

Correlative joint definition for motion analysis and animation

Nicolas Pronost, Anders Sandholm and Daniel Thalmann
Ecole Polytechnique Federale de Lausanne
Virtual Reality Laboratory

Abstract

In this paper we address the problem of creating accurate joint models from real motions while allowing scalability. We propose an automatic method to model, scale and simulate non-idealized joints from the external motion of markers. We demonstrate the method on the human knee joint modeling for musculoskeletal analysis and for character animation. The resulting joints, called correlative joints, are character and motion independent and rely on linear combinations of degrees of freedom calculated from multiple regression laws. We show that by using such models, inverse kinematics solvers find better solutions when tracking motions and solving constraints. Importing correlative joints into new models involves only minimal requirements on landmarks locations and no costly additional computations.

Keywords: Virtual Human, Motion Analysis and Modeling, Biomechanics, Animation

1 Introduction

Accurate track of human joint kinematics is instrumental for investigation of biomechanical mechanisms and motion simulation. Designing the mechanical behavior of a joint relies partially on the minimization of the modeling errors while retaining controllability. To be useful in simulation, a kinematic model has to be a good compromise between generality and accuracy. Indeed, both in biomechanics and animation, generality is required to allow the models

to be scalable to any subjects or virtual characters. Different levels of accuracy may therefore be required, from a natural looking to biomechanically validated motion.

Generality is usually provided by the possibility of matching data of a model to an experimental one. Hence, to model joint kinematics amounts to study scaled properties of a generic model. And then to ensure enough generality without loss of accuracy becomes a challenging problem.

In this paper we present a method to model accurate joint kinematics from real motions while allowing to be easily reused on any subject or virtual character. We can describe it as a functional method to compute joint kinematics which produces generic models. We demonstrate the method on the human knee, hence creating a highly correlative joint. We show that using this model, inverse kinematics (IK) solvers find better solutions. Results are presented both on the subject used to create the model and on several others where the model is scaled to match their morphologies. Improving IK tracking is crucial as it is a fundamental step of a biomechanical motion analysis and is one of the most used technique in animation to impose postures and constraints.

2 Related work

Joint kinematics and its measurement during a movement is an important requirement in the orthopedic and rehabilitation fields [1] as much as in animation [2]. In many laboratories, mo-

tion analysis systems can accurately measure 3D rotation of joints, and especially the knee joint [3, 4] which is intensively studied for its preponderant role in our daily activities.

For quantifying joint motion, the *International Society of Biomechanics* recommends the calculation proposed by Grood and Sun-tay [5]. Regarding angular displacements, their model needs as input the relative orientation between the Cartesian bone-embedded anatomical frames (BAFs). Several calibration procedures were proposed to transform 3D positions of markers as measured by standard motion capture systems into BAFs orientation [6, 7]. The most commonly used method to describe rotations of a joint is the Eulerian or Cardanic description, which decomposes the position of a limb with regard to its parent (or *vice versa*) into three sequential rotations along three pre-defined axes named flexion/extension, abduction/adduction and internal/external axial rotation [5]. These three pre-defined axes are also known as anatomical axes and can be defined by palpation of external anatomical landmarks [6]. Due to localisation errors leading to misorientation and overestimation of angular trajectories [8], researchers proposed different methods to measure joint kinematics. Methods which use both anatomical and functional calibration [9] are reported to be more repeatable than methods based only on anatomical calibration. And as Reinschmidt *et al.* [10] showed, the agreement between skin and bone markers based kinematics for ab/adduction and internal/external knee rotation ranged from good to virtually no agreement, and in some subjects, the errors exceed the actual motion. In these latter experiments, joint kinematics are very accurately tracked but on the other hand, using bicortical pins or *in vivo* medical imaging [11] to create models requires equipment, invasive procedure and time that one can most of the time not afford. Moreover, such methods can not be applied on animated virtual humans.

Musculoskeletal models including different joint descriptions have already been proposed and validated in literature [12]. For instance, Yamaguchi and Zajac [13] proposed a planar static model of the knee joint. A pathway for the instantaneous center of rotation was chosen that gives realistic orientations of the femur rel-

ative to the tibia. Techniques to model joints from surfaces have also been developed [14, 15]. By contrast, we do not want to model joints with any prior knowledge of inner constraints but from its external motion. Moreover, the resulting models have to be scalable to any real subject or animated character.

