td>−120°, 120°</td>
<td>−90°, 45°</td>
<td></td>
</tr>
<tr>
<td>Hip rotation</td>
<td>False</td>
<td>True</td>
<td>−120°, 120°</td>
<td>−70°, 70°</td>
<td></td>
</tr>
<tr>
<td>Knee flexion</td>
<td>False</td>
<td>True</td>
<td>−120°, 10°</td>
<td>/</td>
<td></td>
</tr>
<tr>
<td>Ankle flexion</td>
<td>False</td>
<td>True</td>
<td>−90°, 90°</td>
<td>−60°, 60°</td>
<td>Left side ROM was already −60°, 60°</td>
</tr>
<tr>
<td>Subtalar angle</td>
<td>False</td>
<td>True</td>
<td>−90°, 90°</td>
<td>−60°, 60°</td>
<td>Left side ROM was already −60°, 60°</td>
</tr>
<tr>
<td>MTP angle</td>
<td>True</td>
<td>True</td>
<td>−90°, 90°</td>
<td>−60°, 60°</td>
<td></td>
</tr>
<tr>
<td>Lumbar extension</td>
<td>False</td>
<td>True</td>
<td>−90°, 90°</td>
<td>−90°, 50°</td>
<td></td>
</tr>
<tr>
<td>Lumbar bending</td>
<td>False</td>
<td>True</td>
<td>−90°, 90°</td>
<td>−60°, 60°</td>
<td></td>
</tr>
<tr>
<td>Lumbar rotation</td>
<td>False</td>
<td>True</td>
<td>−90°, 90°</td>
<td>/</td>
<td></td>
</tr>
</tbody>
</table>

The model used to analyse KCMT data was a customized version of Dorn (2010) which, in turn, was based on the standard Gait2392 OpenSim model\(^ {34}\). This model was selected because it included also the upper limbs, whereas the standard Gait2392 model only included the lower limbs. The presence of the shoulder joints was essential because virtual shoulder joint centres tracked by the Kinect v2 markerless pose estimation algorithm were included in the KCMT markerset. Considering the limited number of markers constituting the KCMT markerset, the shoulder joints were included to give stability to the model during inverse kinematics analyses.

The limited KCMT markerset also explains why the subtalar angle and the MTP angle were locked in their default orientation of 0°, to prevent non-physiological IK results. For the same reason, the ROMs of the lower limb joint coordinates were reduced in almost all cases (Table 13). In

\(^{34}\) It should be noted that, when an older model is loaded, OpenSim performs the automatic upgrade of the model XML file to the latest version. This may complicate the text (i.e. line by line) comparison between different versions of the same model.
contrast, the clamped parameter of the pelvis rotation coordinate was changed from true to false, to allow any orientation of the model around the global vertical axis.

Table 13. Joint coordinates of the OpenSim Gait2392 model, customized for use with KCMT data. All coordinates were clamped and the corresponding ROMs were reduced in almost all cases, to avoid incongruent solutions to the IK problem.

<table>
<thead>
<tr>
<th>Coordinate</th>
<th>Locked</th>
<th>Clamped</th>
<th>Original ROM</th>
<th>Modified ROM</th>
<th>Notes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pelvis tilt</td>
<td>False</td>
<td>True</td>
<td>−90°, 90°</td>
<td>−60°, 60°</td>
<td></td>
</tr>
<tr>
<td>Pelvis list</td>
<td>False</td>
<td>True</td>
<td>−90°, 90°</td>
<td>−60°, 60°</td>
<td></td>
</tr>
<tr>
<td>Pelvis rotation</td>
<td>False</td>
<td>False</td>
<td>−90°, 90°</td>
<td>/</td>
<td>Originally clamped</td>
</tr>
<tr>
<td>Hip flexion</td>
<td>False</td>
<td>True</td>
<td>−120°, 120°</td>
<td>−45°, 90°</td>
<td></td>
</tr>
<tr>
<td>Hip adduction</td>
<td>False</td>
<td>True</td>
<td>−120°, 120°</td>
<td>−60°, 45°</td>
<td></td>
</tr>
<tr>
<td>Hip rotation</td>
<td>False</td>
<td>True</td>
<td>−120°, 120°</td>
<td>−60°, 60°</td>
<td></td>
</tr>
<tr>
<td>Knee flexion</td>
<td>False</td>
<td>True</td>
<td>−120°, 10°</td>
<td>/</td>
<td></td>
</tr>
<tr>
<td>Ankle flexion</td>
<td>False</td>
<td>True</td>
<td>−90°, 90°</td>
<td>−60°, 60°</td>
<td>Left side ROM was already −60°, 60°</td>
</tr>
<tr>
<td>Subtalar angle</td>
<td>True</td>
<td>True</td>
<td>−90°, 90°</td>
<td>−60°, 60°</td>
<td>Was already locked</td>
</tr>
<tr>
<td>MTP angle</td>
<td>True</td>
<td>True</td>
<td>−90°, 90°</td>
<td>−60°, 60°</td>
<td>Was already locked. Left side ROM was already −60°, 60°.</td>
</tr>
<tr>
<td>Lumbar extension</td>
<td>False</td>
<td>True</td>
<td>−90°, 90°</td>
<td>−45°, 45°</td>
<td></td>
</tr>
<tr>
<td>Lumbar bending</td>
<td>False</td>
<td>True</td>
<td>−90°, 90°</td>
<td>−45°, 45°</td>
<td></td>
</tr>
<tr>
<td>Lumbar rotation</td>
<td>False</td>
<td>True</td>
<td>−90°, 90°</td>
<td>−60°, 60°</td>
<td></td>
</tr>
</tbody>
</table>

Scaling

Using OpenSim API for MATLAB, the generic models were scaled to match each participant anthropometric measurements (obtained from static trials) and body mass. Following OpenSim best practices for scaling (OpenSim 2017), anatomical markers (i.e. markers attached to bony landmarks) were given greater weights compared to motion markers (markers not attached to bony landmarks). All scaling factors are reported in section 8.7.1 OpenSim scaling parameters. To minimize the inaccuracies introduced by Kinect v2-derived virtual joint centres, lower weights were attributed to these points during scaling, compared to the weights used for coloured markers.

Body segments can be scaled uniformly or non-uniformly. With uniform scaling, a segment is scaled by the same factor along all 3 directions (x, y and z). With non-uniform scaling, a segment is scaled using distinct factors along different axes. The former approach was adopted for all segments of the Vicon model, and for all segments of the KCMT model except the torso, which was scaled non-uniformly due to the limited number of markers available.

Inverse kinematics and synchronization

Once the generic models were scaled to match the subject-specific anthropometric measurements, joint angles were determined using the OpenSim IK solver with marker trajectories from dynamic trials as input. Following OpenSim best practices for IK (OpenSim 2017), motion markers were given greater weights compared to anatomical markers (see
section 8.7.2 OpenSim inverse kinematics parameters). Joint angles obtained from paired Vicon and KCMT trials were synchronized using a custom iterative algorithm based on cross-covariance (see Chapter 6 for details). The hip, knee and ankle flexion angles were used as input for this synchronization algorithm. Before synchronization, spline interpolation was used to upsample both datasets to 480 fps, to obtain a finer synchronization step. After synchronization, both datasets were downsampled to 30 fps, to match Kinect v2 native framerate. Downsampling was achieved by keeping every $n^{th}$ sample starting with the first sample, where $n = 480 / 30$.

Inverse dynamics
Using the temporal shift calculated during synchronization of each pair of KCMT and Vicon trials, the GRF were also synchronized with the corresponding joint angles. Then, joint angles and GRF were passed as input to the OpenSim ID solver, to calculate KCMT- and Vicon-derived joint moments.

Statistical analysis
A Bland-Altman analysis of agreement (Bland & Altman 1986) was performed between paired KCMT- and Vicon-derived joint angles and moments measured in the sagittal plane. Since the differences between KCMT- and Vicon-derived measurement were normally distributed, the limits of agreement (LOA) were expected to contain 95% of these differences (see Chapter 7 for details).

In addition, the mean of each variable (e.g. joint angle or moment) was calculated for each motion capture system using data from all trials and was plotted against normalised time (Figure 62 and Figure 63). Time normalisation was achieved by resampling all trials to a fixed length of 30 samples. Specifically, resampling of joint angles and moments was performed via spline interpolation of the input arrays. In contrast, resampling of the time arrays was obtained by generating a vector of 30 evenly spaced points between 0 and the length of each input array.

In Figure 62 and Figure 63, the vertical offset between each pair of average curves represents the instantaneous bias between KCMT and the Vicon system for a specific variable. The interval given by mean ± SD was also calculated and represented as shaded area in the same plots. The larger the interval represented by the shaded area, the larger the between-trial variation of a variable for a specific motion capture system. In the ideal case of perfect agreement between KCMT and the Vicon system, the two average curves and the two shaded areas would perfectly overlap. This kind of diagram reporting mean ± SD versus trial progress is commonly used in biomechanical studies, because it displays the trend of a variable during the execution of a task. However, it is not suitable to quantify the overall agreement between two measurement methods and is hereby included only to visualize the trend of the agreement during the execution of the SLS task. As discussed in Chapter 7, the best statistical technique to assess the agreement between two methods of measurement is the Bland-Altman analysis.

8.4 Results
LOA between KCMT- and Vicon-derived joint angles were −16° to 13° for the hip, −12° to 0° for the knee and −12° to 9° for the ankle (Figure 60). Biases were −2° for the hip, −6° for the knee and −2° for the ankle. LOA of joint moments were −15 to 23 N m for the hip, −6 to 19 N m for

35 To simplify the statistical analysis, two aspects of this study design were neglected: 1) the distinction between the within-subject error and the between-subjects error; 2) the auto-correlation between consecutive measurements. Both of these aspects were accounted for in the study reported in Chapter 9, but were found to have negligible impact on the agreement results.
the knee and −12 to 5 N m for the ankle (Figure 61). Biases were 4 N m for the hip, 6 N m for the knee and −3 N m for the ankle.

Figure 60. Bland-Altman plots reporting the agreement between KCMT and Vicon for the lower limb joint angles in the sagittal plane during SLS

Figure 61. Bland-Altman plots reporting the agreement between KCMT and Vicon for the lower limb joint moments in the sagittal plane during SLS

The average curves of the joint angles (Figure 62) indicate that the instantaneous bias of the hip flexion angle between KCMT and the Vicon system was variable during the trial duration, reaching a maximum of −7° in correspondence of peak hip flexion (i.e. about 50% of the task duration). In contrast, the biases of knee and ankle flexion angles, and of all sagittal joint moments (Figure 63), were approximately constant and matched the biases observed in the Bland-Altman analysis of agreement.
Figure 62. Sagittal joint angles during SLS, obtained from KCMT (blue) and Vicon (green) and plotted against normalised time. Solid lines represent the mean of all trials, whereas the shaded areas were obtained as mean ± SD of all trials. The dots represent the minima/maxima of each averaged dataset.

Figure 63. Sagittal joint moments during SLS, obtained from KCMT (blue) and Vicon (green) and plotted against normalised time. Solid lines represent the mean of all trials, whereas the shaded areas were obtained as mean ± SD of all trials. The dots represent the minima/maxima of each averaged dataset.

