Estimation of Joint Types and Joint Limits from Motion Capture Data

Morten Engell-Nørregård
Department of Computer Science,
University of Copenhagen, Denmark
morten@eurobeast.dk

Kenny Erleben
eScience Center,
University of Copenhagen, Denmark
kenny@diku.dk

ABSTRACT

It is time-consuming for an animator to explicitly model joint types and joint limits of articulated figures. In this paper we describe a simple and fast approach to automated joint estimation from motion capture data of articulated figures. Our method will make the joint modeling more efficient and less time consuming for the animator by providing a good starting estimate that can be fine-tuned or extended by the animator if she wishes, without restricting her artistic freedom. Our method is simple, easy to implement and specific for the types of articulated figures used in interactive animation such as computer games. Other work for joint limit modeling consider more complex and general purpose models. However, these are not immediately suitable for inverse kinematics skeletons used in interactive applications.

Keywords: Joint-Limits, Joint-Types, Articulated Figures.

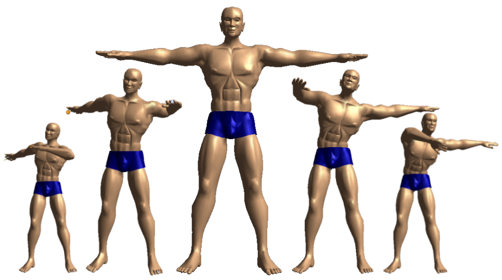


Figure 1: Rigging joint limits can be time-consuming when using inverse kinematic animations as illustrated here. Our method can be used to reduce production time for modeling articulated figures.

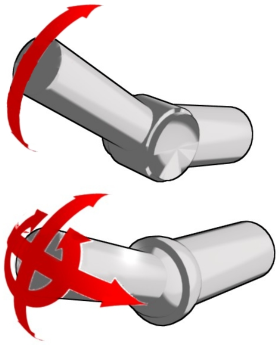


Figure 2: The two joint types considered in this paper are the hinge joint which rotates around a single axis and the ball joint which rotates freely in 3 dimensions.

1 AN ARTISTIC TIME-SAVER

Interactive applications, computer games, and virtual reality applications often contain human characters, creatures, and robots modeled as articulated figures. The articulated figures are brought to life run-time using techniques of motion blending [10], inverse kinematics [16, 5] or forward dynamics [11]. Figure 1 shows an example using inverse kinematics. The articulated figures must be created by an animator before being used run-time. This is termed character rigging. The most wide-spread technique for character rigging consists of modeling a character skin and then creating a bone skeleton which is coupled to the skin by specifying vertex weights. Hence the two terms boning and skinning.

Typical tools such as Autodesk Maya®, Autodesk 3ds Max® or Blender are used by artistic people for this work-process [13, 1, 2]. The two most common joint types are the hinge joint and the ball joint as shown in Figure 2. Box-constraints for each of the degrees of freedom are used to restrict the motion of the joints. Character rigging can be time-consuming and difficult. It is our goal to alleviate this problem by providing a simple method to assist animators during the boning process. We present a method that is able to optimize the work-flow when modeling the joint-types and joint-limits of a bone skeleton without restricting the artistic freedom of the artist. Our idea is to provide the animator with a starting estimate for the two joint types together with values for the box-constraints. The animator can then fine-tune or extend the estimate.

Permission to make digital or hard copies of all or part of this work for personal or classroom use is granted without fee provided that copies are not made or distributed for profit or commercial advantage and that copies bear this notice and the full citation on the first page. To copy otherwise, or republish, to post on servers or to redistribute to lists, requires prior specific permission and/or a fee.

Plzen, Czech Republic.
Copyright UNION Agency – Science Press

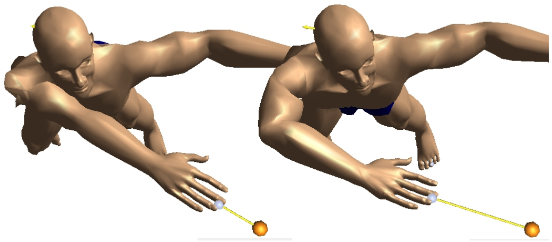


Figure 3: The same pose calculated with joint limits on the right and without on the left. Observe that the joint limits clearly give a more realistic pose.

Joint types and joint limits are important for the immersing of end-users. A realistic model will be able to bring the characters to life as illustrated in Figure 3. This have inspired us to use motion capture data. Motion capture data are already available in most content pipelines and hierarchical structure and motion of the articulated figures are in some cases already derived from this data. We propose to use the same data for estimating the joint types and joint limits of the articulated figure.