We demonstrate the whole process on the human knee (section 3) by creating a joint consisting of three rotational DoF and three correlative translational DoF. The three translations defining the position of the joint center are thereby moving along with the posture. The objective is to automatically determine these relationships. The input data are 3D positions of reflective markers from mocap which are fixed to the skin of real subjects (section 3.1). Multiple linear regression laws are then applied to model the correlative DoFs of the knee joint (section 3.2). Even knowing the artifact issue raised above, we show the robustness of our method on tracking results during an IK operation (section 3.3). We have embedded multiple linear relationships between DoFs in the biomechanics platform *OpenSim* [12] and ran simulations (section 4.1). The procedure to benefit from our models to animate virtual characters is presented in section 4.2 and more examples on creating correlative joints from motion data are showed in section 4.3.

3 Correlative knee joint creation procedure

3.1 Experimental set-up

In an articulated mechanical system, a six-component structure describes the transformation to go from a local body reference frame to the next one. In animated and biomechanical systems, the three translational components are often constants, defining an idealized joint. Unfortunately some joints can not reasonably be modeled with such approximation, they have to integrate moving centers of rotation to better represent the kinematics. As seen in section 2, knee joints greatly benefit from non-idealized kinematics. In this paper we demonstrate the procedure on the human knee by modeling the posture-dependent position of the knee joint center (KJC). The procedure is generic and

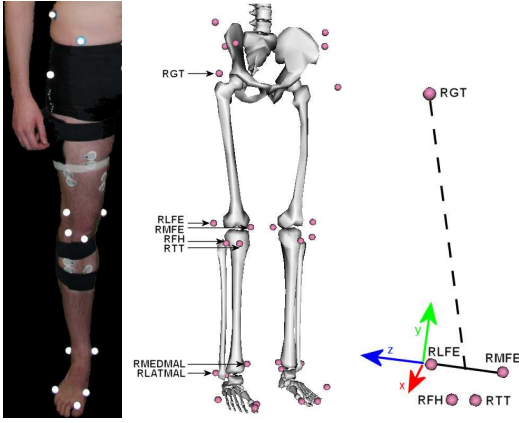


Figure 1: (left) Mocap setup. (middle) Virtual skin marker set. (right) Local reference frame.

may be applied to any body part and any joint of a mechanical system as soon as external markers can be located and tracked.

To develop the knee model, we used seven skin markers placed on anatomical landmarks (see Figure 1) : RGT (great trochanter), RLFE (lateral femur epicondyle), RMFE (medial femur epicondyle), RTT (tibial tuberosity), RFH (fibula head), RMEDMAL (medial malleolus) and RLATMAL (lateral malleolus). The local reference frame which defines also the three rotational DoF, is defined as follows (see right of Figure 1) :

- Z: The line parallel to a line connecting RMFE and RLFE, pointing to the right;
- Y: The line parallel to a line lying in the plane defined by the two RFE's and RGT, orthogonal to the Z-axis, pointing cranially;
- X: The line perpendicular to Y and Z, pointing anteriorly.

We captured different kinds of motions in order to study the influence of the movement on the kinematics of the knee joint and on the generality of our models. Regular gaits and crouch motions have been captured. We divided the crouch motions into 6 sub-motions: with 0° between the feet, with 30° , 60° , 90° , 120° and finally almost 180° (as much as possible for the subject). We captured six subjects performing 10 gaits and 5 crouches of each kind. This leads to more than 30,000 frames (*i.e.* elements of relationship) per subject. Moreover, we captured one standing motion for each subject for scaling purpose and for determining the rotation

axes as previously described. Crouch motions have been chosen as they involve a large range of knee bending, thus we will be able to evaluate our method over this large range.

3.2 Method

The first step consists in calculating the elements of the relationships from the motions. The procedure is depicted in Figure 2. Firstly, the standing motion of the subject is used to define the local coordinate system of the knee. A free joint (six DoF) is then placed at the knee. The three rotations are defined around these three local axis and the three translations are defined along the three local axis of the parent joint, the hip. Finally, we run an IK solver driven by the positions of the seven markers over time. For this task, we used the IK tool of OpenSim [12] where the same weight is used for the seven markers. As a result, we get for each frame, the local KJC positions and the three local angular trajectories. We obtain for each subject 10 datasets of gait and 6×5 datasets of crouch motion.

The second step consists in modeling these independent sequences by multiple variable functions. If linear relationships exist, the following equations will describe the three dimensional position (T_x, T_y, T_z) of the KJC according to the three angular values $(\alpha_1, \alpha_2, \alpha_3)$ around $(\vec{x}, \vec{y}, \vec{z})$ such that :

$$\begin{aligned} T_x &= c_{1_x} \times \alpha_1 + c_{2_x} \times \alpha_2 + c_{3_x} \times \alpha_3 + c_{4_x} \\ T_y &= c_{1_y} \times \alpha_1 + c_{2_y} \times \alpha_2 + c_{3_y} \times \alpha_3 + c_{4_y} \\ T_z &= c_{1_z} \times \alpha_1 + c_{2_z} \times \alpha_2 + c_{3_z} \times \alpha_3 + c_{4_z} \end{aligned} \quad (1)$$