8.5 Discussion

In a previous study on overhead squat (McGroarty et al. 2016), a comparison was performed between the off-the-shelf Kinect v2 and a Vicon motion capture system for measuring the peak knee flexion angle. Seven subjects performed 3 repetitions of overhead squat, while motion data were recorded by a 6-camera Vicon system and by Kinect v2 markerless pose estimation algorithm. The knee flexion was measured for both limbs as the angle between the projections of the upper and lower leg vectors (defined by the hip, knee and ankle joint centres) onto the Kinect YZ plane, where Y and Z were the vertical and depth directions respectively. Only mean and standard deviation of the absolute error between Kinect- and Vicon-derived measurements were discussed in the paper, corresponding to 6.7° ± 2.7° for the left and 7.2° ± 3.7° for the right peak knee flexion angle. However, the use of absolute values of the errors to calculate mean and standard deviation introduced an artefact in the statistical analysis, because the distribution of the error between Kinect and Vicon was incorrectly assumed to be one-sided. In fact, the raw
measurements of the peak knee flexion angle (also reported in the study) indicated that this distribution was two-sided. Therefore, to allow an unbiased comparison between the study by McGroarty et al. and the present one, a Bland-Altman analysis of agreement was performed using those raw data, resulting in LOA equal to −16°, 8° for the left and −17°, 6° for the right peak knee flexion angle, and biases of −4° and −6° respectively. Due to the limited sample size, the margin of error of the 95% confidence intervals of the LOA was ±5°. In the present study, the LOA of the knee flexion angle between KCMT and the Vicon system were −12°, 0°, i.e. narrower compared to those reported by McGroarty et al. (see column named width LOA in Table 14), whereas the bias was equal.

In a Bland-Altman analysis of agreement between Kinect v2 markerless tracking algorithm and Vicon for comfortable pace gait (Mentiplay et al. 2015), LOA of the peak knee flexion angle were 28° to 46° in correspondence of swing phase and −21° to 1° at ground contact. Biases of 37° and −10° were also reported for the two conditions, respectively. It should be noted that these results were obtained after removing outliers, hence, the actual LOA may be significantly larger. Although derived from walking rather than squatting trials, these results support the hypothesis that KCMT is more accurate than Kinect v2 pose estimation algorithm for measuring the knee flexion angle (see Table 14).

In the same study by Mentiplay et al., biases and LOA of the hip and ankle flexion angles were linearly variable instead of constant, meaning that the disagreement for those variables was proportional to the magnitude of the measured angle. Indeed, the LOA of the hip flexion angle ranged from −1°, 22° when the estimate of the true hip flexion was 30°, to −87°, −64° when the estimate of the true hip flexion was 90°. The LOA of the ankle flexion angle ranged from 25°, 53° when the estimate of the true ankle flexion was 10°, to −80°, −52° when the estimate of the true ankle flexion was 70°. In contrast, biases and LOA between KCMT and the Vicon system found in the present study were constant for all variables, meaning that their agreement was independent from the measured magnitude. This finding indicates that, although the widths of the LOA of hip and ankle flexion angles were comparable between the present study and that by Mentiplay et al. (Table 14), KCMT is significantly more accurate in measuring these joint angles compared to the Kinect v2 markerless tracking algorithm.

Table 14. Results of the Bland-Altman analysis of the joint flexion angles, observed in two previous studies using Kinect v2 markerless pose estimation algorithm (McGroarty et al. 2016; Mentiplay et al. 2015) and in the present study using KCMT (shaded rows). A Vicon system was used as gold standard in all cases.

<table>
<thead>
<tr>
<th>Study</th>
<th>Variable</th>
<th>Lower LOA</th>
<th>Upper LOA</th>
<th>Width LOA</th>
<th>Bias</th>
</tr>
</thead>
<tbody>
<tr>
<td>Mentiplay et al. 2015</td>
<td>Hip flexion excursion of the left hip during the entire phase of the left stride, when the estimate of the true value is 30°</td>
<td>−1°</td>
<td>22°</td>
<td>23°</td>
<td>11°</td>
</tr>
<tr>
<td>Mentiplay et al. 2015</td>
<td>Hip flexion excursion of the left hip during the entire phase of the left stride, when the estimate of the true value is 90°</td>
<td>−87°</td>
<td>−64°</td>
<td>23°</td>
<td>−76°</td>
</tr>
<tr>
<td>Present study</td>
<td>Right hip flexion angle</td>
<td>−16°</td>
<td>13°</td>
<td>29°</td>
<td>−2°</td>
</tr>
<tr>
<td>Study</td>
<td>Joint</td>
<td>Measurement</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>-------------------------------</td>
<td>------------------------------</td>
<td>-----------------------------------------------------------------------------</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Knee joint</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>McGroarty et al. 2016</td>
<td>Left peak knee flexion angle during overhead squat</td>
<td>$-16^\circ$ $8^\circ$ $24^\circ$ $-4^\circ$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>McGroarty et al. 2016</td>
<td>Right peak knee flexion angle during overhead squat</td>
<td>$-17^\circ$ $6^\circ$ $23^\circ$ $-6^\circ$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mentiplay et al. 2015</td>
<td>Knee flexion excursion of the left knee during the swing phase of the left stride</td>
<td>$28^\circ$ $46^\circ$ $18^\circ$ $37^\circ$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mentiplay et al. 2015</td>
<td>Knee flexion excursion of the left knee during the initial contact phase (i.e. absorption) of the left ground contact. Restricted to the first 50% of the ground contact.</td>
<td>$-21^\circ$ $1^\circ$ $22^\circ$ $-10^\circ$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Present study</strong></td>
<td>Right knee flexion angle</td>
<td>$-12^\circ$ $0^\circ$ $12^\circ$ $-6^\circ$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Ankle joint</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mentiplay et al. 2015</td>
<td>Ankle flexion excursion of the left ankle during the entire phase of the left stride, when the estimate of the true value is 10°</td>
<td>$25^\circ$ $53^\circ$ $28^\circ$ $39^\circ$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Mentiplay et al. 2015</td>
<td>Ankle flexion excursion of the left ankle during the entire phase of the left stride, when the estimate of the true value is 70°</td>
<td>$-80^\circ$ $-52^\circ$ $28^\circ$ $-66^\circ$</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td><strong>Present study</strong></td>
<td>Right ankle flexion angle</td>
<td>$-12^\circ$ $9^\circ$ $21^\circ$ $-2^\circ$</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

The comparison between the results of the present study and those reported in literature suggests that KCMT is more accurate than Kinect v2 markerless pose estimation algorithm for measuring lower limb kinematics in the sagittal plane. However, the different degrees of agreement observed for the three analysed joint angles indicate that further work is required for the proposed methodology to improve the measurement of the hip and ankle flexion angles. The use of virtual joint centres provided by the Kinect v2 markerless algorithm to track the torso may explain the lower agreement found for the hip flexion, compared to the other two joint angles. Furthermore, it could also explain the variability of the bias during the task progression observed for the hip flexion angle, but not for the other angles. A more detailed discussion on this and other limitations of the study is reported in the next section.

In a previous study of the differences in kinematics of SLS between ACL-injured patients and healthy controls (Yamazaki et al. 2009), the mean joint angle at knee maximum flexion was compared between the uninjured limb of ACL-injured patients and the dominant limb of healthy participants. Considering only female individuals for homogeneity with the present study, the
results showed statistically significant differences between ACL-injured subjects and controls of 7.7° for the knee flexion angle and of −18.1° for the hip flexion angle. In the present study, the LOA between KCMT and Vicon were −12°, 0° for the knee flexion angle and −16°, 13° for the hip flexion angle, indicating that KCMT is not accurate enough to detect significant differences for the knee flexion, and can just detect those for the hip flexion. Therefore, further work is required for this novel methodology to be used in clinical and screening settings for the identification of individuals at risk of ACL injury in a group of participants performing SLS. Improvements of Kinect v2 tracking will need to be aimed at reducing the biases and narrowing the LOA of all flexion angles.

8.5.1 Limitations

There are a number of limitations of the present study that ought to be considered.

First, 7 markers were insufficient to describe all the degrees of freedom (DOFs) of the customized OpenSim Gait2392 model used in this analysis (see section 8.7.3 Number of markers and degrees of freedom). Specifically, the lack of a third pelvis marker and the consequent use of virtual joints centres provided by the Kinect v2 markerless pose estimation algorithm caused the lower agreement observed for the hip flexion, and indirectly affected the accuracy of the other joint angles. Because joint moments were derived from joint angles via ID, this limitation affected their agreement too. There are two possible alternatives to increase the number of markers for the KCMT system. The first would be to select different colours (i.e. hues) from those already used in this study. However, due to limitations in how Kinect RGB camera perceives colours (see Chapter 4), this approach was not viable. Alternatively, a single colour for all markers could be used. This method would require the implementation of a trajectorization and labelling algorithm, to identify all markers based on their relative positions on the body rather than using their colour. This solution will be discussed in Chapter 10 of this thesis.

Second, as mentioned in section 8.3.2 Experimental protocol of the present chapter, KCMT and Vicon markers were attached to different positions on participants’ bodies. Therefore, the trajectories used to calculate joint angles via IK were different for the two systems, thus affecting their agreement. To remove this source of error in future studies, 3D-printed marker arrays may be used to support 2 reflective markers at the extremities and a coloured marker in correspondence of their midpoint (see Figure 64 and Chapter 5). Then, the trajectory of the midpoint of the two reflective markers could be compared to the trajectory of the centre of the coloured marker attached to the same array. However, such 3D-printed arrays were developed only at a later stage, for the agreement study reported in Chapter 9 of this thesis.

Figure 64. A 3D-printed marker array used in the agreement study reported in Chapter 9 of this thesis. The midpoint of the two reflective markers corresponds to the centre of the coloured marker.

Third, the position of Kinect v2 with respect to the subject was suboptimal. Because the SLS task occurred mainly on the sagittal plane, the ideal position for Kinect would have been on the side of the participant. However, because virtual joints were required to compensate for the limited
number of coloured markers, and considering that the markerless pose estimation algorithm requires the participant to face Kinect, this position could not be changed in this study.

Fourth, the distance between the participant and Kinect v2 was large. Because the original aim of the study was to capture also DVJ trials using Kinect, a sufficient distance was used to ensure that the participant was always within Kinect FOV, even when standing on a step of about 30 cm of height from the floor. Since the error affecting Kinect depth measurements increases with distance from the target (see sections 3.2.2 to 3.2.5 of this thesis), this aspect of the setup had a negative impact on the accuracy and precision of KCMT marker coordinates. This limitation can be easily addressed in future studies, avoiding landing tasks and positioning Kinect at the minimum distance which allows the participant to be within the FOV.

Fifth, it should be noted that the default, unscaled version of the OpenSim Gait2392 model represents a subject that is about 1.8 m tall and has a mass of 75.16 kg (Delp 1990; OpenSim 2018). Yet, the population of this study was composed of adolescent females. Given known differences in skeletal anatomy between males and females, such as the position of the adult hip centre location and an increased constitutional femoral valgus in females, a systematic error may have been introduced by combining motion data measured in adolescent females with the skeletal anatomy of a male individual. To minimise this error and match the anthropometry of each participant, a three-dimensional linear scaling process was applied to the generic model, using distances between markers as reference. With this approach, a scaled Gait2392 model reproducing the participant’s bone dimensions was generated for each individual.

Using generic musculoskeletal models based on measurements from cadavers of adult donors is a standard practice in musculoskeletal modelling. In gait analysis of kids, these models can provide comparable results when compared to more sophisticated (but time-consuming and expensive) subject-specific models based on magnetic resonance imaging (Correa & Pandy 2010; Correa et al. 2011). Considering the sample size of the present study, any imaging-based approach would have been impractical in terms of time and costs. Furthermore, the systematic error introduced by gender-related anthropometric differences between a participant and the corresponding scaled model can be considered negligible compared to the errors affecting Kinect v2 depth measurements (see Chapter 3) and those discussed above in this section.

Sixth, skin motion artefact may have affected marker positions relative to bones, and consequently may have contributed to the differences observed between Vicon-and KCMT-derived joint angles and moments.