It should be noted that our method could also be applied to exemplar based motion created by artistic people using traditional key-framing techniques. Thus motion capture data is not a necessary condition. In Figure 1 hand animated exemplar motion was used in place of motion capture data.

1.1 Previous Work

Estimation of skeletons and animations from marker points are known [8, 9] . We do not consider the skeleton structure or the motions themselves, but rather the motion *range* of the skeleton.

A qualitative kinematic model for the shoulder complex is presented in [4]. The shoulder complex is viewed as a two mechanism system. Articulated human figures in computer animation typically models the shoulder complex by two bones one "scapular/clavicular" like bone connected to the humerus bone.

In [12] the human shoulder is modeled by a hierarchical inverse kinematics skeleton. They model the scapula-thoracic joint by breaking the closed chain and using the scapula as an end-effector constrained to the surface of an ellipsoidal thorax. Joint limits are modeled using joint sinus cones. Joint sinus cones are more general than the box-constraints used in for instance computer games.

Shoulder joint limits are modeled in [7] using quaternion field boundaries. From motion capture data the authors sample the orientation of the shoulder joint using a quaternion representation. The quaternion field boundaries are not easily adopted to the box-constraints and the back-projection methods used to deal with joint limits in inverse kinematics.

In [14] a general joint component framework is described. A joint component framework is derived and by connecting the components in networks one create the joint set functions. The paper presents components corresponding to rotation joints with moving rotation center and dependent joint parameters among many others. The authors extend joint reach cones [15] to deal with a moving rotation center. One concern is that the rotation joint component is a non-smooth function making it non-obvious how to use traditional inverse-kinematics methods.

Recently [11] an implicit parameterization of the joint motion by B-splines have been suggested for multi-body dynamics. Due to the implicit nature of the motion joint limits are not modeled explicitly.

In most of the work cited above the authors leave the actual setup of the joint limits to the artist. Our work is mostly similar to the ideas presented in [7]. Our approach differ in that we consider the bone skeletons used in present software by artists. Further we break down the problem into a two-phase process of first determining the joint type and then the joint limits.

2 MOTION ANALYSIS AS A TWO-PHASE PROCESS

We want to analyze the motion of a single specific bone of an articulated figure. Our task is to describe the bones motion relatively to its parent. For instance by determining whether a bone is connected to its parent through a hinge joint type and further what the physical parameters of that hinge joint are, or phrased differently the valid range of motion of the hinge joint.

We know the motion as a sequence of relative bone transformation samples. The i^{th} transformation sample is represented as,

$$T_i = \begin{bmatrix} Q_i & \vec{t}_i \\ \vec{0}^T & 1 \end{bmatrix}. \tag{1}$$

The bone transformation describes the relative coordinate transformation between the joint frame of the bone and the joint frame of the parent bone. Observe that we have mis-used the usual notation of homogeneous coordinate matrices by letting the rotational part of the transformation matrix be represented by the unit-quaternion Q_i. The translational part is given by the vector $\vec{t}_i$. We are also given a unique relative transform of the bone known as the bind pose,

$$T_b = \begin{bmatrix} Q_b & \vec{t}_b \\ \vec{0}^T & 1 \end{bmatrix}. \tag{2}$$

One can think of the bind-pose as the default pose of the bone. Our task is two-fold. Firstly we wish to determine the joint type of the bone and secondly we wish to estimate the joint parameters once we know the joint type.

The motion samples are obtained by sampling motion capture data using key-frame interpolation. The skeleton and animations used, were obtained from the Carnegie Mellon University motion capture database.

2.1 Discriminating Joints

To determine the joint-type we will try to determine the dimensionality of the motion space of the bone. We consider human motion it is therefore unlikely that translational motion has any major impact and we disregard it completely from further analysis. Secondly the human body can at a coarse level be considered to consist of only two joint archetypes: The ball joint and the hinge joint. In conclusion we are only interested in being able to discriminate between these two joint types.

If we consider the motion samples then T_i can be used to show if any motion happens. In particular the Q_i part is of interest. This is a unit quaternion and can be interpreted as an axis-angle representation of rotation

$$Q_i = \begin{bmatrix} s_i \\ \vec{v}_i \end{bmatrix} = \begin{bmatrix} \cos\left(\frac{\theta_i}{2}\right) \\ \vec{n}_i \sin\left(\frac{\theta_i}{2}\right) \end{bmatrix} \tag{3}$$

where $\vec{n}_i$ is a unit-vector and θ_i is the rotation angle around the rotation axis defined by $\vec{n}_i$.

