# KINEMATICS BASED MOTION COMPRESSION FOR HUMAN FIGURE ANIMATION

*Shiyu Li, Masahiro Okuda, Shin-ichi Takahashi*

Faculty of Environmental Engineering, The University of Kitakyushu, Japan

### ABSTRACT

This paper presents a technique for compression of motion data using an inverse kinematics algorithm. In a motion chain, the movement of each joint is represented by a series of vector signals in 3D space. In general, specific types of joints such as end effectors often require higher precision than other general types of joints in, for example, CG animation and robot manipulation. When compressing these motion data, the distortion of parent joint coming from quantization in turn affects its child joint and is accumulated to the end effector. To address this problem and control the movement of the whole body, we propose a prediction method based on the inverse kinematics. Our method achieves efficient compression with a high compression rate and high quality of the motion data. By comparing with a conventional prediction we demonstrate the advantage of ours with some example.

## 1. INTRODUCTION

Motion capture systems have been widely used in CG amusement and human motion analysis such as games and athlete training. To reuse human motion capture data, users need a motion library to store existing motion data with human like character. Large motion databases do not accept the uncompressed forms, since the motion data are huge. Moreover, motion data retrieval and transmission often require compactly coded motion [3].

The previous methods for animation compression focus on the reduction of the number of motion samples [5][2] or the size of database by utilizing wavelet technique to compress each sample [1][6]. In [1], Ahmed et al. use the discrete wavelet transform to achieve the motion compression. For the motion compression, one should take into account some special joints such as end effectors which often require higher compressing precision than other general types of joints due to motion characters. The previous methods do not handle this. In [6], we propose a constraint based compression algorithm for motion data with special characteristics which can be indicated by motion behavior or specified by users. This method combines wavelet transform and forward kinematics to achieve an efficient motion compression by using adaptive quantization to establish the optimal positions of joints.

However, according to the forward kinematics, in a motion chain, the distortion of a joint which comes from any quantization scheme introduces the warp of the position to its child joint. The distortion of this child joint in turn affects its grand child joint, and so on. The warp may accumulate to the end-effector which is usually treated as the most important joint for motion feature. To reduce the propagation of the warp to the end effector, we need to minimize the error between actual position and compressed result in lower level of hierarchy. Unfortunately the forward kinematics can not solve the problem perfectly.

On the other hand, to achieve the high compression rate, the number of data and the magnitude of the values are also important issues. A linear prediction has been widely and successfully applied in compression of time series data. If we can predict every next frame, we only need to save the first frame and difference between real value and its prediction. The better the predictions are, the more common corrections we can get and the more bits we can save. Considering the merit of predicting technique we prefer to utilize this technique to achieve high coding efficiency.

To control the movement of the whole body, it is common to use the inverse kinematics. It is presumed that specified joints, called the end effectors are assigned in target positions from preceding positions. By position changes of the end effectors one may get variations of the motion of the entire body using the inverse kinematics algorithm. Therefore for most joints, when recovering motion in decoder, we only require a series of small modifications to the corresponding values. Obviously, the inverse kinematics based approach can solve the above problems and achieve efficient compression of the motion data with a high compression rate and high quality of the decoded motion.

As the motion sample data, the BVH format [7] is adopted. Furthermore, we apply a reduced format of two rotation angles to be compressed instead of Euler angles, which is translated from the BVH format and gotten in preprocessing of motion data. The reduced format can improve the compression efficiency further. Moreover, one can give a closed form of the Jacobian matrix, which saves computation time. It will be described in the following section.

The remainder of the paper is organized as follows. After a brief description about the transformation from the general motion format to the two-angle format in section 2, we explain the inverse kinematics based compression algorithm in section 3. The motion compression procedure with a predicting technique is assigned in section 4. And in section 5, we demonstrate the advantage of our approach with some example motions. Final section is conclusion.

## 2. PRELIMINARY

### 2.1 General Format

In CG application, a human figure is modeled by a hierarchical chain, in which connections between two neighboring joints are rigid, for example, the joints of a shoulder and an elbow move, but the distance between them is not changed. In this framework, the motion data in 3D space, $x$, $y$, $z$, are converted to a series of rotations. A motion chain which is hierarchically constructed by some linked joints has one end that is free to move, which is called an end effector. The other end of the chain is fixed or defined as an origin of coordinates, and is called a terminator. See Fig.1. The roots may have multiple trees and end effectors.

0-7803-8874-7/05/$20.00 ©2005 IEEE

Kinematics based motion data processing is to handle the motion chains, such as trunk, upper limbs and lower limbs.