8.6 Conclusion

In conclusion, the results of this study indicated that the agreement between KCMT and a conventional motion capture system for measuring lower limb kinematic and dynamic variables was joint-specific. The best agreement was achieved for the knee flexion angle, whereas further work is required to achieve the same level of agreement for hip and ankle flexion angles. The comparison between the results of the present study and those from literature indicated increased accuracy of the novel methodology relative to the proprietary Kinect v2 markerless pose estimation algorithm for measuring lower limb joint angles in the sagittal plane. However, given the typical magnitude of kinematic differences between female subjects at risk of ACL injury and controls reported in a previous study on SLS, further work is required before KCMT can be used in a clinical context for the assessment of ACL injury risk.
Future studies will be focused on addressing the limitations of the proposed methodology. The use of custom 3D-printed marker arrays will remove the component of disagreement due to different positions between corresponding coloured and reflective markers. Furthermore, the implementation of a labelling algorithm may allow to track more than 7 markers, removing the need for virtual joint coordinates provided by the Kinect v2 pose estimation algorithm. These solutions, together with an optimized position of Kinect v2, may result in more accurate measurements and in the possibility to analyse biomechanical variables also outside the sagittal plane.
8.7 Appendix

In the present Appendix to Chapter 8, all the parameters used in OpenSim to process Kinect- and Vicon-derived data were reported.

8.7.1 OpenSim scaling parameters

Table 15. Measurements used to scale the OpenSim Gait2392 model using Vicon data. The distances between pairs of markers from the Schache-CHESM v0.92F markerset were used to compute scale factors for the specified body segments along the three axes.

<table>
<thead>
<tr>
<th>Name</th>
<th>Apply</th>
<th>Marker pairs</th>
<th>Scaled segments</th>
<th>Local scaling axes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Torso</td>
<td>True</td>
<td>T2 SACR</td>
<td>Torso</td>
<td>XYZ</td>
</tr>
<tr>
<td>Pelvis</td>
<td>True</td>
<td>LASI RASI</td>
<td>Femur (both)</td>
<td>XYZ</td>
</tr>
<tr>
<td>Thigh</td>
<td>True</td>
<td>RASI RMEPI, LASI LMEPI, RASI RLEPI, LASI LLEPI, RASI RPAT, LASI LPAT</td>
<td></td>
<td></td>
</tr>
<tr>
<td>Shank</td>
<td>True</td>
<td>RMEPI RMMAL, LMEPI LMMAL, RLEPI RLMAL, LLEPI LLMAL</td>
<td>Tibia (both), Talus (both)</td>
<td></td>
</tr>
<tr>
<td>Foot</td>
<td>True</td>
<td>LHEEL2 LTOE2, RHEEL2 RTOE2</td>
<td>Calc (both), Toes (both)</td>
<td></td>
</tr>
</tbody>
</table>

Table 16. Marker weights used to scale the OpenSim Gait2392 model using Vicon data. Higher weights were given to markers placed on bony landmarks. Weights are relative, for example a weight of 1 compared to 10 is the same as 100 compared to 1000.

<table>
<thead>
<tr>
<th>Marker name</th>
<th>Weight</th>
<th>Apply</th>
</tr>
</thead>
<tbody>
<tr>
<td>MAN</td>
<td>1</td>
<td>True</td>
</tr>
<tr>
<td>T2</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>T10</td>
<td>1</td>
<td>True</td>
</tr>
<tr>
<td>LASI/RASI</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>SACR</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>LTHAP/RTHAP</td>
<td>1</td>
<td>True</td>
</tr>
<tr>
<td>LTHLP/RTHLP</td>
<td>1</td>
<td>True</td>
</tr>
<tr>
<td>LTHAD/RTHAD</td>
<td>1</td>
<td>True</td>
</tr>
<tr>
<td>LTHLD/RTHLD</td>
<td>1</td>
<td>True</td>
</tr>
<tr>
<td>LPAT/RPAT</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>LLEPI/RLEPI</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>LMEPI/RMEPI</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>LTIAP/RTIAP</td>
<td>1</td>
<td>True</td>
</tr>
<tr>
<td>LTLAT/RTLAT</td>
<td>1</td>
<td>True</td>
</tr>
<tr>
<td>LTIAD/RTIAD</td>
<td>1</td>
<td>True</td>
</tr>
<tr>
<td>LLMAL/RLMAL</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>LMMAL/RMMAL</td>
<td>1000</td>
<td>True</td>
</tr>
</tbody>
</table>

36 The files containing the OpenSim scaling parameters were denoted as Schache_CHESM_v092F_SCALE_setup_Timmi.xml and scaleSetupSingleLegRightKinEdgeV1.6Gait2392Arms.xml, and were used to analyse Vicon and KCMT data, respectively.
Table 17. Measurements used to scale the OpenSim Gait2392 model using KCMT data. The distances between pairs of markers from the KCMT v1.6 markerset were used to compute scale factors for the specified body segments along the selected axes.

<table>
<thead>
<tr>
<th>Name</th>
<th>Apply</th>
<th>Marker pairs</th>
<th>Scaled segments</th>
<th>Local scaling axes</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pelvis_xyz</td>
<td>True</td>
<td>ASISRght ASISLeft</td>
<td>Pelvis</td>
<td>XYZ</td>
</tr>
<tr>
<td>Torso_y</td>
<td>True</td>
<td>ShoulderLeft ASISLeft, ShoulderRight ASISRight</td>
<td>Torso, Scapula (both), Clavicle (both)</td>
<td>Y</td>
</tr>
<tr>
<td>Torso_xz</td>
<td>True</td>
<td>ShoulderLeft ShoulderRight</td>
<td>Torso, Scapula (both), Clavicle (both)</td>
<td>XZ</td>
</tr>
<tr>
<td>Thigh</td>
<td>True</td>
<td>ASISRght KneeLat</td>
<td>Femur (both)</td>
<td>XYZ</td>
</tr>
<tr>
<td>Shank</td>
<td>True</td>
<td>KneeLat Malleolus</td>
<td>Tibia (both), Talus (both)</td>
<td>XYZ</td>
</tr>
<tr>
<td>Foot</td>
<td>True</td>
<td>Malleolus Metatarsus</td>
<td>Calc (both), Toes (Both)</td>
<td>XYZ</td>
</tr>
</tbody>
</table>

Table 18. Marker weights used to scale the OpenSim Gait2392 model using KCMT data. Higher weights were given to markers placed on bony landmarks. Weights are relative, for example a weight of 1 compared to 10 is the same as 100 compared to 1000.

<table>
<thead>
<tr>
<th>Marker name</th>
<th>Weight</th>
<th>Apply</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShoulderLeft</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>ShoulderRight</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>SpineShoulder</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>SpineBase</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>ASISLeft</td>
<td>5000</td>
<td>True</td>
</tr>
<tr>
<td>ASISRight</td>
<td>5000</td>
<td>True</td>
</tr>
<tr>
<td>Thigh</td>
<td>2000</td>
<td>True</td>
</tr>
<tr>
<td>KneeLat</td>
<td>5000</td>
<td>True</td>
</tr>
<tr>
<td>Tibia</td>
<td>2000</td>
<td>True</td>
</tr>
<tr>
<td>Malleolus</td>
<td>5000</td>
<td>True</td>
</tr>
<tr>
<td>Metatarsus</td>
<td>5000</td>
<td>True</td>
</tr>
</tbody>
</table>
8.7.2 OpenSim inverse kinematics parameters\textsuperscript{37}

Table 19. Marker weights used for the IK analysis of Vicon data. Markers used in static trials only were identified by an * and were disabled in this analysis.

<table>
<thead>
<tr>
<th>Marker name</th>
<th>Weight</th>
<th>Apply</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Trunk</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>MAN</td>
<td>100</td>
<td>True</td>
</tr>
<tr>
<td>T2</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>T10</td>
<td>100</td>
<td>True</td>
</tr>
<tr>
<td><strong>Pelvis</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LASI/RASI</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>SACR</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td><strong>Thigh</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LTHAP/RTHAP</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>LTHLP/RTHLP</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>LTHAD/RTHAD</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>LTHLD/RTHLD</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>LPAT/RPAT</td>
<td>100</td>
<td>True</td>
</tr>
<tr>
<td>LLEPI/RLEPI</td>
<td>100</td>
<td>True</td>
</tr>
<tr>
<td>LMEPI/RMEPI</td>
<td>100</td>
<td>False</td>
</tr>
<tr>
<td><strong>Tibia</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LTIAP/RTIAP</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>LTIAD/RTIAD</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>LLMAL/RLMAL</td>
<td>100</td>
<td>True</td>
</tr>
<tr>
<td>LMMAL/RMMAL</td>
<td>100</td>
<td>False</td>
</tr>
<tr>
<td><strong>Foot</strong></td>
<td></td>
<td></td>
</tr>
<tr>
<td>LMFL/RMFL</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>LMFS/RMFS</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>LTOE2/RTOE2</td>
<td>100</td>
<td>True</td>
</tr>
<tr>
<td>LHEEL/RHEEL</td>
<td>100</td>
<td>True</td>
</tr>
<tr>
<td>LHEEL2/RHEEL2*</td>
<td>100</td>
<td>False</td>
</tr>
<tr>
<td>LTOE/RTOE*</td>
<td>100</td>
<td>False</td>
</tr>
<tr>
<td>LMT5/RMT5</td>
<td>100</td>
<td>True</td>
</tr>
<tr>
<td>LMT1/RMT1</td>
<td>100</td>
<td>True</td>
</tr>
</tbody>
</table>

Table 20. Marker weights used for the IK analysis of KCMT data

<table>
<thead>
<tr>
<th>Marker name</th>
<th>Weight</th>
<th>Apply</th>
</tr>
</thead>
<tbody>
<tr>
<td>ShoulderLeft</td>
<td>1000</td>
<td>False</td>
</tr>
<tr>
<td>SpineShoulder</td>
<td>2000</td>
<td>True</td>
</tr>
<tr>
<td>ShoulderRight</td>
<td>1000</td>
<td>False</td>
</tr>
<tr>
<td>SpineBase</td>
<td>1000</td>
<td>True</td>
</tr>
<tr>
<td>ASISLeft</td>
<td>5000</td>
<td>True</td>
</tr>
<tr>
<td>ASISRight</td>
<td>5000</td>
<td>True</td>
</tr>
<tr>
<td>Thigh</td>
<td>2000</td>
<td>True</td>
</tr>
</tbody>
</table>

\textsuperscript{37} The files containing the OpenSim inverse kinematics parameters were denoted as \texttt{Schache_CHESM_v092F_IK_setup_Timmi.xml} and \texttt{ikSetupSingleLegRightKinEdgeV1.6Gait2392Arms.xml}, and were used to analyse Vicon and KCMT data, respectively.
8.7.3 Number of markers and degrees of freedom

The limited number of markers available for KCMT posed the problem of selecting optimal body locations to attach the markers, to maximize the tracking accuracy of the lower limbs. The minimum number of markers required to track a multibody model depends on the number of degrees of freedom (DOFs) of the model. The DOFs can be calculated using Kutzbach criterion (also known as the mobility formula):

$$DOF = 6(N - 1 - j) + \sum_{i=1}^{j} f_i$$

where $N$ is the number of moving bodies plus the fixed frame, $j$ is the number of kinematic pairs connecting the bodies and $f_i$ is the number of DOFs of each kinematic pair. Taking into account only a single lower limb, and considering that the subtalar and MTP angles were locked in the KCMT model (see subsection OpenSim models in section 8.3.3), there were $N = 5$ bodies, namely fixed frame, pelvis, femur, tibia and foot. The kinematic pairs were $j = 3$, i.e. a ball joint between frame and femur, a revolute pair between femur and tibia, and another revolute pair between tibia and foot. Because the ball joint has 3 DOFs (3 rotations) and the revolute pair has 1 DOF (1 rotation), the following number of DOFs for the model was obtained:

$$DOFs = 6(5 - 1 - 3) + 3 + 2 \cdot 1 = 11$$

These 11 DOFs were: 1-6) three rotations and three translations of the pelvis in the global coordinate system; 7-9) three rotations of the femur relative to the pelvis, corresponding to hip flexion, adduction and rotation; 10) one rotation of the tibia relative to the femur, corresponding to the knee flexion; 11) one rotation of the foot relative to the tibia, corresponding to the ankle flexion.