Consider the behavior of $\vec{v}_i$. If the joint is rigid that means we have no motion at all and we must have the same $\vec{v}_i$ for all values of i. Next imagine that we have a hinge joint type. This means that the relative motion is a rotation around a fixed rotation axis. Since the axis is unchanged all $\vec{v}_i$'s must be parallel. However, the magnitudes are varying in the range $[-1..1]$. This implies that looking at the $\vec{v}_i$'s they must all lie along a radial line segment possible passing through the origin. Finally, in the case of a ball joint type the rotation axis is constantly changing. Looking at the $\vec{v}_i$'s we will have a spherical shell. Thus the space of $\vec{v}_i$'s now span a volume. Observe the first case is 0-dimensional, the second case 1-dimensional and the final case is 2 and 3 dimensional. Figure 4 illustrates the dimensions for a hinge and ball joint.

The dimensionality can be determined by performing an eigen-value analysis of the $\vec{v}_i$ point set. Let

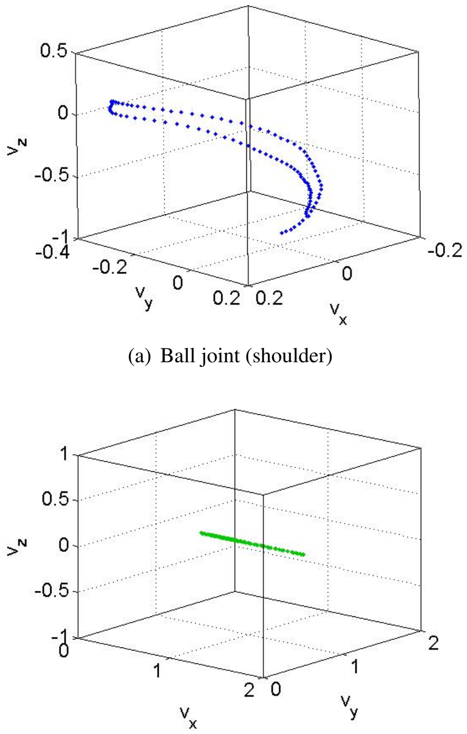
$$\vec{c} = \frac{1}{N}\sum_{i}^{N} \vec{v}_i \tag{4}$$

where N is the number of samples. The covariance matrix, $C \in \mathbb{R}^{N\times N}$, is

$$C = PP^T \tag{5}$$

where

$$P = \left[(\vec{v}_1 - \vec{c}) \quad \cdots \quad (\vec{v}_N - \vec{c}) \right]. \tag{6}$$



(a) Ball joint (shoulder)

(b) Hinge joint (elbow)

Figure 4: Samples of the vector part of the quaternion Q_i plotted in the motion space of the joint rotation axis. Notice how the shoulder joint (a) extends in all 3 dimensions while the hinge joint (b) only extends in 1 dimension.

Next we perform an eigen-value decomposition of the covariance matrix,

$$C = VDV^T, \tag{7}$$

where $V \in \mathbb{R}^{N\times N}$ is an othogonal matrix of unit eigenvectors and $D \in \mathbb{R}^{N\times N}$ is the diagonal matrix of corresponding eigen-values.

If all diagonal entries of D are zero we have 0-dimensionality. If we have 1 non-zero diagonal entry in D then we have 1-dimensionality and so on. We have now solved the first phase of the process, being able to determine the joint type. In the next phase we must estimate the joint parameters that describe the physical range of valid motion.

2.2 Estimating Joint Parameters

Once we know the joint type it becomes easier to estimate the joint parameters. In the following we will proceed by a case-by-case analysis of each joint type.

The case of the immovable joint we handle by imaging that the joint is a ball joint type. We will find joint parameter values equivalent to the fixed pose of the joint. The joint limits will then be set equal to these fixed joint parameter values, resulting in a fixated ball joint.

For the hinge joint type, we can easily find the rotation axis as

$$\vec{u} = \frac{\sum_{i=1}^{N} \vec{v}_i}{\| \sum_{i=1}^{N} \vec{v}_i \|}. \tag{8}$$

Here we exploited the quaternion equivalence to the axis-angle representation of rotations. The bind-pose signifies the current pose value so we compute the corresponding rotation angle,

$$\theta = 2 \operatorname{atan}_2 \left(\cos\left(\frac{\theta_b}{2}\right), \left| \sin\left(\frac{\theta_b}{2}\right) \right| \right) \tag{9}$$

$$= 2 \operatorname{atan}_2 \left(\cos\left(\frac{\theta_b}{2}\right), \| \vec{n}_b \| \left| \sin\left(\frac{\theta_b}{2}\right) \right| \right) \tag{10}$$

$$= 2 \operatorname{atan}_2 \left(s_b, \| \vec{v}_b \| \right), \tag{11}$$

where s_b and $\vec{v}_b$ are given by Q_b from (2). Next we may compute

$$\theta_{\max} = \max_i \{ 2 \operatorname{atan}_2 (s_i, \| \vec{v}_i \|) \} \tag{12a}$$

$$\theta_{\min} = \min_i \{ 2 \operatorname{atan}_2 (s_i, \| \vec{v}_i \|) \} \tag{12b}$$

In this analysis we have overlooked two important aspects. Firstly, the bind-pose may not be included in the sampled motion. Thus we can not be sure that, $\theta_{\min} \leq \theta \leq \theta_{\max}$. Secondly, rotation angles are periodic and the usual min-max approach for getting interval bounds are therefore flawed. For now we will overlook the problems and defer them to Section 2.3 and 2.4.