In the motion chain, to calculate the position of a joint we need to create a rigid transformation matrix by local translation and rotation information. A rotation matrix $R$ is composed of 3 Euler rotation matrices with respect to $X$, $Y$, $Z$ axes [4]. Suppose a rotation order is $YXZ$, by concatenating the Euler rotation matrices, we can get $R=R_z*R_x*R_y$. By applying a matrix $T$ which is a homogeneous matrix to represent both the translation and the rotation by one common equation, the position of a joint in global coordinate $p_G$ can be described by $p_G=T*p_L$, where $p_G=[1\ p_X\ p_Y\ p_Z]^T$, $p_L$ is the position of this joint in local coordinate and $p_L=[1\ p_x\ p_y\ p_z]^T$, $T=\begin{bmatrix}1 & \vdots & 0\\ \cdots & & \cdots \\ l & \vdots & R\end{bmatrix}$, and $l$ is a translation vector$[1\ l_x\ l_y\ l_z]^T$. Once the local transformation of a joint is created, it will be concatenated with the transformation of its parent, then its grand parent, and so on. The position of this joint in world coordinate can be obtained by $p_{world}=T_{root}*T_{grandparent}*T_{parent}*T_{child}*p_{child}$.

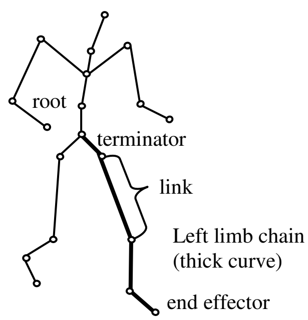



Fig. 1 Human Figure

### 2.2 Two rotation angle format

Generally the motion data is described by series of position $x$, $y$, $z$ in three dimensional space, Euler angles, roll-pitch-yaw, or etc. In a sense of compression efficiency, these formats are verbose, since we assume the motion chains are expressed by rigid transform and joint positions are described by two values except for the end effectors. To remove the verbosity, we first convert the data to a two-angle format.

Suppose the length of a link connecting a child joint and a parent joint is $r$, which is gotten by the position offset of the child joint $p_{child}=[\ p_{xchild}\ p_{ychild}\ p_{zchild}]^T$.

After a rotation $R_{ZXY}$ as order $Y$, $X$, $Z$, the new position of this child joint in its parent coordinate is $p_{parent}=R_{ZXY}*p_{child}$, where $p_{parent}=[\ p_{Xparent}\ p_{Yparent}\ p_{Zparent}]^T$.

To represent the $p_{parent}$ by two angles:

$$\begin{cases} p_{Xparent} = r*\sin\phi*\cos\varphi \\ p_{Yparent} = r*\sin\phi*\sin\varphi \\ p_{Zparent} = r*\cos\phi \end{cases} \tag{1}$$

where $\phi$ and $\varphi$ can substitute for three Euler angles to be compressed.

## 3. INVERSE KINEMATICS BASED ALGORITHM

According to the forward kinematics, in a motion chain, the transformation of a parent joint causes a change of its child joint position. The change of this child joint in turn affects its grand child joint, and so on. Finally the changes are accumulated to the end effector. See Fig. 2(a). Motion is inherited down the hierarchy from the parents to the children. For simplicity, we discuss two-dimensional case. And the position of the end effector $p(x, y)$ in two dimensional can be determined

$$p\ (x, y) = f(\boldsymbol{\theta}) \tag{2}$$

where $\boldsymbol{\theta}$ is a set of rotation angles at each joint $(\theta_1, \theta_2, \theta_3, \ldots \theta_n)$.

In Inverse Kinematics (IK), motion is inherited up the hierarchy, from the extremities to the root (Fig. 2(b)). The role of the IK algorithm is to automatically work out how each joint in a chain should be transformed so that the end effector can reach the goal. To find the set of the changes of the joint angles which satisfy a given displacement of the position of the end effector, we need to solve

$$\boldsymbol{\theta} = f^{-1}(p(x, y)) \tag{3}$$

However, there are, in general, an infinite number of solutions of (3). To solve the high redundancy, Jacobian-based method is utilized. Therefore (2) is written in differential form:

$$\dot{p}\ (x, y) = J\dot{\boldsymbol{\theta}} \tag{4}$$

$J$ is Jacobian matrix and $J \equiv \partial f / \partial \boldsymbol{\theta}$.