To obtain accurate joint angles from marker trajectories, a certain number of markers is required for each body segment, below which the model would be unconstrained. A minimum of three markers is required to track an unconstrained rigid body in 3D space. When a child body is linked to a parent body via a spherical joint, a minimum of two markers is required for the child body. When the two bodies are linked by a revolute joint, one marker is sufficient for the child body because of the single DOF available. However, because marker trajectories are affected by measurement error, it is common practice to attach redundant markers on each body segment, to compensate for tracking inaccuracies and soft tissue artefacts. Since the subtalar and MTP angles were locked to 0° in the KCMT model, the foot had a single DOF (ankle flexion) and therefore one marker (Metatarsus) was sufficient to measure its motion. Similarly, the tibia had a single DOF (knee flexion), therefore one marker would have been sufficient. However, because this study was focused on the knee, two markers – namely Tibia and Malleolus – were used to improve scaling and motion tracking accuracy of the tibia. The femur was linked to the pelvis via a spherical joint, therefore two markers were used, i.e. Thigh and KneeLat. Because the two coloured markers attached to the pelvis (ASISRight and ASISLeft) were insufficient to track the six DOFs of this body segment, a virtual joint centre tracked by the Kinect v2 markerless pose estimation algorithm was also included in the KCMT markerset (SpineBase). To track the torso,
other three virtual joints were used, namely ShoulderLeft, SpineShoulder and ShoulderRight. The effects of using these virtual joints on accuracy are illustrated in section 8.5 Discussion of the present chapter.

8.8 References


CHAPTER 9

Accuracy of a novel marker tracking approach based on the low-cost Microsoft Kinect v2 sensor

Alessandro Timmi\textsuperscript{a,*}, Gino Coates\textsuperscript{a}, Karine Fortin\textsuperscript{b}, David Ackland\textsuperscript{c}, Adam L. Bryant\textsuperscript{b}, Ian Gordon\textsuperscript{d} and Peter Pivonka\textsuperscript{a,e}

\textsuperscript{a}St Vincent's Department of Surgery, The University of Melbourne, 29 Regent St, Fitzroy VIC 3065, Australia
\textsuperscript{b}Centre for Health, Exercise & Sports Medicine, The University of Melbourne, 202-206 Berkeley St, Carlton VIC 3053, Australia
\textsuperscript{c}Department of Mechanical Engineering, The University of Melbourne, Grattan St, Parkville VIC 3010, Australia
\textsuperscript{d}Statistical Consulting Centre, School of Mathematics and Statistics, The University of Melbourne, Parkville VIC 3010, Australia
\textsuperscript{e}School of Chemistry, Physics and Mechanical Engineering, Queensland University of Technology, Brisbane QLD 4000, Australia

The content of this chapter was published in the journal \textit{Medical Engineering & Physics} (Timmi et al. 2018).

9.1 Abstract

Microsoft Kinect for Windows v2 is a motion analysis system that features a markerless human pose estimation algorithm. Given its affordability and portability, Kinect v2 has potential for use in biomechanical research and within clinical settings; however, recent studies suggest high inaccuracy of the markerless algorithm compared to marker-based motion capture systems. A novel tracking method was developed using Kinect v2, employing custom-made coloured markers and computer vision techniques. The aim of this study was to test the accuracy of this approach relative to a conventional Vicon motion analysis system, performing a Bland-Altman analysis of agreement. Twenty participants were recruited, and markers placed on bony prominences near hip, knee and ankle. Three-dimensional coordinates of the markers were recorded during treadmill walking and running. The limits of agreement (LOA) of marker coordinates were narrower than −10 and 10 mm in most conditions, however a negative relationship between accuracy and treadmill speed was observed along Kinect depth direction. LOA of the surrogate knee angles were within −1.8°, 1.7° for flexion in all conditions and −2.9°, 1.7° for adduction during fast walking. The proposed methodology exhibited good agreement with a marker-based system over a range of gait speeds and, for this reason, may be useful as low-cost motion analysis tool for selected biomechanical applications.
9.2 Introduction

Marker-based optical motion capture systems and wearable inertial sensors have been employed widely in clinical gait analysis. In the most favourable conditions, optical marker-based tracking can provide an overall accuracy of 63±5 μm (Windolf et al. 2008). However, these systems are not always practical in the clinical setting, given the substantial setup costs and space requirements (Whittle 2007). A high-end 12-camera Vicon motion capture system costs approximately AU$ 250,000 (Thewlis et al. 2013). Wearable devices based on inertial sensors are low-cost, but are subject to position measurement errors as a consequence of acceleration integration (Muro-de-la-Herran et al. 2014). Markerless tracking has recently gained popularity in the biomechanics community, especially after the introduction of a low-cost (AU$ 200) and portable device denoted as Kinect (Microsoft, Redmond, US). The second iteration of this device, Kinect for Windows v2, has been employed in gait analysis (Mentiplay et al. 2015), balance and postural assessment (Clark et al. 2015) and rehabilitation training (Capecci et al. 2016); however, the accuracy of this system has been the primary factor limiting its widespread use.

Kinect v2, also known as Kinect for Xbox One, is an RGB-D camera based on time-of-flight (TOF) depth sensing technology (Li 2014) and featuring a markerless human pose estimation algorithm (Shotton et al. 2011). Several studies have assessed agreement between Kinect v2 markerless algorithm and marker-based optical motion capture systems. Average errors in joint centre coordinates can be as high as 207±71 mm during upright standing (Xu & McGorry 2015) and limits of agreement (LOA) equal to 28°, 46° are reported for peak knee flexion angle at a self-selected walking speed (Mentiplay et al. 2015). LOA of 7°, 25° were found for trunk anterior-posterior flexion, and of 5°, 17° for trunk lateral flexion (Clark et al. 2015). Average errors of 24°±10° and 26°±8° were observed respectively for the right and left peak knee flexion angles during squatting (Capecci et al. 2016). The tracking accuracy of Kinect may be influenced by the subject’s orientation and distance from Kinect itself, noise affecting depth data, the subject’s body shape, and limitations in the pose estimation algorithm (Wang et al. 2015). Motion speed might also influence accuracy, due to the low capture frame rate of Kinect v2 (30 fps) and local alterations in depth measurements caused by moving objects (Sarbolandi et al. 2015).

A strategy to enhance the accuracy of Kinect v2 is to employ anatomical markers, thus avoiding the error introduced by the markerless tracking approach. Several studies have explored the use of markers in combination with Kinect. However, these published approaches may require additional equipment and calibration procedures (Paolini et al. 2013), are capable of monitoring only specific tasks or variables (Macpherson et al. 2016), and have not had their accuracy rigorously evaluated (Su et al. 2016; Ye et al. 2015b; Ye et al. 2015a; Ye et al. 2016).

The primary aim of this study was to develop a novel marker tracking approach employing Kinect v2 and computer vision techniques. The secondary aim was to evaluate its accuracy. Agreement between this methodology and a conventional marker-based motion capture system was assessed. Markers attached on bony prominences near hip, knee and ankle were tracked in 3D by both systems during treadmill walking and running at different speeds. It was hypothesized that the use of coloured markers with the Kinect motion analysis system would provide more accurate kinematic measurements compared to markerless tracking, and that increases in marker speed would reduce tracking accuracy.
9.3 Methods

9.3.1 Participants

Twenty healthy participants (9 females, 31±6 years, 67±11 kg, 171±10 cm, BMI 22.9±2.1) were recruited and provided informed consent to this study. The sample size of 20 was calculated using a procedure recommended for agreement studies (Bland 2004), assuming a standard deviation $s = 5$ mm for Kinect v2 depth measurement error (Lachat, Macher, Mittet, et al. 2015; Lachat, Macher, Landes, et al. 2015; Corti et al. 2015). The study was approved by the Human Research Ethics Committee at The University of Melbourne.

9.3.2 Experimental protocol

Reflective and coloured markers mounted to custom-designed arrays (Figure 65) were attached to the hip (left iliac crest), knee (lateral epicondyle of the left femur) and ankle (lateral left malleolus, Figure 66). Reflective markers were tracked using a 12-camera Vicon motion capture system (Vicon, Oxford, UK) sampling at 120 Hz, while coloured markers were tracked by a novel Kinect v2-based system sampling at 30 Hz. The two systems were controlled by separate computers. Kinect v2 was positioned on the side of a treadmill (Landice L7, NJ, US) at 1.5 m distance from its midline and at 0.75 m height from the ground. The device tilt angle was adjusted to ensure that its field of view contained the subjects’ lower limbs, including a safety margin. A pre-heating time of 30 minutes was allowed for Kinect v2 before data collection, to maximize measurement stability (Lachat, Macher, Mittet, et al. 2015).

![Figure 65. Custom 3D-printed arrays, with two passive retro-reflective markers tracked by the Vicon motion capture system and a blue coloured marker tracked by Kinect v2. The centre of the coloured marker and the midpoint of the two reflective markers are coincident.](image)

Coloured markers were custom-made using polystyrene foam spheres. A diameter of 38 mm was selected for them, since pilot data demonstrated that smaller markers could not be reliably tracked during dynamic tasks. The spheres were painted using matte acrylic paints (Jo Sonja’s, Chroma Inc.), to avoid reflections which would alter colours and affect depth measurements. Magenta, green and blue paints were selected for the hip, knee and ankle markers respectively. Significantly different hues, as well as high saturation and brightness were used to avoid identification errors among markers and with the background. Symmetric three-marker arrays were 3D-printed using black filament (Figure 65). Each array held 2 reflective markers (⌀9 mm) on its extremities and a coloured marker in the middle. The centres of the three markers lied on a straight line.

Data collection was performed while subjects ambulated on the treadmill at four different speeds: slow walking (0.83 m/s), fast walking (1.31 m/s), slow running (2.00 m/s) and fast running (2.50 m/s). Three-dimensional marker coordinates were simultaneously acquired using Kinect and the Vicon system over a 15-s period.
9.3.3 Coloured marker tracking

A real-time algorithm was developed to measure the three-dimensional trajectories of coloured markers. Multiple data streams (i.e., colour, infrared and depth) obtained from Kinect v2 were processed using custom software based on OpenCV library (Bradski 2000) and Kinect software development kit (SDK, v.2.0.1410.19000). This software was installed on a computer with the following specifications: Intel Core i7-4770, AMD Radeon HD 8490, 16 GB RAM, Windows 8.1.

HSV ranges were pre-defined for each coloured marker and stored in an XML file, which was used for all trials. These ranges were centred on the hue, saturation and value of each marker, and their width was ±5° out of 180° for hue, ±50 out of 255 for saturation and ±75 out of 255 for value. The tracking algorithm was composed of the following steps: first, the coloured image was converted from RGB to HSV colour space and then binarized using the HSV ranges of the first marker in the XML file. Second, contours were detected in the binary image. A bounding box (BB) was applied to each contour. Among valid contours, the one with most white pixel was selected as candidate marker. Third, the BB of the candidate contour was mapped from the colour image to the infrared image and was denoted as infrared ROI (IRROI). The colour sensor is offset from the sensor producing depth and infrared images. To map between corresponding locations on the two sensors, the Coordinate Mapper class (Microsoft 2014a) provided by Kinect SDK was used. Specifically, the method converting colour pixels to depth pixels was used in this case. Fourth, the Hough Circle Transform (HCT) was applied to the IRROI to search for a circle, corresponding to the spherical marker. The following parameters were selected for the HCT: minimum radius = 3 pixels; maximum radius = minimum between width and height of the IRROI; accumulator threshold = 13; minimum distance = 100 pixels. Fifth, the centre of the roundest circle among those detected by the HCT was mapped from the infrared image to camera space, obtaining the 3D position vector of the centre of the marker spherical surface. This operation is called un-projecting and was performed via the Coordinate Mapper. Specifically, the method converting depth pixels to camera space points was used. Lastly, to obtain the coordinates of the inner centre of the marker, the position vector was extended by the known marker radius. The process described above was repeated for the remaining marker colours defined in the XML file, before passing to the next frame object. Markers were automatically labelled based on their colours. This system will be referred to as Kinect coloured marker tracking (KCMT) in this paper.