Rotational joints and their limits are often descriped by Euler parameters in motion capture formats and inverse kinematics methods. Thus for the ball joint type we will work with the Euler parameters, we have chosen a ZYZ convention, see Appendix A for details. The joint parameters are computed for the bind-pose as,

$$(\phi, \psi, \theta) = \mathrm{ZYZ}(Q_b) \tag{13}$$

Currently our joint-limit functions only allow for a boxed domain. This is of course a crude approximation to real-world ball joint types of humans. However, it greatly simplifies our task. We proceed by converting all rotational motion samples into the equivalent Euler parameters,

$$(\phi_i, \psi_i, \theta_i) = \mathrm{ZYZ}(Q_i) \tag{14}$$

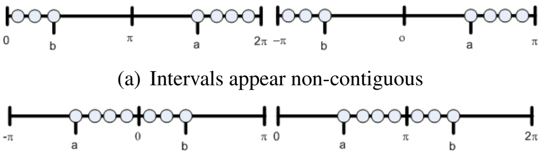
Next one can find a tight fitting box around the Euler samples

$$\phi_{\min} = \min_i \{\phi_i\} \quad \text{and} \quad \phi_{\max} = \max_i \{\phi_i\} \tag{15a}$$

$$\psi_{\min} = \min_i \{\psi_i\} \quad \text{and} \quad \psi_{\max} = \max_i \{\psi_i\} \tag{15b}$$

$$\theta_{\min} = \min_i \{\theta_i\} \quad \text{and} \quad \theta_{\max} = \max_i \{\theta_i\} \tag{15c}$$

Again the above analysis is over-simplified and we have done the same two mistakes as we did for the hinge joint type. However, note that the ZYZ convention is the savior. It means that each of the Euler parameters can be analyzed independently of each other.



(a) Intervals appear non-contiguous

(b) Intervals appear contiguous

Figure 5: Contiguous intervals may appear non-contiguous if one analyze the numerical values of the angles. However, if one changes the interval on which angles are represented the intervals will appear contiguous.

2.3 The Agony of Rotation Angles

Human motion is piecewise continuous. Thus it is a fair assumption that the range of motion can be considered as being a contiguous interval. However, when dealing with rotational motion it is not straightforward to obtain the contiguous interval. This is illustrated in Figure 5. The figure suggests that one solution may be to change an interval range $[-\pi..\pi]$ into the range $[0..2\pi]$ or vice versa. However, not knowing which case we are dealing with makes it difficult to decide if the interval range should be changed. Thus we will consider a different approach.

Assume the θ-values are sorted in ascending order, $\theta_1 < \theta_2 < \cdots < \theta_N$. Now we can compute the angle difference between two consecutive angle values i and $j = (i \oplus 1)$ in counter-clock-wise direction, here $\oplus$ is defined as addition modulus N,

$$\Delta\theta_i = \begin{cases} \theta_j - \theta_i & ; \theta_j > \theta_i \\ \theta_j + 2\pi - \theta_i & ; \text{otherwise} \end{cases}. \tag{16}$$

The largest angle difference,

$$\Delta\theta_m = \max_i \{\Delta\theta_i\}, \tag{17}$$

will contain the angle values outside the contiguous interval. Figure 6 illustrates the method. Having found k we now have

$$\theta_{\min} = \theta_{m \oplus 1} \tag{18a}$$

$$\theta_{\max} = \theta_m \tag{18b}$$

We test if

$$\theta_{\min} < \theta_{\max} \tag{19}$$

if the test fails then we keep adding 2π to $\theta_{\max}$ until the test passes.

2.4 Bind-pose is Infeasible

Note that the joint parameter value of the bind-pose can be computed correctly as well. One can obtain the bind-pose angle, θ as we described earlier in Section 2.2.