To determine the three sets of coefficients (c_1, c_2, c_3, c_4) , we calculate multiple linear regression laws on the datasets. Multiple regression solves for the unknown coefficients by minimizing the sum of the squares of the deviations of the data (angular trajectories) from the model (KJC positions) using a least-squares fit. The algorithm is as follows:

1. create the column matrix of experimental data E :
 $E = T_x(f), f \in [1 \dots F]$ (respectively T_y and T_z) where F is the number of frames of the motion
2. create the 3 column matrices

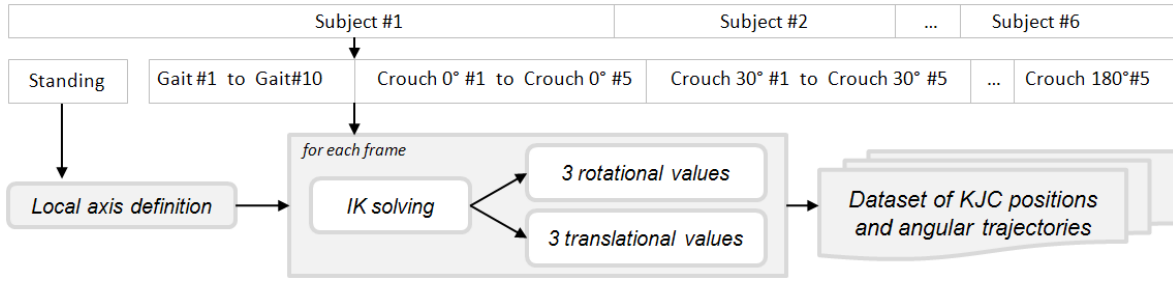


Figure 2: Overview of the analysis procedure used for all subjects, motions and sub-motions, leading to $10 + 6 \times 5$ datasets per subject relating KJC positions to angular trajectories.

of angular trajectories

$$\begin{aligned} &\alpha_1(f) \\ &\alpha_2(f), f \in [1 \dots F] \\ &\alpha_3(f) \end{aligned}$$

3. create the design matrix M :

$$M = \begin{pmatrix} 1 & \alpha_1(1) & \alpha_2(1) & \alpha_3(1) \\ \vdots & \vdots & \vdots & \vdots \\ 1 & \alpha_1(F) & \alpha_2(F) & \alpha_3(F) \end{pmatrix}$$

4. solve the equation :

$$C = M \setminus E$$

where $C = [c_{4_x} c_{1_x} c_{2_x} c_{3_x}]$ (respectively y and z)

5. validate the result

$$\text{valid} = (\max(|M \times C - E|) < \min(E))$$

In our study, all the resulting regression laws were validated: the maximums of the absolute values of the deviation of the data were always smaller than the minimums of the data values. This certifies that the model accurately follows the data and that linear functions are enough to model such joints. Mathematical justifications to solve the equation of item 4 can be found in the *Matlab*'s backslash operator documentation.

Table 1 presents the coefficients computed from the crouch motions. We do not show the coefficients specific to the sub-motions (different degrees between the feet), but the section 3.3 presents the modeling variations using a global-based law and a sub-motion-based law.

We can notice here the importance of the c_4 values which represent the constant components of the KJC position. These values are strongly related to the morphology of the subject. The taller the subject, the larger the c_4 value. This property shows already that the laws are subject-specific, *i.e.* one cannot directly use

a correlative joint on any subject/virtual character. Nevertheless, this information can be used to develop simpler models. Indeed, the constant terms c_4 are definitively a good guess for setting a three DoF idealized joint. The following section presents evaluations of the laws regarding to sub-motions and simulation results of the kinematics of the leg.

3.3 Evaluation

In the first part of this section we present the accuracy induced by modeling the knee joint by multiple linear functions. Table 2 shows the average deviation between the KJC positions from IK and the positions estimated from the linear functions described from equation 1. The mean deviation is about 1.55 mm for gait motions and 1.32 mm for crouch motions. If we use a sub-motion specific function (*i.e.* using coefficients generated with the corresponding angles between the feet), the mean deviation decreases to 0.98 mm ($\sim 26\%$ less). That indicates that as expected, the sub-motion models are more accurate as they are more specific and that the benefit is not negligible. It will certainly also be the case for other subsets of motion such as for locomotion consisting of gait (speed-dependent), run, jump, side-walk, etc.