The contrast between black arrays (only required for validation) and skin caused undesired edges in the infrared image, affecting marker detection. To prevent this issue and to create a locally uniform background, black cloth tape was attached between skin and arrays (Figure 66). To limit the automatic exposure adjustments of Kinect v2 RGB camera, brightness conditions were controlled during testing. The exposure time of the Kinect RGB camera was monitored in real-time using the Kinect SDK. Two photographic lights (each featuring four 115-W compact fluorescent lamps) were pointed towards the capture volume and white sheets were laid on the ground. With this setup, an exposure time of 19 ms was achieved.
9.3.4 Data analysis

Vicon marker coordinates were transformed to Kinect coordinate system, where X was the walking direction (posteroanterior), Y was the vertical axis (superior) and Z was the depth axis (mediolateral). Gaps in coloured marker trajectories corresponded to 0.1% of the captured frames and were filled using spline interpolation. Vicon and KCMT data were synchronized using a custom iterative algorithm based on cross-covariance. Before synchronization, spline interpolation was used to upsample both datasets to 480 fps, to obtain a finer synchronization step. After synchronization, both datasets were downsampled to 30 fps, to match Kinect v2 native framerate. Downsampling was achieved by keeping every n\(^{th}\) sample starting with the first sample, where n = 480 / 30.

A Bland-Altman analysis of agreement (Bland & Altman 1986; Bland & Altman 1999; Bland & Altman 2007) was performed between the coordinates of the centre of each coloured marker (measured using KCMT) and the mid-point of the two adjacent reflective markers (measured using the Vicon system). All frames of the gait trials were used for this analysis. The agreement between KCMT and the Vicon system was also determined for two surrogate knee angles. These surrogate angles were delimited by two vectors connecting the three marker arrays: \(a\), from the hip to the knee, and \(b\), from the knee to the ankle. These vectors were projected onto the XY plane, to determine the surrogate flexion, and onto the YZ plane, to determine the surrogate adduction. The agreement for these angles was estimated using all frames of the gait trials, and also in correspondence with selected gait events, i.e. foot-strike (FS) and toe-off (TO). These two gait events were detected as positive and negative peaks, respectively, in the X coordinate of the ankle marker captured by the Vicon system. Further details about coordinate transformation, statistical analysis and knee joint angles calculations are reported in the Supplementary Material.
9.4 Results

The LOA between KCMT and the Vicon system for marker coordinates ranged from 2.2, 9.6 mm for the hip X coordinate during slow walking up to −20.3, 39.9 mm for the ankle Z coordinate during fast running (Table 21). In most cases, however, the LOA were narrower than −10, 10 mm and the bias was less than 6 mm (Figure 67). The hip marker exhibited almost constant LOA across all axes and for all treadmill speeds (Figure 67a, d and g). In contrast, the ankle marker displayed a negative relationship between agreement and treadmill speed. This relationship was less evident for the X and Y coordinates, with LOA up to −6.3, 9.2 and −15.2, 7.4 mm respectively during fast running, and stronger for Z (Figure 67c, f and i). The knee marker showed an intermediate trend compared to the other two markers, with LOA up to −11.7, 9.1 mm for the Z coordinate during fast running (Figure 67b, e and h). The confidence intervals (CI) of the LOA ranged between ±0.7 and ±2.1 mm, with the largest interval reported for the ankle Z coordinate during fast running (Table 21).

Figure 67. Bias and limits of agreement versus treadmill speed for each marker coordinate

The agreement of the surrogate knee flexion angle was estimated using all frames of the gait trials (Figure 68a) and in correspondence with gait events (Figure 68b and c). Negligible bias was found in all cases (−0.2° at most). LOA ranged from −0.7°, 0.7° for FS during slow walking up to −1.8°, 1.4° for TO during fast walking. A negative relationship between treadmill speed and
agreement was observed in the analysis including all frames, with LOA ranging from $-1.1^\circ$, $1.0^\circ$ during slow walking up to $-1.4^\circ$, $1.7^\circ$ during fast running (Figure 68a). CI ranged between $\pm 0.1^\circ$ and $\pm 0.2^\circ$ (Table 21).

The agreement of the surrogate knee adduction angle estimated using all frames of the gait trials displayed a negative relation with treadmill speed (Figure 68d): LOA ranged from $-1.6^\circ$, $1.3^\circ$ during slow walking up to $-9.3^\circ$, $6.8^\circ$ during fast running, while the bias varied from $-0.2^\circ$ to $-1.2^\circ$. The analysis in correspondence of FS (Figure 68e) exhibited almost constant LOA ($-1.0^\circ$, $1.4^\circ$ at worst) and bias ($0.3^\circ$ at most) for all treadmill speeds. A negative relationship between agreement and speed was also observed for TO, with LOA ranging from $-2.3^\circ$, $0.7^\circ$ for slow walking up to $-5.3^\circ$, $1.1^\circ$ for fast running, and bias from $-0.8^\circ$ up to $-2.1^\circ$. The CI of the LOA were within $\pm 0.3^\circ$ in all cases (Table 21).

![Figure 68. Bias and limits of agreement versus treadmill speed for the surrogate knee flexion and adduction angles across all frames of the gait trials (a and d plots), and in correspondence of specific gait events: foot-strike (FS, b and e plots) and toe-off (TO, c and f plots)](image-url)
Table 21. Results of the Bland-Altman analyses of agreement, reported as biases, limits of agreement (LOA) and margins of error (MOE, intended as half the width of the 95% confidence intervals of the limits of agreement).

<table>
<thead>
<tr>
<th>Coordinate</th>
<th>Bias</th>
<th>Lower LOA</th>
<th>Upper LOA</th>
<th>MOE</th>
<th>Bias</th>
<th>Lower LOA</th>
<th>Upper LOA</th>
<th>MOE</th>
</tr>
</thead>
<tbody>
<tr>
<td>Slow walk</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Fast walk</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hip X (mm)</td>
<td>5.9</td>
<td>2.2</td>
<td>9.6</td>
<td>1.0</td>
<td>6.0</td>
<td>2.2</td>
<td>9.9</td>
<td>1.0</td>
</tr>
<tr>
<td>Hip Y (mm)</td>
<td>−3.0</td>
<td>−7.8</td>
<td>1.9</td>
<td>1.5</td>
<td>−2.7</td>
<td>−7.7</td>
<td>2.3</td>
<td>1.4</td>
</tr>
<tr>
<td>Hip Z (mm)</td>
<td>−5.1</td>
<td>−9.3</td>
<td>−0.9</td>
<td>0.9</td>
<td>−5.1</td>
<td>−9.3</td>
<td>−1.0</td>
<td>0.9</td>
</tr>
<tr>
<td>Knee X (mm)</td>
<td>3.3</td>
<td>−1.0</td>
<td>7.6</td>
<td>0.8</td>
<td>3.2</td>
<td>−1.1</td>
<td>7.6</td>
<td>0.8</td>
</tr>
<tr>
<td>Knee Y (mm)</td>
<td>−1.4</td>
<td>−6.7</td>
<td>3.9</td>
<td>1.2</td>
<td>−1.3</td>
<td>−7.4</td>
<td>4.9</td>
<td>1.2</td>
</tr>
<tr>
<td>Knee Z (mm)</td>
<td>−2.0</td>
<td>−7.1</td>
<td>3.1</td>
<td>1.0</td>
<td>−1.7</td>
<td>−7.5</td>
<td>4.1</td>
<td>0.9</td>
</tr>
<tr>
<td>Ankle X (mm)</td>
<td>1.1</td>
<td>−3.3</td>
<td>5.5</td>
<td>0.8</td>
<td>0.7</td>
<td>−4.7</td>
<td>6.1</td>
<td>0.8</td>
</tr>
<tr>
<td>Ankle Y (mm)</td>
<td>−0.8</td>
<td>−6.7</td>
<td>5.2</td>
<td>1.2</td>
<td>−1.6</td>
<td>−9.1</td>
<td>6.0</td>
<td>1.2</td>
</tr>
<tr>
<td>Ankle Z (mm)</td>
<td>1.4</td>
<td>−5.5</td>
<td>8.3</td>
<td>1.2</td>
<td>4.9</td>
<td>−8.2</td>
<td>18.0</td>
<td>1.1</td>
</tr>
<tr>
<td>Knee flexion (°)</td>
<td>−0.1</td>
<td>−1.1</td>
<td>1.0</td>
<td>0.1</td>
<td>−0.1</td>
<td>−1.2</td>
<td>1.1</td>
<td>0.1</td>
</tr>
<tr>
<td>Knee flexion FS (°)</td>
<td>0.0</td>
<td>0.0</td>
<td>0.7</td>
<td>0.1</td>
<td>0.0</td>
<td>−0.7</td>
<td>0.8</td>
<td>0.1</td>
</tr>
<tr>
<td>Knee flexion TO (°)</td>
<td>0.0</td>
<td>0.0</td>
<td>1.6</td>
<td>0.2</td>
<td>0.0</td>
<td>−1.8</td>
<td>1.4</td>
<td>0.2</td>
</tr>
<tr>
<td>Knee adduction (°)</td>
<td>−0.2</td>
<td>−0.2</td>
<td>1.3</td>
<td>0.0</td>
<td>−0.6</td>
<td>−2.8</td>
<td>1.7</td>
<td>0.0</td>
</tr>
<tr>
<td>Knee adduction FS (°)</td>
<td>0.3</td>
<td>0.3</td>
<td>0.6</td>
<td>0.1</td>
<td>0.3</td>
<td>−0.7</td>
<td>1.2</td>
<td>0.1</td>
</tr>
<tr>
<td>Knee adduction TO (°)</td>
<td>−0.8</td>
<td>−1.0</td>
<td>0.7</td>
<td>0.2</td>
<td>−1.0</td>
<td>−2.9</td>
<td>0.9</td>
<td>0.2</td>
</tr>
<tr>
<td>Slow run</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td>Fast run</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Hip X (mm)</td>
<td>5.8</td>
<td>1.9</td>
<td>9.8</td>
<td>1.0</td>
<td>5.9</td>
<td>1.9</td>
<td>10.0</td>
<td>0.9</td>
</tr>
<tr>
<td>Hip Y (mm)</td>
<td>−3.2</td>
<td>−8.7</td>
<td>2.2</td>
<td>1.3</td>
<td>−3.3</td>
<td>−8.7</td>
<td>2.1</td>
<td>1.3</td>
</tr>
<tr>
<td>Hip Z (mm)</td>
<td>−4.7</td>
<td>−9.4</td>
<td>0.0</td>
<td>0.9</td>
<td>−4.6</td>
<td>−9.4</td>
<td>0.2</td>
<td>0.9</td>
</tr>
<tr>
<td>Knee X (mm)</td>
<td>3.1</td>
<td>−1.7</td>
<td>7.9</td>
<td>0.7</td>
<td>3.1</td>
<td>−2.0</td>
<td>8.2</td>
<td>0.7</td>
</tr>
<tr>
<td>Knee Y (mm)</td>
<td>−1.6</td>
<td>−7.6</td>
<td>4.5</td>
<td>1.1</td>
<td>−1.6</td>
<td>−7.5</td>
<td>4.3</td>
<td>1.0</td>
</tr>
<tr>
<td>Knee Z (mm)</td>
<td>−1.3</td>
<td>−9.2</td>
<td>6.5</td>
<td>0.8</td>
<td>−1.3</td>
<td>−11.7</td>
<td>9.1</td>
<td>0.9</td>
</tr>
<tr>
<td>Ankle X (mm)</td>
<td>1.1</td>
<td>−5.7</td>
<td>7.9</td>
<td>0.9</td>
<td>1.4</td>
<td>−6.3</td>
<td>9.2</td>
<td>0.9</td>
</tr>
<tr>
<td>Ankle Y (mm)</td>
<td>−2.5</td>
<td>−11.7</td>
<td>6.6</td>
<td>1.0</td>
<td>−3.9</td>
<td>−15.2</td>
<td>7.4</td>
<td>1.1</td>
</tr>
<tr>
<td>Ankle Z (mm)</td>
<td>6.4</td>
<td>−16.0</td>
<td>28.7</td>
<td>1.6</td>
<td>9.8</td>
<td>−20.3</td>
<td>39.9</td>
<td>2.1</td>
</tr>
<tr>
<td>Knee flexion (°)</td>
<td>0.0</td>
<td>−1.4</td>
<td>1.4</td>
<td>0.1</td>
<td>0.1</td>
<td>−1.4</td>
<td>1.7</td>
<td>0.1</td>
</tr>
<tr>
<td>Knee flexion FS (°)</td>
<td>0.0</td>
<td>0.0</td>
<td>0.8</td>
<td>0.1</td>
<td>0.0</td>
<td>−1.2</td>
<td>1.2</td>
<td>0.1</td>
</tr>
<tr>
<td>Knee flexion TO (°)</td>
<td>−0.1</td>
<td>0.0</td>
<td>1.3</td>
<td>0.2</td>
<td>0.0</td>
<td>−1.4</td>
<td>1.3</td>
<td>0.2</td>
</tr>
<tr>
<td>Knee adduction (°)</td>
<td>−0.7</td>
<td>−0.7</td>
<td>4.0</td>
<td>0.1</td>
<td>−1.2</td>
<td>−9.3</td>
<td>6.8</td>
<td>0.1</td>
</tr>
<tr>
<td>Knee adduction FS (°)</td>
<td>0.2</td>
<td>0.2</td>
<td>0.7</td>
<td>0.1</td>
<td>0.2</td>
<td>−1.0</td>
<td>1.4</td>
<td>0.1</td>
</tr>
<tr>
<td>Knee adduction TO (°)</td>
<td>−1.3</td>
<td>−2.1</td>
<td>1.1</td>
<td>0.2</td>
<td>−2.1</td>
<td>−5.3</td>
<td>1.1</td>
<td>0.3</td>
</tr>
</tbody>
</table>
9.5 Discussion