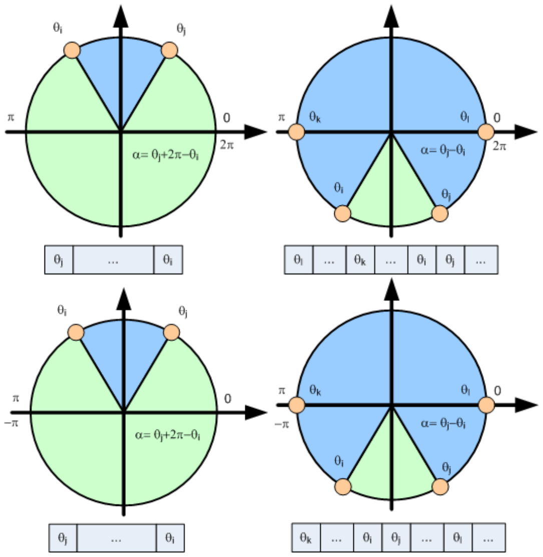


Figure 6: Angle intervals (blue) shown on the unit-circle. Sorting angle values in ascending order makes it easy to search for the largest difference between two consecutive angles in a counter-clock-wise manner. The largest difference will be the empty gap (green) bounding the interval we are searching for.

However, one needs to make sure that θ is feasible with respect to the minimum and maximum values. Thus we test

$$\theta_{\text{min}} \leq \theta \tag{20}$$

If not we add 2π to θ until the test succeeds. Finally we test if

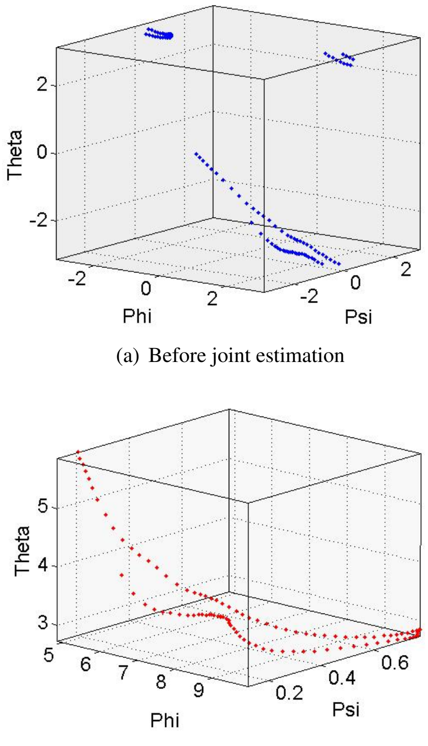
$$\theta \leq \theta_{\text{max}} \tag{21}$$

If this test fails our bind-pose is not feasible with respect to the motion capture data we have analyzed.

This may appear strange but imagine we have a human figure, and we are analyzing a running motion. During running the shoulder complex will never lift the arms above the head even though this would be a legal motion for a shoulder. In fact it is likely that in this case our analysis will suggest a hinge joint type for the shoulders and set joint limits such that the arms are never lifted above horizontal level. The bind-pose is often a pose similar to Leonardo Da-vincis "Vitruvian man" where the arms are kept horizontally.

When failure of the test $\theta \leq \theta_{\text{max}}$ occurs then the next problem is to decide how to solve the problem? We propose to pick one of the poses from the motion capture sequence and use this as the initialization pose. Another obvious choice is to use the mean angle for the initialization,

$$\theta = \frac{\theta_{\text{max}} + \theta_{\text{min}}}{2}. \tag{22}$$



(a) Before joint estimation

(b) After joint estimation

Figure 7: Illustration of 3 dimensional distribution of Euler-parameters before and after the contiguous angle analysis has been performed. The chosen example joint is a ball joint, the left shoulder joint.

This will definitely be a feasible value. Alternatively we can compute the mean point

$$\vec{m} = \begin{bmatrix} m_x \\ m_y \end{bmatrix} = \frac{1}{N} \begin{bmatrix} \sum_{i=1}^{N} \cos(\theta_i) \\ \sum_{i=1}^{N} \sin(\theta_i) \end{bmatrix} \tag{23}$$

if $\vec{m}$ is zero then we must give up since that would indicate that the motion is unlimited or not sampled sufficiently. Otherwise we convert the mean point to an angle

$$\theta_{\text{mean}} = \ \text{atan}_2 \left(m_y, m_x \right). \tag{24}$$

The mean angle would be feasible and would in a sense yield the most likely pose.

3 RESULTS

The system was tested using a number of different motion capture animations of gymnastics exercises. Some of the exercises are shown in figure 8. Some of these yielded restrictive bounds while some gave more general bounds. This depended of the local motion of the individual joints. The motions where chosen so all joint where moved in at least some of the motions.

In Figure 7 we have shown the result of performing a contiguous angle analysis on a shoulder joint. For this example 100 key-frame samples were used for a motion of 2 seconds of duration. Observe that after the estimation the motion trace is contiguous and not disconnected.