In Figure 3, we propose visualizations of the deviations in the crouch models. The three axes define the 3D space of the knee joint orientation. The color represents the deviation, from blue (minimal deviation in the subject dataset) to red (maximal deviation). We can notice that the deviation is mainly homogeneously distributed. All postures are equally well modeled. The angular trajectories are closed to each others for a given subject, facilitating the linear relation-

	Subject1	Subject2	Subject3	Subject4	Subject5	Subject6
T_x						
c_{1_x}	0.0573	0.1142	0.0496	0.0322	-0.0007	0.0404
c_{2_x}	0.0361	0.0340	0.0470	0.0328	0.0348	0.0351
c_{3_x}	0.0582	0.3426	0.0788	0.1105	-0.0297	0.0203
c_{4_x}	0.0748	0.0691	0.0783	0.0637	0.0713	0.0878
T_y						
c_{1_y}	0.0306	0.0501	0.0657	0.0613	0.1460	0.1001
c_{2_y}	-0.0442	-0.0366	-0.0364	-0.0143	-0.0260	-0.0585
c_{3_y}	-0.1055	-0.1093	0.0330	0.0044	0.1869	0.2011
c_{4_y}	-0.5035	-0.4274	-0.4920	-0.4662	-0.5105	-0.5310
T_z						
c_{1_z}	-0.0199	-0.0100	0.0362	-0.0536	0.0000	-0.0586
c_{2_z}	0.0210	0.0515	0.0116	0.0376	0.0148	0.0277
c_{3_z}	-0.2218	-0.1856	0.0142	-0.2319	-0.0292	-0.3299
c_{4_z}	0.0346	0.0414	0.0318	0.0260	0.0292	0.0347

Table 1: Multiple regression coefficients for the crouch motions

	Subject1	Subject2	Subject3	Subject4	Subject5	Subject6
Crouch	1.3 / 1.1	2.0 / 1.4	0.9 / 0.7	1.1 / 0.9	1.6 / 1.0	1.0 / 0.8
Gait	1.1 / -	0.9 / -	1.5 / -	1.7 / -	3.1 / -	1.0 / -

Table 2: Average deviation (mm) between the positions of KJC from IK and the correlative positions from the regression laws (motion specific / sub-motion specific).

ships. Even if more complex equations might be defined, it seems that to use linear relationships is enough to model such joint kinematics, as the validation step of the algorithm tended to show.

4 Results

4.1 Integration and simulation in *OpenSim*

In order to demonstrate the capabilities of the method, we present in this section quantitative results on motion analysis. We want to show improvements in the tracking of captured motions, using the IK tool embedded in *OpenSim*. Results proposed in this section are obtained with the *Interior Point Optimizer* method (IPOPT), but equivalent results have been calculated with *Feasible Sequential Quadratic Programming* (FSQP) [16] and a Jacobian-based method. To evaluate the tracking, we use the sum of the frame-by-frame RMS distance between the experimental markers and the virtual markers (attached to the model and moved by

the IK solution).

To take into account our new joint description, we have implemented a new kind of constraint in *OpenSim*. This constraint describes the linear relationship between one DoF and others (no limitation on the number and the kinship). Let us consider the correlative DoF X which depends on the values of n other DoFs $Y_{1...n}$:

$$X = \left(\sum_{i=1}^n c_i \times Y_i \right) + c_{n+1} \quad (2)$$

where $c_{1...n+1}$ are scalars.

We do not require these DoFs to be all different but each of them has to be different from X . This new feature enables us to directly deal with multiple linear relationships. Then for instance to model the crouch-specific knee, $c_{1...n+1}$ are set to the values of table 1.

We proceed several simulations with this feature. Firstly, we compared three models of the knee joint. One that we call *3DOF*, is made of two translational and one rotational DoF. Its rotation axis was determined by a planar static model and the two translations depend on the ro-

