Functional Joints

The movement required

The calculation of a functional joint requires movement of one segment relative to another segment. The algorithm (see below) searches for a point (or for a one degree of freedom joint, an axis) that is stationary relative to the 2 segments (or 2 sets of markers).

For joints modeled with 3 degrees of freedom, a movement trial in which the joint has modest range of motion about all three axes of rotation should be used for computing the functional joint. The movement should have sufficient range of motion that the computation statistics produce a reasonable stationary point, but the range of motion should not be too large because soft tissue artifact (e.g. movement of markers relative to the underlying skeleton) should be minimized. Our experience is that 20 degrees range of motion is usually sufficient to compute a stationary point reliably.

Several cycles of movement should be included in the motion, but it isn't clear exactly how many cycles there should be. We recommend more than 5 cycles, but users should experiment with their own data setups because the optimal number of cycles depends on many issues, including the joint tested, and the amount of soft tissue artifact present in the data.

The movement trial used is “contrived” specifically for the purpose of computing the functional joint. It should not be assumed that any movement (e.g. a walking trial) is sufficient. This “functional joint” trial should be separate from the movement data trials.

Begon M, Monnet T, Lacouture P (2007) Effects of movement for estimating the hip joint center. Gait and Posture 25, 353-359

This article defines characteristics of the movement profile that should be used for the functional joint calculation. This is not the only movement profile, and we recommend that users “play around” with other options until they find a movement that is reliable for their subject population. It is very difficult for many subjects to balance themselves for the functional trials, so many investigators support the subject and move the thigh for the subject.

If you are interested in the user actively performing the movement, we recommend a hula movement for the hip joints.

Functional Axis

If the joint is precisely one degree of freedom, it is possible to compute an axis, but it is not possible to compute a stationary point. In practice, joints aren't only one degree of freedom, and there is often some soft tissue artifact. This “noise” is then the source of data for computing a stationary point. The closer to one degree of freedom the less reliable the stationary point.

For the knee joint, we recommend projecting a lateral and medial knee marker onto the functional knee axis.

Warning. The functional knee axis is not the flexion/extension axis of your knee angle. Visual3D (and most software) use right handed orthogonal coordinate systems. The z-axis (axial direction) is defined by a vector between the distal and proximal ends of the segment. The x-axis lies in the frontal plane, but is forced to be perpendicular to the z-axis, which is unlikely to be parallel to the functional knee axis (it is just in the same plane).

Mathematically the functional axis computed has no direction. It could point medial or lateral to the segment, but the algorithm doesn't have enough information to know the correct direction.

Principles of the Gilette algorithm

adapted from: Schwartz MH, Rozumalski A (2005) A new method for estimating joint parameters from motion data. Journal of Biomechanics, 38, 107-116] Specify a segment coordinate system in which the motion capture data is to be resolved, and into which the landmark represented. For the case of the hip joint center, for example, the pelvis is considered a stationary coordinate system. Specify the motion capture markers attached to the moving segment. For the case of the hip joint center, markers attached to the thigh are used.

Algorithm
For all combinations of 3 frames (a,b,c) of data from the moving trial

Compute Finite Helical Axis (A) for frames a and b.

Compute Finite Helical Axis (B) for frames a and c.

Compute Finite Helical Axis (C) for frames b and c.

Use only helical axes for which the amount of rotation is greater than a minimum value (e.g. 5 degrees).

Compute the intersection of Axes A & B Compute the intersection of Axes A & C Compute the intersection of Axes B & C

With “real” data there is rarely an intersection of the two Finite Helical Axes. Compute the intersection as a line segment hat is the shortest route between the two axes. The “intersection” is the mid-point of this straight line. This intersection is considered one estimate of the joint center (eg a candidate joint center). All candidate joint centers are added to a “candidate array”. This candidate array gets very big very quickly because the number of permutations grows dramatically with the number of frames considered. The joint center returned is the mode of the set of candidates. The 3D mode is challenging to compute, so Visual3D estimates the mode. In principle Visual3D would use all candidates, but this is often impractical, so Visual3D samples the candidate array randomly. The number of candidates selected is an option defined by the user. The mode is computed as follows: Compute the mean value of all candidates Specify a sphere surrounding this location (the size of the initial sphere is an option defined by the user). A fairly big number is used typically, but the actual value hasn't been found to be particularly important) While() the number of candidates is greater than 500 (defined by the user)

The “candidate array” is reduced by removing all candidates that are not in the interior of this sphere.

Compute the median value of each component of the remaining candidates

Reduce the radius of the sphere (the percent decrease is an option defined by the user)

End While() When only 500 candidates remain, the function joint is defined as the mean value of these remaining candidates. **Note: The number of combinations gets very big very quickly and can easily crash the system if the user isn't careful. An option exists to select a subset of combinations at random (e.g. 2,000,000 combinations) as a representative sample.**

Principles of the Mayo Algorithm

adapted from Jensen E, Lugade V, Crenshaw J, Miller E, Kaufman K (2016) A principal component analysis approach to correcting the knee flexion axis during gait. Journal of Biomechanics, in press
Abstract

Accurate and precise knee flexion axis identification is critical for prescribing and assessing tibial and femoral derotation osteotomies, but is highly prone to marker misplacement-induced error. The purpose of this study was to develop an efficient algorithm for post-hoc correction of the knee flexion axis and test its efficacy relative to other established algorithms. Gait data were collected on twelve healthy subjects using standard marker placement as well as intentionally misplaced lateral knee markers. The efficacy of the algorithm was assessed by quantifying the reduction in knee angle errors. Crosstalk error was quantified from the coefficient of determination (r2) between knee flexion and adduction angles. Mean rotation offset error (αo) was quantified from the knee and hip rotation kinematics across the gait cycle. The principal component analysis (PCA)-based algorithm significantly reduced r2 ($p<0.001$) and caused αo,knee to converge toward 11.9±8.0° of external rotation, demonstrating improved certainty of the knee kinematics. The within-subject standard deviation of αo,hip between marker placements was reduced from 13.5±1.5° to 0.7±0.2° ($p<0.001$), demonstrating improved precision of the knee kinematics. The PCA-based algorithm performed at levels comparable to a knee abduction–adduction minimization algorithm (Baker et al., 1999) and better than a null space algorithm (Schwartz and Rozumalski, 2005) for this healthy subject population.

Defining a Functional Joint

Defining a Functional Joint

Functional Joints Post Processing

Functional Joints from Streaming Data

Add_Functional_Joint_Landmark

Add Functional Joint Landmark

Example: Functional Joint

Example: Functional Hip

Example: Functional Knee