Our method is intended as a pre-processing tool during modeling of characters and hence no real-time performance requirements are needed to be fulfilled. Thus we only need to consider a performance good enough for not stalling the animation tool used by an artist.

For 100 motion samples the joint type discrimination and angle analysis is computationally fast enough not to be noticed by the end-user. For a 30 bone character as shown in Figure 8 the analysis takes less than 50 ms and 20 ms on average on a modest laptop computer (Pentium®core duo T5500 1.66 GHz).

If the number of samples are too few it may become difficult to determine a contiguous interval. Also the tightness of the limits may be too tight if the motion samples are not taken from extreme poses.

The joint estimation is data driven and the model is local. Surely bad motions can be picked yielding over-restricted motion ranges. However, due to the local modeling the overall motion-type is insignificant to the results in the individual joints. The motion samples are not used to perform a motion reconstruction. Therefore we only need samples close enough to the minimim and maximum bounds and a few in between samples to make out which parts of the angle intervals corresponds to the contiguous part of the motion.

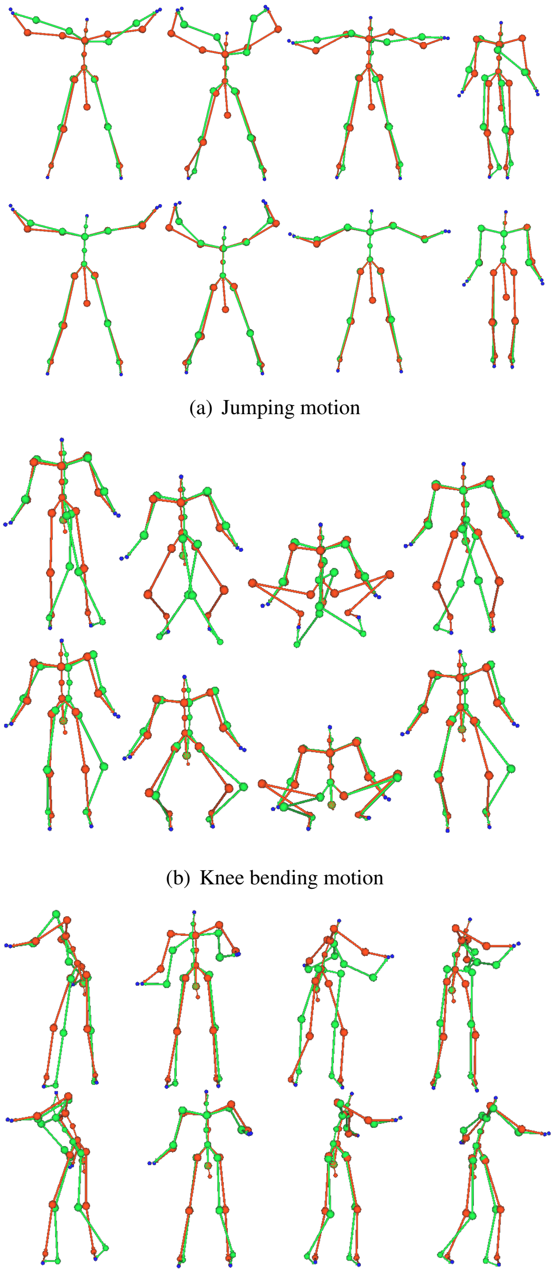
The system supports arbitrary tight sampling by interpolation of the given motion capture values. Thus to few samples are rarely a problem. A minimum of 4-6 samples are necessary to make this interpolation feasible though.

Figure 8 and the supplementary video shows the quality achieved by estimating joint limits on several different motions using only 100 samples for the analysis. The usual way of handling joint limits in industry is to design joint limits for specific animations, thus joint limits which are much more restrictive than real human joint limits are obtained. The reason for this is that the inherent redundancy of human motion makes it difficult to control animations using general joint limits. Our system makes it possible for an animator to make the joint limits as general or as specific as he or she sees fit, Based on the generality of the chosen reference animations. Thus the animator is given explicit control, without losing generality.

Figure 9 shows motion samples of a shoulder joint for three different motions. As illustrated the 100 samples appear to capture the overall motion of the shoulder joint. Thus in practice we find this number of samples to be sufficient.

4 DISCUSSION

Human motion is piecewise continuous and non-linear in position and velocity. Thus it is questionable whether an eigen-value analysis is useful. A more advanced



(a) Jumping motion

(b) Knee bending motion

(c) Arm swinging motion

Figure 8: Examples of the impact of joint limits. The top rows of each motion example shows a sequence without joint limits, the bottom rows shows the same sequence, using joint limits calculated with our method. The red skeleton is the motion capture reference while the green is the inverse kinematics solved. Observe that the inverse kinematic solution resembles the motion capture motion better using our joint estimation method.