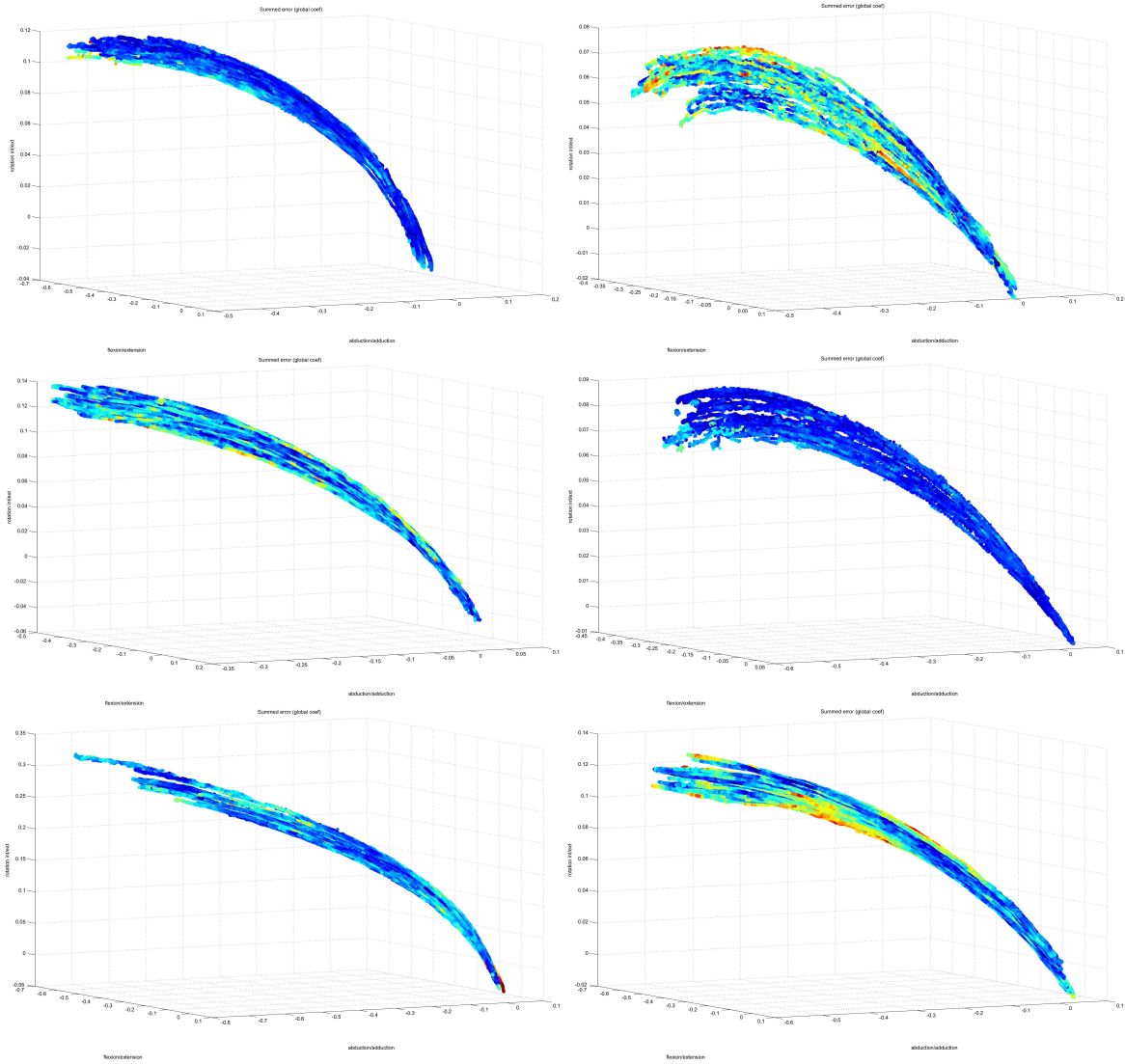


Figure 3: Deviation between the positions of KJC from IK and the positions modeled by multiple linear functions (crouch motions of the six subjects).

tational value and are described by splines [13]. A second, called *IDOF* is made of only one rotational DoF, around the same axis than the previous model. The last one is ours (3^+DOF) made of three rotational and three correlative translational DoF. The location of the KJC of the *IDOF* model is set to the default position from the *3DOF* model which has been biomechanically validated. We did not use the c_4 values to place this KJC as we indeed want to compare our model to validated literature models.

In table 3, we present average values of IK errors during the track of mocap data using these three models. Errors were obtained on 10 gaits and 10 crouches that are different from the motions used to create our models. The re-

ported RMS errors are not normalized by neither the number of frames nor the number of markers. The error differences between the gaits and the crouches mainly come from the number of frames (a complete crouch takes more time to perform than a gait cycle). As expected, we observe that the $3DOF$ error is lower than the $1DOF$ error and that the 3^+DOF error is lower than both of them. We present error values only for one subject, but the same experiment was done on the others. Results show that IK errors are reduced by :

- $44(\pm 8)\%$ between 3^+DOF and $1DOF$ for gait motions
- $31(\pm 5)\%$ between 3^+DOF and $3DOF$ for gait motions

- 71(± 9)% between 3⁺DOF and 1DOF for crouch motions

- 66(± 6)% between 3⁺DOF and 3DOF for crouch motions

This demonstrates the benefit induced by our model. Indeed, such large improvements of IK tracking have a direct influence on the motion analysis accuracy and on the naturalness of animated motions.

Subject1	1DOF	3DOF	3 ⁺ DOF
Gait	3.02	2.48	1.19
Crouch	272.94	261.46	76.34

Table 3: Average values of RMS distances (m).

The previous results were obtained by applying the knee model to the corresponding subject and motion. Now, we study what happens if a specific knee model is used to simulate another subject or the same subject but on another kind of motion. We therefore set up two experiments.