The aim of this study was twofold: first, to develop a novel low-cost methodology to track coloured markers in 3D using Kinect v2 and image analysis techniques; second, to assess its accuracy relative to a conventional marker-based motion capture system. The Bland-Altman analyses confirmed the hypotheses: the proposed tracking method, KCMT, is in good agreement with the Vicon motion capture system and, based on results from previous studies, is one order of magnitude more accurate than Kinect v2 markerless pose estimation algorithm. Direct comparison between KCMT and Kinect v2 markerless algorithm will be required to confirm these encouraging results. Moreover, the agreement between KCMT and the Vicon system demonstrated a negative relationship with gait speed.

Xu & McGorry (Xu & McGorry 2015) compared Kinect v2 markerless algorithm with a marker-based system and found the accuracy to be posture- and joint-dependent. For an upright standing posture, they found the average error across all participants and joint centres was 87±15 mm. Poorest agreement was at the feet, with an average deviation of 207±71 mm reported for the left foot. Different standing postures yielded even larger deviations. Given the degree of inaccuracy, the authors discouraged the use of Kinect v2 pose estimation algorithm for inverse dynamics. Wang et al. (Wang et al. 2015) oriented Kinect v2 in different configurations relative to the participants during both sitting and standing tasks. Considering only standing exercises with Kinect facing participants (best-case scenario), the mean error of the markerless algorithm for lower body joints ranged between 103 and 146 mm (SD = 30 mm), with ankle and foot kinematics being most inaccurate. In the present study, KCMT was tested during demanding conditions including running, with the Bland-Altman analyses resulting in bias lower than 6 mm and LOA for marker trajectories narrower than −10, 10 mm in most cases. Consequently, indirect comparison with results from literature indicates that improved accuracy may be achieved by using the KCMT method proposed in the present study. Future studies ought to verify the possibility of tracking an increased number of coloured markers attached to pelvis and lower limbs. In this manner, a local coordinate system may be defined for each body segment. Subsequently, hip, knee, and ankle joint angles may be calculated according to ISB recommendations (Grood & Suntay 1983; Joob 2002). We note that, if the reported degree of accuracy is confirmed under such a more sophisticated protocol, the KCMT may ultimately be used to drive inverse dynamics analyses.

The agreement between KCMT and the Vicon system for 3D coordinates exhibited a negative relationship with marker speed, especially in the depth (Z-axis) direction (Figure 67). Indeed, the lowest agreement was observed for the Z coordinate of the ankle marker (the most distal) during fast running. This finding is consistent with Sarbolandi et al. (Sarbolandi et al. 2015), who reported that motion locally alters depth measurements in TOF sensors. Because the knee adduction angle was calculated using the Z coordinate of markers, its accuracy was more affected by speed compared to flexion. X and Y coordinates are derived from depth measurements; therefore, they are also affected by motion artefacts, albeit to a lesser extent than Z. The agreement of knee angles calculated using all frames of the gait trials showed a negative relationship with treadmill speed (Figs. 4a and d). In contrast, treadmill speed had little effect on the agreement in correspondence of FS or TO (Figs. 4b, c, e and f), because leg speed was at a minimum in those instants. Since the results suggest higher accuracy at the hip and lower accuracy at the ankle, care should be taken in interpreting marker trajectories and joint angles related to distal limb segments, particularly if measured along Kinect depth axis and during high-speed tasks.
Mentiplay et al. (Mentiplay et al. 2015) performed a Bland-Altman analysis between Kinect v2 pose estimation and a Vicon system for hip, knee and ankle kinematics during comfortable pace gait. LOA in this study were 28°, 46° for peak knee flexion during the swing phase, −21°, 1° for peak knee flexion at ground contact and −13°, 10° for peak knee adduction during contact. It should be noted that these results were obtained after removing outliers, hence, the actual LOA may be significantly larger. In contrast, the LOA between KCMT and the Vicon system were within −1.8°, 1.7° for the knee flexion angle in all cases and −2.9°, 1.7° for the knee adduction angle during fast walking.

Using anatomical markers with Kinect may appear cumbersome; however, this approach may be required to obtain accuracy in-line with that of conventional marker-based motion capture systems. Other studies have explored the use of markers with Kinect. Paolini and co-workers (Paolini et al. 2013) attached a red patch to a shoe and positioned a Kinect v1 in front of a treadmill to track foot motion during slow walking at 0.83 m/s. Their system also required the calibration of an external RGB camera glued on top of Kinect. The authors reported an error of 26.5±9.1 mm (RMSE±SD) in the depth direction. At the same speed, the depth coordinate of the ankle marker measured by the KCMT was considerably more accurate, featuring an error of 1.4±6.9 mm (Table 21, reported as bias±2SD).

Su et al. (2016), tracked 36 yellow markers using Kinect v1 for postural assessment, but the accuracy of this methodology was evaluated only during static tests. The mean errors were 1.1° for angles identified by 3 markers, and 6 mm for distances between 2 markers. However, these variables were measured on the Kinect XY plane, i.e. excluding the depth coordinate (Z), which is the least accurate. Furthermore, because the accuracy of this methodology was evaluated only during static tests, results for the KCMT at slow walking speed were used for comparison. The bias of the surrogate knee flexion angle was −0.1°, whereas the bias on the XY plane was 5.9 mm at worst (Table 21). These results indicate that the KCMT is more accurate even under more challenging conditions, i.e. during dynamic instead of static tracking.

Macpherson et al. (2016) used Kinect v1 to track 4 reflective markers attached to participant’s backs in order to assess pelvic and trunk kinematics during treadmill locomotion. The LOA reported by Macpherson for the pelvis landmark are comparable to those reported for the hip marker tracked by the KCMT. However, no fast-moving distal markers were analysed in that study. Furthermore, their system was not designed to track the lower limbs and, according to the authors of that study, cannot be used for more complex tasks.

Another method based on reflective markers and Kinect v2 for gait analysis was proposed by Ye et al. (Ye et al. 2015b; Ye et al. 2015a; Ye et al. 2016). However, only limited results were reported about the agreement between this solution and a Vicon motion capture system. Specifically, only 4 sample plots of the knee angle were included (out of 40 trials collected), from which the agreement appears variable across the gait cycle. Root-mean-square error of the knee angle was calculated on a per-experiment basis, returning a maximum of 6°. In comparison, the LOA of the KCMT were within −1.8 to 1.7° for the surrogate knee flexion angle in all conditions (Table 21). Moreover, this solution was task-specific, because several parameters would need to be changed to get maximum accuracy under different capture conditions.

In contrast with the aforementioned studies, KCMT supports a customizable markerset and uses Kinect v2 built-in RGB camera. Markers are automatically labelled based on colours, without the need for task-specific models or user intervention, making the proposed solution more flexible
in terms of applications and easier to use. Furthermore, the linear and angular accuracy of the system was evaluated at different motion speeds.

While results obtained with KCMT are encouraging, there are several limitations that ought to be considered. Higher distance between subject and Kinect corresponds to lower tracking accuracy, due to limitations of TOF technology. As a single-camera system, the capture volume of Kinect is smaller compared to that of a multi-camera system. Moreover, markers must be in sight of Kinect, to minimize the risk of marker occlusions. The use of colours currently limits the number of markers that may be tracked at one time to a maximum of 7, because significantly different hues must be used to avoid labelling errors. This, in turn, limits the number of joint angles that can be simultaneously measured. However, this limitation may be overcome in the future, using a single colour for all markers and implementing an automatic labelling algorithm, based on relative positions of markers on the body. Brightness of the scene must be sufficient to ensure low exposure time of Kinect RGB camera, otherwise excessive motion blur may occur and affect tracking accuracy. A bright background is fundamental and photographic lights may also be required. This may limit the possible environments in which KCMT may be employed.

In conclusion, a novel marker-based tracking method for Kinect v2 was presented. The approach demonstrated good agreement with an optical marker-based motion capture system over a range of gait speeds, suggesting that the proposed strategy may represent a practical and low-cost alternative motion analysis system. The proposed method may be useful in gait analysis or motion assessment in settings that are not amenable to large-scale multi-camera optical motion analysis systems.

9.6 Funding
This study was part-supported by a Defence Health Foundation grant (2015). AT is the recipient of a Melbourne International Research Scholarship provided by The University of Melbourne. ALB is the recipient of a NHMRC Career Development Fellowship (#1053521).

9.7 Ethical Approval
The study was approved by the Human Research Ethics Committee at The University of Melbourne (#1646087).

9.8 References


9.9 Supplementary Material

9.9.1 Coordinate transformation

Vicon marker coordinates were transformed to Kinect v2 space using Eq. (9.1):

\[ r_k = R_k^T (r_v - d_{vk}) \] (9.1)

where \( r_v \) is the position of a given point expressed in Vicon space, \( r_k \) is the position of the point transformed from Vicon to Kinect space, \( R_k \) is the rotation matrix of Kinect reference frame in Vicon space and \( d_{vk} \) is the translation vector from Vicon to Kinect origin, expressed in Vicon space.