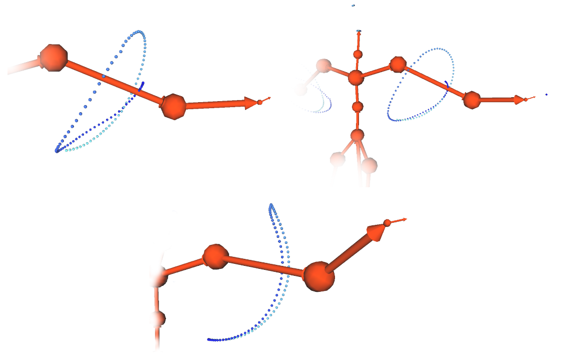


Figure 9: We use 100 motion samples for discriminating joint types and finding contiguous angle intervals for motions of roughly 2 seconds of duration. The resulting samples of three different shoulder motions are shown.

analysis such as principle geodesics analysis [6] should be able to deal inherently with the fact that rotational angles lies on a rather complex hypersphere. We have reformulated the dimensionality problem to fit an Euclidean space.

Our problem is that of determining the dimension of a sub-set of a motion space for a single joint. The actual motion analysis we perform to describe the physical boundaries of motion is independent of the eigenvalue analysis. We could perhaps limit the dimensionality even more by only considering the dimensions that account for say 95% of the variation [3]. This would perhaps be better than our approach of using a small threshold on the absolute value of the eigen-values. The dimension analysis is simplified by having a low dimensional explicit parameterized model which we seek to fit motion capture data with. As such our problem can be understood as a regression of real world data onto a much simpler and more primitive model. In general the dimension analysis seems reasonable.

A weak point of our method is that we only consider a local analysis of each joint independently of other joints. In fact the motion analysis is further localized to deal with each joint parameter independently of each other. This is a crude simplification. For more accurate modeling of human motion one should consider a global analysis. However, this is not warranted in case of inverse kinematics skeletons for interactive applications.

In future work it could be interesting to learn the manifold of the feasible motion space including the dependencies between joint parameters and augment the simple inverse kinematic skeleton with a more complex joint limit model. For instance by tessellation of the motion samples and form a boundary representation of the configuration space. They could be used to infer linear approximations to joint limits for a current iterate of the joint parameters.

ACKNOWLEDGMENTS

We wish to thank Kim Steenstrup Pedersen and Jon Sporring for valued input and discussions on our work. This research was done as part of the HUMIM project under the eScience Centre, Faculty of Science, University of Copenhagen. The data used in this project was obtained from mocap.cs.cmu.edu. The database was created with funding from NSF EIA-0196217.

REFERENCES

[1] Jaejin Choi. *Maya Character Animation*. Sybex Books, December 2002.

[2] Yancey Clinton. *Game Character Modeling and Animation with 3ds Max*. Focal Press, October 2007.

[3] T. F. Cootes, C. J. Taylor, D. H. Cooper, and J. Graham. Active shape models—their training and application. *Comput. Vis. Image Underst.*, 61(1):38–59, 1995.

[4] Z. Dvir and N. Berme. The shoulder complex in elevation of the arm: A mechanism approach. *Journal of Biomechanics*, 11(5):219–225, 1978.

[5] Martin Fêdor. Application of inverse kinematics for skeleton manipulation in real-time. In *SCCG '03: Proceedings of the 19th spring conference on Computer graphics*, pages 203–212, New York, NY, USA, 2003. ACM Press.

[6] Thomas P. Fletcher, Conglin Lu, and Sarang Joshi. Statistics of shape via principal component analysis on lie groups. In *In Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 95–101, Los Alamitos, CA, USA, 2003.

[7] Lorna Herda, Raquel Urtasun, Andrew Hanson, and Pascal Fua. Automatic determination of shoulder joint limits using quaternion field boundaries. *International Journal of Robotics Research*, 22(6):419–434, June 2003.

[8] O'Brien J., Bodenheimer R., Brostow G., and Hodgins J. Automatic joint parameter estimation from magnetic motion capture data. *Graphics Interface*, pages 53–60, 2000.

[9] Adam Kirk, James F. O'Brien, and David A. Forsyth. Skeletal parameter estimation from optical motion capture data. In *SIGGRAPH '04: ACM SIGGRAPH 2004 Sketches*, page 29, New York, NY, USA, 2004. ACM.

[10] Lucas Kovar and Michael Gleicher. Flexible automatic motion blending with registration curves. In *SCA '03: Proceedings of the 2003 ACM SIGGRAPH/Eurographics symposium on Computer animation*, pages 214–224, Aire-la-Ville, Switzerland, Switzerland, 2003. Eurographics Association.