In the *experiment 1*, we scale the model of subject 2 to subject 1 (denoted $model1_{from2}$) and track gaits and crouches of subject 1. To scale the model, we update the coefficients as follows:

$$c_{i_d}^{scaled} = \frac{c_{i_d}(s_1+s_2)}{2}, \quad i \in [1, 2, 3, 4], \quad d \in [x, y, z] \quad (3)$$

with

$$s_1 = \frac{|\overrightarrow{RGT}_{model1} \overrightarrow{RLFE}_{model1}|}{|\overrightarrow{RGT}_{model2} \overrightarrow{RLFE}_{model2}|}$$

$$s_2 = \frac{|\overrightarrow{RMFE}_{model1} \overrightarrow{RLFE}_{model1}|}{|\overrightarrow{RMFE}_{model2} \overrightarrow{RLFE}_{model2}|}$$

This scaling method is based on the same approach that classical methods in the literature [12] where such linear scaling function has been validated over the range of human bodies.

We then compare both tracking results of $model1_{from2}$ and the specific model of subject 1 (denoted $model1$). The table 4 shows the resulting errors from 10 gaits and 10 crouches. We first see that the error using a scaled model is, as expected, larger than the error using our specific model. We also notice that the scaled error is much lower than the error of the 3DOF model. That implies that one can use a generic 3⁺DOF model to perform analysis on another subject;

IK results will still be better than a scaled 3DOF model from literature. This property was verified by carrying out the same experiment on the other models/subjects/motions, not detailed here.

Subject 1	$model1$	$model1_{from2}$
Gait	76.34	87.35
Crouch	1.19	2.10

Table 4: Experiment 1 : tracking errors (m) of $model1$ and $model1_{from2}$.

In the *experiment 2*, we use the 3⁺DOF model defined from gait motions to track crouch motions and *vice versa*. The results on subject 1 are presented in table 5. The error induced by a model defined from a different kind of motion is about 40% for gait and 70% for crouch. On the other hand, the errors are still lower than the error from the 3DOF model, thus showing the benefit of our models. Gaits are better tracked in the experiments as there is less angular trajectories to extrapolate from a crouch model than the opposite.

Subject1	3 ⁺ DOF gait	3 ⁺ DOF crouch
Gait	1.19	1.68
Crouch	130.34	76.34

Table 5: Experiment 2 : tracking errors (m) of motion specific models.

4.2 Application to virtual character animation

In this section we give information about how to use our correlative joints in virtual character animation. We denote $model_{corr}$ the model on which we have created correlative joints and $model_{anim}$ the model of the virtual character we want to animate.

Firstly, the correlative joints from $model_{corr}$ have to be defined with the same orientation than $model_{anim}$ as follows:

1. To choose three anatomical markers on the parent body of the joint on $model_{corr}$ (for example RGT, RMFE and RLFE for the knee);

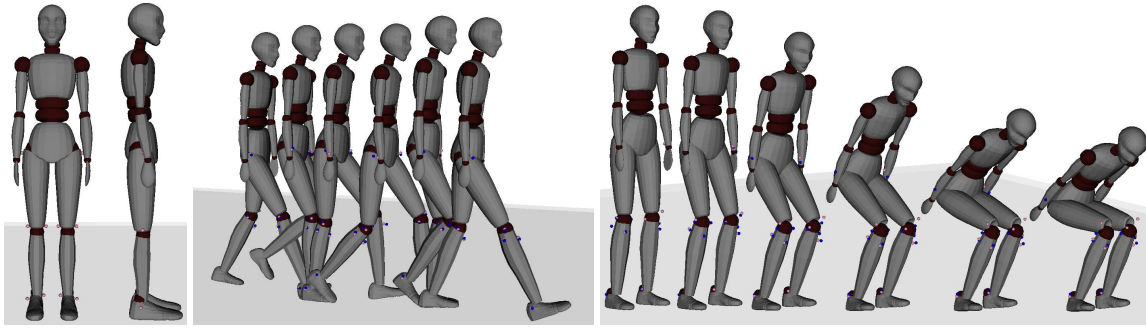


Figure 4: The virtual markers are placed on the predefined anatomical landmarks (left) and used to animate the lower limb during gait (middle) and crouch motions (right).

2. To place three virtual markers on $model_{anim}$ on the equivalent landmarks (see left of Figure 4);
3. To scale the coefficients of the relationships according to the respective markers distances (as in equation (3));
4. To create the local coordinate system from the markers for both models (as in section 3.1 for the knee);
5. To apply the two successive transformations to compute the local joint orientation in $model_{anim}$.