To determine \( R_k \), 4 reflective markers were attached on the top surface of Kinect (Figure 69), forming an L-shape parallel to its X and Z axes. The Y axis was determined using the right-hand rule. To obtain the translation vector \( d_{vk} \), a marker array was placed in the centre of the capture volume and tracked using both systems. Using \( r_k, r_v \) and \( R_k, d_{vk} \) was calculated by inverting Eq. 9.1.

Figure 69. Four reflective markers were attached to the top surface of Kinect v2, to determine its orientation in Vicon space

A coordinate transformation based on the 3D positions of manually-placed markers is affected by human error. A number of strategies were implemented to minimize this error. First, the edges of the device were used as reference to position the markers. Second, although 3 points are sufficient to determine the 3D pose of an object, 4 markers were used in this study to minimize the pose estimation error. Third, the distance between these markers was maximized as much as possible, considering the limited area provided by the Kinect top surface. Fourth, relatively small reflective markers were used (Ø9 mm).

9.9.2 Statistical analysis

The agreement between the proposed Kinect coloured marker tracking (KCMT) methodology and the Vicon motion analysis system was assessed using the Bland-Altman analysis of agreement (Bland & Altman 1986; Bland & Altman 1999; Bland & Altman 2007). This statistical technique was preferred to others – namely intraclass correlation coefficient, Pearson correlation, concordance correlation coefficient or root-mean-square error – for several reasons: i) it doesn’t depend on the variability of the sample; ii) it enables to separate systematic and random error; iii) it provides results using the same scale of the measurements; iv) the magnitude of the acceptable error is not a statistical decision, but a biomechanical one (Bland & Altman 1990).

The differences between the two measurement systems were modelled using Eq. (9.2):
\[ D_{i,t} = \mu_D + S_i + E_{i,t} \]  

(9.2)

where \( D_{i,t} \) is the \( t^{th} \) difference KCMT – Vicon for the \( i^{th} \) individual, \( \mu_D \) is the overall mean, \( S_i \) is a random between-subject effect for subject \( i \), and \( E_{i,t} \) is a random within-subject error term.

It was assumed that \( S_i \) and \( E_{i,t} \) were independent from each other, had zero mean and variance equal to \( \sigma_S^2 \) and \( \sigma_E^2 \) respectively. To allow for autocorrelation, an auto-regression order 1 model was fitted: this model assumes that the correlation between \( E_{i,t} \) and \( E_{i,t-1} \) is equal to \( \rho \), which was estimated.

Since \( D_{i,t} \) has mean \( \mu_D \) and variance \( \sigma_S^2 + \sigma_E^2 \), the LOA were calculated using Eq. (9.3):

\[ \text{LOA} = \hat{\mu}_D \pm 1.96 \sqrt{\hat{\sigma}_S^2 + \hat{\sigma}_E^2} \]  

(9.3)

where the \( ^\wedge \) notation denotes sample estimates of population parameters. Approximate 95% confidence intervals (CI) were determined as well.

The differences between the two motion capture systems were calculated using all frames from the gait trials, i.e. all measurements recorded at all time points. Furthermore, for the surrogate knee angles, the Bland-Altman analysis was repeated in correspondence of specific gait events, i.e. FS and TO. In this case, only measurements recorded at those time points were used in the calculation of the differences. In both analysis types, the same Eq. 9.3 was used to determine the LOA; the only difference was represented by the input data, i.e. all frames in the first case, as opposed to selected frames in the second.

### 9.9.3 Calculation of surrogate knee joint angles

To calculate the knee joint angles, two local coordinate systems must be fully determined, one attached to the femur and one to the tibia. Therefore, the coordinates of at least 3 points attached to each segment must be known. Since this study only employed two coloured markers for each segment (the knee marker being shared between femur and tibia), the 3D orientation of each local coordinate systems could not be fully determined, i.e. only the orientation of the longitudinal (shaft) axis of each body segment was known. The relative transverse-plane rotation between femur and tibia was therefore not determined.

Two vectors were defined connecting the three marker arrays: \( \mathbf{a} \), from the hip to the knee, and \( \mathbf{b} \), from the knee to the ankle. Because participants ambulated along the X-axis, Kinect XY plane was approximately parallel to the participants’ sagittal plane. Therefore, the difference between the absolute orientations of the projections of \( \mathbf{b} \) and \( \mathbf{a} \) onto the plane XY was denoted as surrogate knee flexion angle (\( \theta \), Figure 70), whereas the difference of the projections onto the plane YZ was denoted as surrogate knee adduction angle (\( \phi \), Figure 71).

As a consequence of this approach, the surrogate flexion and adduction angles represented an approximation of the corresponding actual angles, because they were calculated as projections of the knee angle on Kinect XY (Figure 70) and YZ planes (Figure 71) respectively, rather than on the sagittal and frontal plane of the femur coordinate system, respectively. The consequence of this approach was that the surrogate adduction angle was influenced by an off-plane flexion component when the participant’s leg was not perfectly lying on a plane parallel to XY (and vice versa for the flexion angle). However, this approximation didn’t affect the agreement results from a quantitative standpoint, because the surrogate angles were calculated in the same way for both Kinect and Vicon.
The same approximation doesn’t hold for the knee rotation angle. For example, when the knee is extended, the hip, knee and ankle markers are approximately aligned. In this case, the projections of the vectors $\mathbf{a}$ (connecting hip to knee) and $\mathbf{b}$ (connecting knee to ankle) on the XZ plane are relatively short compared to the average error of KCMT (which is in the order of magnitude of 1 cm). Therefore, in this case, the calculation of the approximated knee rotation angle is affected by extremely low signal-to-noise ratio. On the other hand, when the knee is bent, the XZ projection of the angle between $\mathbf{a}$ and $\mathbf{b}$ is essentially a measurement of the hip transverse-plane rotation, rather than knee rotation. Therefore, in this case, this angle is not a measurement of the actual intra-rotation between femur and tibia.

Figure 70. Projection of the knee angle onto the XY plane of Kinect v2 coordinate system, denoted as surrogate knee flexion angle ($\theta$). The hip, knee and ankle markers are indicated as H, K and A, respectively.

Figure 71. Projection of the knee angle onto the YZ plane of Kinect v2 coordinate system, denoted as surrogate knee adduction angle ($\phi$). The hip, knee and ankle markers are indicated as H, K and A, respectively.
9.9.4 References


CHAPTER 10
Ongoing and future work

10.1 Future developments: increasing the number of markers

The attempt to increase the number of markers by finding different colours (besides the 7 currently in use) was unsuccessful, because the Kinect RGB camera confused some hues with each other (e.g. violet with blue) or with the participant’s skin (e.g. orange, see Chapter 4). A possible alternative to increase the number of markers would be to use a single colour for all of them, thus overcoming the current limit of 7 different hues. However, with this approach, the KCMT system could no longer distinguish between each marker: the system would only be able to return all markers found in each frame, without knowing how marker positions are related across consecutive frames, i.e. without being able to reconstruct their trajectories (Figure 72A). Therefore, other criteria instead of colour would be required to reconstruct marker trajectories in 3D (Figure 72B). Furthermore, each trajectory should be labelled, i.e. each marker should be identified according to a pre-determined markerset template. While this operation may be done manually for a few trials, an algorithm would be required to automate this time-consuming process (Figure 72C).

To minimise labelling errors (i.e. “swaps”) when identifying trajectories of markers that overlap with respect to the camera, a hybrid approach using seven colours as well as multiple markers per colour would also be possible. With this method, different colours could be assigned to different anatomical parts, thus reducing the complexity of the labelling problem and therefore increasing the chances of the labelling algorithm finding the correct solution. However, for simplicity, only the single colour approach will be discussed in the rest of the present chapter.

![Figure 72. Without colours to identify each marker, other criteria must be used to reconstruct and label the 3D trajectories of each marker. (A) Marker coordinates before reconstruction. (B) Marker trajectories after reconstruction. (C) Marker trajectories after labelling.](image)

10.1.1 Trajectoryization algorithm

To implement a trajectoryization algorithm, the current KCMT algorithm should be modified to detect multiple markers of the same colour in each frame, instead of a single marker per colour. This would involve binarizing each RGB image a single time, instead of one time for each colour, with consequent reduction in processing time. Once the 3D coordinates of all markers have been obtained for all available frames, their trajectories could be reconstructed using the **closest point** algorithm. Because markers can no longer be distinguished by their colours, a way to follow
them along the duration of a trial would be to compare their position at frame $t$ with their position at frame $t+1$, identifying each marker at one frame with its closest neighbour at the following frame. Assuming the 3D position of a marker $M_i$ at time $t-1$ is represented as $M_i^{t-1}$, then the position of $M_i$ at time $t$ ($M_i^t$) is the one that is closest to $M_i^{t-1}$ among those available.

**10.1.2 Labelling algorithm**

A labelling algorithm should be able to univocally identify the reconstructed marker trajectories, giving each of them the correct marker name (i.e. *label*). This can be achieved using a static trial of the fully marked-up subject as reference, where all markers have been already manually labelled by the user. Analysing this data using a combination of rigid body kinematics, statistics and graph theory, it is possible to estimate which trajectory in a dynamic trial corresponds to which marker in the static trial.

A prototype of this algorithm (Figure 73) was developed and tested using trials recorded during the Single-leg squat study presented in Chapter 8 of this thesis. In such trials, 7 coloured markers\(^{38}\) were attached to the participants’ skin: ASISLeft, ASISRight, Thigh, KneeLat, Tibia, Malleolus, Metatarsus. To test the proposed labelling algorithm, marker labels were removed from the dynamic trials, converting them to unlabelled trials containing only the unnamed trajectories of the 7 markers. Labels were preserved in a static trial, recorded with the subject standing still.

Using labels from the static trial, markers attached to the same body segment (i.e. bone) were paired. The distance between markers in each pair was considered constant, therefore these pairs were denoted as *rigid segments* (Table 22). The mean length of each segment was calculated from the static trial. To account for the skin artefact and the measurement error affecting the segment lengths, a maximum threshold of 12 mm was allowed for the standard deviation of the segment lengths; this value was selected empirically and was stored as a constant for later use ($\text{SD}_{\text{threshold}}$).

\(^{38}\) For the purpose of testing the proposed labelling algorithm, the 4 virtual joint centres used in the single-leg squat study as part of the KCMT markerset were ignored.
Figure 73. Flowchart of the proposed labelling algorithm
Table 22. Pairs of markers attached to the same bone were defined as rigid segments, because their distance was constant

<table>
<thead>
<tr>
<th>Rigid segment</th>
<th>Marker 1</th>
<th>Marker 2</th>
</tr>
</thead>
<tbody>
<tr>
<td>#1</td>
<td>ASISLeft</td>
<td>ASISRight</td>
</tr>
<tr>
<td>#2</td>
<td>Thigh</td>
<td>KneeLat</td>
</tr>
<tr>
<td>#3</td>
<td>Tibia</td>
<td>KneeLat</td>
</tr>
<tr>
<td>#4</td>
<td>KneeLat</td>
<td>Malleolus</td>
</tr>
<tr>
<td>#5</td>
<td>Tibia</td>
<td>Malleolus</td>
</tr>
<tr>
<td>#6</td>
<td>Malleolus</td>
<td>Metatarsus</td>
</tr>
</tbody>
</table>

Using the coordinates of unlabelled markers stored in a dynamic trial, mean and standard deviation of the distances between all possible pairs of markers were calculated. Among all these marker pairs, rigid segments were detected as those having a standard deviation of the length less than the threshold: \( SD < SD_{\text{threshold}} \). Once the expected number of rigid segments was detected from the dynamic trial, each had to be matched with its counterpart defined from the static trial. Because the segments were rigid, their lengths measured in the dynamic trial had to be the same measured in the static trial. Therefore, to match a static length with the corresponding dynamic one, all dynamic lengths were subtracted to the static one, one at a time. The dynamic length returning the least absolute value of the difference was selected as match for that static length.