[11] Sung-Hee Lee and Demetri Terzopoulos. Spline joints for multibody dynamics. In *SIGGRAPH '08: ACM SIGGRAPH 2008 papers*, pages 1–8, New York, NY, USA, 2008. ACM.

[12] Walter Maurel and Daniel Thalmann. Human shoulder modeling including scapulo-thoracic constraint and joint sinus cones. *Computers & Graphics*, 24(2):203–218, 2000.

[13] Tony Mullen. *Introducing Character Animation with Blender*. John Wiley & Sons, February 2007.

[14] Wei Shao and Victor Ng-Thow-Hing. A general joint component framework for realistic articulation in human characters. In *I3D '03: Proceedings of the 2003 symposium on Interactive 3D graphics*, pages 11–18, New York, NY, USA, 2003. ACM.

[15] Jane Wilhelms and Allen Van Gelder. Fast and easy reach-cone joint limits. *J. Graph. Tools*, 6(2):27–41, 2001.

[16] Jianmin Zhao and Norman I. Badler. Inverse kinematics positioning using nonlinear programming for highly articulated figures. *ACM Trans. Graph.*, 13(4):313–336, 1994.

A OBTAINING ZYZ EULER ANGLES

Extracting ZYZ Euler angles robustly and directly from a unit-quaternion is not trivial. Of course one can convert to another representation such as rotation matrices. Below we outline our approach which is based completely on quaternions and explicitly takes Gimbal lock into account.

Here ϕ, ψ and θ defines the rotation given by the unit-quaternion, Q, such that

$$Q \equiv R_z(\phi) R_y(\psi) R_z(\theta); \tag{25}$$

Our task is to find ϕ, ψ, and θ given Q. We exploit the following idea below to reduce the problem. We use a clever test-vector, $\vec{k} = \begin{bmatrix} 0 & 0 & 1 \end{bmatrix}^T$ and try to rotate this vector with the given rotation. That is

$$Q\vec{k}Q^* \equiv R_z(\phi) R_y(\psi) R_z(\theta)\vec{k} = R_z(\phi) R_y(\psi)\vec{k}, \tag{26}$$

where Q^* is the conjugated quation of Q. Denoting $Q\vec{k}Q^* = \vec{u}$, a unit vector, we no longer need to worry about θ. Now we must have

$$\vec{u} = \begin{bmatrix} u_x \\ u_y \\ u_z \end{bmatrix} = R_z(\phi) R_y(\psi) \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix} = \begin{bmatrix} \cos(\phi)\sin(\psi) \\ \sin(\phi)\sin(\psi) \\ \cos(\psi) \end{bmatrix} \tag{27}$$

From the z-component we solve

$$\psi = \cos^{-1}(u_z) \tag{28}$$

This forces ψ to always be in the interval $[0..\pi]$. We know that $\sin(\psi)$ is always going to be positive, which mean that we can divide the second equation by the first equation and obtain

$$\frac{\sin(\phi)}{\cos(\phi)} = \tan(\phi) = \frac{u_y}{u_x} \tag{29}$$

From this we have

$$\phi = \ \text{atan}_2(u_y, u_x) \tag{30}$$

That means that ϕ will always be in the interval $[-\pi..\pi]$. Observe if ψ is zero then u_y and u_x is both zero and our approach will always compute ϕ to be the value zero. The case is actually worse than it seems. Because with $\psi = 0$ the ZYZ Euler angles are in a Gimbal lock where the two Z-axis transformations are completely aligned. Thus we test for Gimbal lock if $\psi < \varepsilon$ where ε is a small user selected threshold. In case of Gimbal lock we use a unit test-vector along the x-axis

$$\vec{w} = \begin{bmatrix} w_x \\ w_y \\ w_z \end{bmatrix} = Q \begin{bmatrix} 1 \\ 0 \\ 0 \end{bmatrix} Q^* \tag{31}$$

and compute

$$\phi = \ \text{atan}_2\left(w_y, w_x\right) \tag{32}$$

and set $\psi = \theta = 0$. We now know how to compute ϕ and ψ even in case of a Gimbal lock. So now we can compute

$$Q_{zy} \equiv R_z(\phi) R_y(\psi) \tag{33}$$

and from this we know

$$Q = Q_{zy} Q_z(\theta) \tag{34}$$

so

$$Q_{zy}^* Q = Q_z(\theta) = \begin{bmatrix} \cos(\frac{\theta}{2}) \\ \sin(\frac{\theta}{2})\vec{k} \end{bmatrix} \tag{35}$$

and we get θ by

$$\theta = 2 \ \text{atan}_2\left(\sin\left(\frac{\theta}{2}\right), \cos\left(\frac{\theta}{2}\right)\right) \tag{36}$$