The proper behavior of the joint will depend on the accuracy of the markers positioning. This issue, which is also shared by motion analysis setups, could become tricky for not-really-human characters.

Secondly, the animation engine has to be able to evaluate equation 1 when the value of a DoF involved in a correlative joint changes. The implementation of this ability depends on the engine, but most of them provide script programming which can be used for this purpose. As a consequence, when editing a motion, either from mocap or keyframing, inverse and direct kinematics solvers will benefit from the correlative joint feature (see Figure 4).

While creating or editing a new motion involving a correlative joint, the animator does not require necessarily to create a motion specific model as described in this paper. Indeed, as soon as the model has been created on a sufficient range of value, the model is valid for any motion within this range (see experiment 2). Then

assuming that correlative models have been created from biomechanical datasets and accurate motion analysis (for instance using bicortical pins), the development of realistic virtual character animations is straight forward and without additional cost.

4.3 More correlative joints from motion

We have demonstrated the method on the human knee joint. This joint is mechanically complex and a high-degree model (3 rotations and 3 correlative translations) was necessary to represent its kinematics. Our method can also be used to model simpler joints. The Figure 5 illustrates three of our experiments in retrieving lower and higher pair joints from the motion of external markers placed over the different body parts.

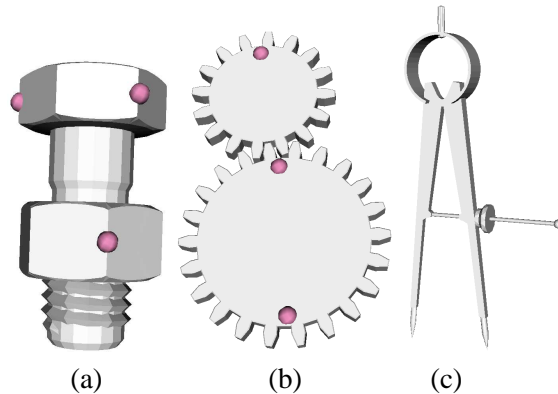


Figure 5: Lower and higher pair mechanisms modeled with correlative joints.

An helical joint is a one rotational DoF lower pair mechanism providing a single-axis translation. The typical illustration is the screw-bolt system (Figure 5(a)). The translation is gener-

ated by the geometry of the threads on the rod. The relationship between the rotation of the bolt and its translation is linear if the threads are homogeneous. Using our method we can automatically create a joint describing this kinematics from the motion of markers attached to the screw and the bolt (without prior knowledge of the threads). The correlative translation T is then given by : $T = \alpha \times R + c$, where R is the rotation of the bolt. α and c are constants determined by the regression law. They can respectively be associated to the thread-pitch and the root-location parameters.

A transmission is a system providing speed and torque conversions from a rotating source to another device using gear ratio. The typical illustration is a tandem of gears (Figure 5(b)). The rotation of the powered gear is transmitted to the next one through the contact of tooth. The relationship between the two rotations is linear. Our method can as well create this higher pair joint from motion. The correlative rotation R_b of the second gear is : $R_b = \beta \times R_a$. β is calculated by the regression law and corresponds to the gear ratio.

The Figure 5(c) shows a combination of these two types of joint within a more complex system. This bow drawing compass with a lateral wheel has only one input parameter: the wheel rotation. This rotation R drives the rotations of the two legs (R_{leg1}, R_{leg2}), the rotation of the rod R_{rod} attached to $leg1$ and going through $leg2$, and the translation T of the wheel along the rod. Our method automatically calculates the relationships (α_1 to α_5) between these parameters and R providing us with the kinematical model of the compass :

$$\left\{ \begin{array}{l} R_{leg1} = \alpha_1 \times R \\ R_{leg2} = \alpha_2 \times R \\ R_{rod} = \alpha_3 \times R \\ T = \alpha_4 \times R + \alpha_5 \end{array} \right.$$

The resulting model consists of three transmission joints and one helical joint. Using this model, to solve the IK problem where the distance between the two legs is a constraint will only involve the optimization of the rotation of the wheel R . In this case, an analytic solution can even be used as the dimension of the problem is now the same size as the constraint. These examples show the ability of the method

to retrieve and simulate as well simple kinematical models. We can also notice that even if the model is defined in a particular coordinate system, there is no need to know in advance the respective direction of the translations and rotations involved in the relationships. The user has only to decide the DoFs that will be in control and the correlative ones.

5 Conclusion

In this paper we have presented a method to create an accurate and scalable model of a joint from the motion of external markers. The computed joint can be reused on any real subject or virtual character with only minimal requirements on anatomical landmarks. To perform direct and inverse computational approaches, no costly additional algorithm is involved. Only the evaluation of equation 1 has been added, composed of few additions and multiplications.