Once all rigid segments from the dynamic trial were matched with those from the static trial, the system could not label each marker individually yet, because each segment was composed of 2 markers. This ambiguity was solved using an instrument from graph theory, the connectivity of each marker. Connectivity was defined as the number of rigid segments departing from each marker, imagined as a node in a graph (Figure 74). The algorithm calculated the connectivity of an unlabelled marker as the number of its occurrences within all the rigid segments detected in the dynamic trial. It should be noted that the pelvic markers were disconnected from the femur markers, because no rigid segment can be defined connecting the ASIS to the femur.

Unlabelled markers were sorted from the most connected to the least. Starting with the most connected marker, all segments containing the current unlabelled marker were found, and the marker labels contained in these segments were stored in a list. The marker name with most occurrences (i.e. the mode) in the list was selected as label for the current marker, and this marker was added to the list of labelled markers. This process was repeated in an iterative way, each time removing labelled marker names from the list, until all markers were labelled.

The proposed algorithm was able to successfully identify the 7 coloured markers from single-leg squat trials, using a labelled static trial and an array defining the rigid segments as input. Further tests will be required to assess the robustness of this algorithm with different types of trials and with an increased number of markers attached to the participant’s skin.
Figure 74. A graph representing the 7 coloured markers used in the Single-leg squat study. Each node represents a marker and each line represents a rigid connection between two markers, indicating that they are attached to the same body segment (i.e. bone). It should be noted that ASIS markers are disconnected from the rest of the graph, because no rigid segment can be defined connecting the ASIS to the femur.
CHAPTER 11

Conclusion

The initial hypothesis of this PhD project was that single-leg squat data recorded using the Kinect v2 pose estimation algorithm were sufficiently accurate to be used for discerning subjects at risk of ACL injury from controls. The results of the preliminary study reported in Chapter 2 of this thesis, although not statistically significant because based on a single subject, suggested differently. Considering the knee flexion angle, the limits of agreement between Kinect and Vicon during double-leg squat were in the order of 20°, while the average difference between subjects at risk of ACL injury and controls during single-leg squat is 7.7° (Yamazaki et al. 2009). The poor accuracy of Kinect v2 pose estimation was confirmed by subsequent studies, reporting limits of agreement for flexion angles of the lower limbs in the order of 20° (Mentiplay et al. 2015; McGroarty et al. 2016) and linear errors for virtual joint centre positions above 10 cm (Wang et al. 2015; Xu & McGorry 2015).

At that stage, it was unclear if the poor agreement between Kinect v2 and Vicon was due to inaccuracy of the underlying depth data used as input by the pose estimation algorithm, or to intrinsic limitations of the pose estimation algorithm itself. For this reason, a literature review was carried out (Chapter 3) to identify and quantify the different sources of error affecting Kinect v2 depth data. It was found that, although there were many factors affecting the accuracy and precision of Kinect v2 depth measurements, some of them could be minimized by modifying the data capture protocol, e.g. observing a 20-minute warm-up phase before recording, reducing the distance Kinect-target, avoiding black and reflective surfaces, avoiding the corners of the depth sensor, and avoiding fast movements. The remaining sources of error were unavoidable and, if estimated at a distance of about 2 m between Kinect v2 and subject, they accounted for approximately ±10 mm of systematic error and ±5 mm of random error. In contrast, the Kinect v2 pose estimation algorithm was affected by errors in the order of 100 mm, with peaks up to 200 mm (Xu & McGorry 2015; Wang et al. 2015). Based on these numbers, it was concluded that most of the measurement error affecting Kinect v2 markerless tracking was introduced by the pose estimation algorithm, rather than being intrinsic to the raw depth data.

This finding motivated the development of an alternative motion tracking algorithm for Kinect v2, based on custom-made coloured markers and computer vision techniques (Chapter 4). This algorithm, developed in collaboration with Mr Gino Coates, combined the colour, infrared and depth data streams provided by Kinect v2 to identify and track coloured markers attached to the participants’ skin. This novel methodology was denoted as Kinect coloured marker tracking (KCMT) system.

To compare the KCMT against an established motion capture system (Vicon), different tools were developed, including a methodology to transform 3D marker coordinates from Vicon to Kinect frame of reference and vice-versa (Chapter 5), and a synchronization algorithm based on cross-covariance (Chapter 6). Furthermore, the Bland-Altman analysis was selected as the best approach to assess the agreement between the novel motion tracking system and the gold standard one (Chapter 7).

For the first agreement study (Chapter 8), single-leg squat (SLS) was selected as benchmark task for two reasons: first, because this task can be used for clinical assessment of ACL injury risk in young individuals (Zeller et al. 2003; Willson et al. 2006; Yamazaki et al. 2009); second, because
the limited framerate of Kinect v2 (30 fps) and the effect of motion speed on its depth measurements demand that slower tasks like SLS are selected instead of faster tasks like landing. Since the KCMT system could capture a maximum of 7 coloured markers at the same time, 4 virtual joint centres tracked by the Kinect v2 pose estimation algorithm were included in the markerset. This hybrid approach, combined with some limitations in the capture protocol of this study, affected the accuracy of the Kinect-derived flexion angles. Although these angles were more accurate than those obtained from the Kinect v2 pose estimation algorithm, their limits of agreement with Vicon were above 10°, indicating that the KCMT system was still not capable to discern between subjects at risk of ACL injury and controls.

To overcome the limitations identified in the previous study, a second agreement study was carried out between KCMT and Vicon, this time using an improved data collection protocol and treadmill locomotion as benchmark task (Chapter 9). The distance between participant and Kinect was reduced, ensuring that the field of view contained the subjects’ lower limbs, including a safety margin. To avoid the use of inaccurate virtual joint centres from the pose estimation algorithm, only 3 body landmarks were captured, i.e. ASIS, lateral epicondyle of the femur and lateral malleolus. Distances between corresponding Vicon and Kinect markers were eliminated by using symmetric 3D-printed arrays, which featured 2 reflective markers at the extremities and a coloured marker in correspondence of their midpoint. This improved data collection protocol resulted in substantially more accurate results for the KCMT system compared to the previous study. LOA between KCMT and Vicon for the linear coordinates of the 3 landmarks were within −10, 10 mm in most conditions, except for the depth coordinate (Z) of the ankle during running. The wider LOA in this case were attributed to the effect of speed on depth measurements: indeed, because the malleolus was the most distal among the 3 captured landmarks, it was also the one moving at highest speed. For the knee joint angles, the agreement was within −1.4°, 1.7° for knee flexion at all treadmill speeds, and within −2.8°, 1.7° for knee adduction up to fast walking speed. In comparison, average differences observed during SLS between subjects at risk of ACL injury and controls were 7.7° for knee flexion and 5.2° for knee valgus (Yamazaki et al. 2009). Although the agreement results for the KCMT were obtained using a different task (i.e. treadmill locomotion instead of SLS), they suggest that this novel solution may be sufficiently accurate to identify subjects at risk of ACL injury.

Table 23. The results of the Treadmill Study presented in Chapter 9 of this thesis indicate that the KCMT system is more accurate than Kinect v2 pose estimation algorithm, and that it may be sufficiently accurate to identify subjects at risk of ACL injury, based on comparison with data from Yamazaki et al. (2009).

<table>
<thead>
<tr>
<th>Treadmill Study (LOA)</th>
<th>Mentiply et al. (2015) (LOA)</th>
<th>Xu &amp; McGorry (2015); Wang et al. (2015), (mean)</th>
<th>Yamazaki et al. (2009) (mean)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Flexion (°)</td>
<td>−1.4, 1.7</td>
<td>−21, 1</td>
<td>7.7</td>
</tr>
<tr>
<td>Adduction (°)</td>
<td>−2.8, 1.7 (fast walking)</td>
<td>−13, 10</td>
<td>5.2</td>
</tr>
<tr>
<td>Linear accuracy (mm)</td>
<td>−10, 10</td>
<td>−</td>
<td>~100</td>
</tr>
</tbody>
</table>
However, for this hypothesis to be verified, another validation study should be carried out, this time using SLS as benchmark task and no virtual joint centres from the Kinect v2 pose estimation algorithm as part of the KCMT markerset. Because the Kinect v2 RGB camera cannot identify more than 7 coloured markers due to its incapability to discern certain hues, a different approach must be taken to increase the number of markers. One possibility is to use all markers of the same colour and implement an algorithm to reconstruct and label their trajectories, based on criteria other than colour. A possible approach to this problem was illustrated in Chapter 10.

11.1 Relevance of this thesis

In conclusion, a novel marker-based motion capture system denoted as KCMT was developed, based on the affordable and portable Microsoft Kinect v2 device. While a 12-camera Vicon motion capture system similar to that used as gold standard in this thesis costs approximately AU$ 250,000 (Thewlis et al. 2013), the proposed KCMT system was built using an AU$ 200 Kinect v2 sensor and freely available software libraries. It should be noted that, as of October 2017, Microsoft has discontinued the production of Kinect v2. Other depth sensors with similar features are currently available on the market, such as Intel RealSense, Orbbec Astra, and Stereolab ZED.

The validation study reported in Chapter 9 of this thesis indicated good agreement between KCMT and a multi-camera optical motion capture system over a range of gait speeds. Furthermore, the effect of speed on the accuracy of Kinect v2 depth measurement was quantified for the first time. Comparison with a published study on gait indicates that KCMT is one order of magnitude more accurate than the off-the-shelf Kinect v2 pose estimation algorithm at walking speed. Furthermore, the demonstrated level of agreement with a gold standard motion capture system suggests that the proposed solution may be sufficiently accurate to identify subjects at risk of ACL injury based on their lower limb kinematics.

The proposed solution has some limitations that should be considered. First, the accuracy of the system depends on the distance between Kinect v2 and the subject. Indeed, the depth measurement error increases with the square of the distance. Second, being a single-camera system, Kinect v2 delimits a smaller capture volume compared to conventional multi-camera motion capture systems. Third, only 7 coloured markers can be currently captured at the same time, due to a limitation of Kinect v2 RGB camera. A possible solution to this problem, based on the use of a single colour markerset, was illustrated in Chapter 10 of this thesis. Fourth, a sufficient level of brightness must be maintained in the room, to ensure the exposure time of Kinect v2 RGB camera stays below ~20 ms. This is due to another limitation of the RGB camera, whose exposure cannot be adjusted manually, but is automatically adapted by the firmware depending on the average brightness of the scene. Fifth, speed influences the accuracy of depth measurements, which in turn affects the accuracy of the tracked markers. The effect of speed is due to a limitation of the time-of-flight depth sensing technology and has been quantified in Chapter 9 of this thesis. A way to minimize this issue is to select slow speed tasks, such as treadmill walk and squat. Lastly, the encouraging agreement results found in the Treadmill Study (Chapter 9) should be confirmed by a new study, using single-leg squat as benchmark task and an increased number of coloured markers. If the agreement results with Vicon are confirmed, the proposed methodology may be used as a valid tool to screen large cohorts of young individuals for ACL injury risk, by assessing their lower limb biomechanics during single-leg squats.
The results of this thesis have several potential benefits. First, thanks to its affordability, the proposed technology may be used for on-site screening of large cohorts of young individuals in schools, gyms and clinical practices, to timely identify those at risk of ACL injury. Early screening, coupled with appropriate training protocols, may result in a reduction of injury events, which in turn may reduce the economic burden on public health connected to ACL surgeries (Myklebust, Engebretsen, et al. 2003). Young individuals may also benefit from early screening, as the avoidance of such serious injury would protect their regular growth, improve their sports participation rate, and reduce their long-term risk of developing osteoarthritis secondary to an ACL injury (Griffin et al. 2006; Lohmander et al. 2007; Myklebust, Holm, et al. 2003; Caine & Golightly 2011; von Porat et al. 2004).

11.2 References


Wang, Q. et al., 2015. Evaluation of pose tracking accuracy in the first and second generations