Many complex biological joints can be modeled using our technique. We demonstrated that the human knee joint can be modeled using a correlative joint. And we believe that most of human joints, even complex as the shoulder, will benefit from such mechanical description.

We showed that by using our model, inverse kinematics solvers find better solutions both on the subject used to create the model (error reduced by 53 up to 80%) and on others subjects where the relationships were scaled to match the new morphologies (error reduced by 25 up to 41%). Biomechanical simulations will better match the experimental data allowing more accurate estimation of forces, muscle activations etc. And character animation will benefit in more natural motions with no major cost.

Finally, we remind the good performances of our model compared to 1DOF and 3DOF models of literature. We believe that this work contributes to the development of reliable methods of simulation for animation and biomechanical analysis.

Acknowledgements

This work is supported by the European Marie Curie Program under the 3D ANATOMICAL HUMAN project (MRTN-CT-2006-035763).

References

- [1] R.J. De Asla, L. Wan, H.E. Rubash, and G. Li. Six dof in vivo kinematics of the ankle joint complex: Application of a combined dual-orthogonal fluoroscopic and magnetic resonance imaging technique. *Journal of Orthopaedic Research*, 24(5):1019–1027, 2006.
- [2] N.I. Badler, C.B. Phillips, and B.L. Webber. *Simulating humans: computer graphics animation and control*. Oxford University Press, Inc., New York, NY, USA, 1993.
- [3] A.D. Georgoulis, A. Papadonikolakis, C.D. Papageorgiou, A. Mitsou, and N. Stergiou. Three-dimensional tibiofemoral kinematics of the anterior cruciate ligament-deficient and reconstructed knee during walking. *American Journal of Sports Medicine*, 31(1):75–79, 2003.
- [4] U. Della Croce, V. Camomilla, A. Leardini, and A. Cappozzo. Femoral anatomical frame: assessment of various definitions. *Medical Engineering & Physics*, 25(5):425 – 431, 2003.
- [5] E.S. Grood and W.J. Suntay. A joint coordinate system for the clinical description of three-dimensional motions: Application to the knee. *Journal of Biomechanical Engineering*, 105(2):136–144, 1983.
- [6] A. Cappozzo, F. Catani, U. Della Croce, and A. Leardini. Position and orientation in space of bones during movement: anatomical frame definition and determination. *Clinical Biomechanics*, 10(4):171 – 178, 1995.
- [7] N. Hagemeister, G. Parent, M. Van de Putte, N. St-Onge, N. Duval, and J. de Guise. A reproducible method for studying three-dimensional knee kinematics. *Journal of Biomechanics*, 38(9):1926 – 1931, 2005.
- [8] U. Della Croce, A. Cappozzo, and D.C. Kerrigan. Pelvis and lower limb anatomical landmark calibration precision and its propagation to bone geometry and joint angles. *Medical and Biological Engineering and Computing*, 37(2):155–161, 1999.
- [9] A.G. Schache, R. Baker, and L.W. Lamoreux. Defining the knee joint flexion-extension axis for purposes of quantitative gait analysis: An evaluation of methods. *Gait & Posture*, 24(1):100 – 109, 2006.
- [10] C. Reinschmidt, A.J. Van den Bogert, A. Lundberg, B.M. Nigg, N. Murphy, A. Stacoff, and A. Stano. Tibiofemoral and tibioalcanal motion during walking: external vs. skeletal markers. *Gait & Posture*, 6(2):98 – 109, 1997.
- [11] J. Bingham and G. Li. An optimized image matching method for determining in-vivo tka kinematics with a dual-orthogonal fluoroscopic imaging system. *Journal of Biomechanical Engineering*, 128(4):588–595, 2006.
- [12] S.L. Delp, J.P. Loan, M.G. Hoy, F.E. Zajac, E.L. Topp, and J.M. Rosen. An interactive graphics-based model of the lower extremity to study orthopaedic surgical procedures. *IEEE Transactions on Biomedical Engineering*, 37(8):757–767, 1990.
- [13] G.T. Yamaguchi and F.E. Zajac. A planar model of the knee joint to characterize the knee extensor mechanism. *Journal of Biomechanics*, 22(1):1 – 10, 1989.
- [14] W. Shao and V. Ng-Thow-Hing. A generic joint component framework for realistic articulation in human characters. In *ACM Symposium on Interactive 3-D Graphics*, pages 11–18, 2003.
- [15] S.H. Lee and D. Terzopoulos. Spline joints for multibody dynamics. *ACM Trans. Graph.*, 27(3):1–8, 2008.
- [16] C.T. Lawrence and A.L. Tits. Nonlinear equality constraints in feasible sequential quadratic programming. *Optimization Methods and Software*, 6:265–282, 1996.