Then equation (4) can be written by

$$\dot{\boldsymbol{\theta}} = J^{-1}\dot{p}(x, y) \tag{5}$$

(5) linearly relates the change of the end effector to the change of joint angle and is used as the basis in our prediction algorithm. When $\Delta p$, which is a displacement of end effector from previous position to current position, is given, the change of each joint can be determined.

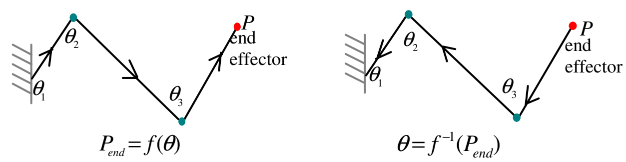



Fig. 2 (a) Forward Kinematics (b) Inverse Kinematics

Our IK algorithm consists of the following steps.

1. Calculate the increment of the position of the end effector from frame $i$-1, $p_{i-1}$, to frame $i$, $p_i$, $\Delta p = p_i - p_{i-1}$
2. Calculate Jacobian matrix

$$J\ (\phi_1, \varphi_1, \phi_2, \varphi_2, \phi_3, \varphi_3, \ldots, \phi_n, \varphi_n)$$

using the angles of last frame

$$(\phi_1, \varphi_1, \phi_2, \varphi_2, \phi_3, \varphi_3, \ldots, \phi_n, \varphi_n),$$

where $\phi_j$ and $\varphi_j$ are the two-angles in (1). Since

$$p_i\ (x, y, z) = [f_1(\phi, \varphi)\ \ f_2(\phi, \varphi)\ \ f_3(\phi)]^T,$$

where $f_1(\phi, \varphi) = \sum_{j=1}^{n} L_j * \sin(\phi_j)\cos(\varphi_j)$,

$$f_2(\phi, \varphi) = \sum_{j=1}^{n} L_j * \sin(\phi_j)\sin(\varphi_j),$$

$f_3(\phi) = \sum_{j=1}^{n} L_j * \cos(\phi_j)$ by equation (1) and $L_j$ is the length of link $j$. Then by (4), $\Delta p = J[\dot{\phi}_j\ \dot{\varphi}_j]$ and

$J = [\partial f_1(\phi,\varphi)/\partial\phi_j\ \ \partial f_1(\phi,\varphi)/\partial\varphi_j, \partial f_2(\phi,\varphi)/\partial\phi_j\ \ \partial f_2(\phi,\varphi)/\partial\varphi_j,\ \partial f_3(\phi)/\partial\phi_j\ 0]$,

where, $\partial f_1(\phi,\varphi)/\partial\phi_j = L_j\cos(\phi_j)\cos(\varphi_j)$,

$\partial f_1(\phi,\varphi)/\partial\varphi_j = -L_j\sin(\phi_j)\sin(\varphi_j)$,

$\partial f_2(\phi,\varphi)/\partial\phi_j = L_j\cos(\phi_j)\sin(\varphi_j)$,

$\partial f_2(\phi,\varphi)/\partial\varphi_j = L_j \sin(\phi_j)\cos(\varphi_j)$,

$\partial f_3(\phi)/\partial\phi_j = -L_j \sin(\phi_j)$

3. Get the pseudo inverse of $J$, $J^+ = J^T(JJ^T)^{-1}$, since $J$ is not always invertible [6]

4. Calculate the value in frame $i$ for $\phi_1$, $\varphi_1$, $\phi_2$, $\varphi_2$,… $\phi_n$, $\varphi_n$, in each chain. $\boldsymbol{\theta}$ is the vector of angles for each chain and is given by $\boldsymbol{\theta}_i=\boldsymbol{\theta}_{i-1}+J^+\Delta p$

In step2, if the general format, which is composed of rotation angles about $X$, $Y$, $Z$ axes, is applied, to calculate the Jacobian matrix, each item almost involves all the related angles.

While in our two-angle format, since $\phi$ and $\varphi$ are calculated by the product of all the related rotation matrices from current joint to root joint, using two-angle format, the Jacobian $J$ is given in a closed form, which saves computation time.

## 4. CONSTRAINT BASED COMPRESSION WITH PREDICTION TECHNIQUE

Considering motion characters, we have to assign more bits to some special joints such as the end effector than other general types of joints. An adaptive quantization approach preserves features of the motion greatly. To achieve this, the hierarchical stepsize for different joints can be implemented in quantization step.

Meanwhile, since the amount of motion data is considerable, high compression rate is needed. Prediction based techniques have been widely and successfully applied in compression of series of data. If we can predict every next frame, we only have to save the first frame and the difference between real value and predicting result. The better the predictions, the more common corrections we can get and the more bits we can save.

The aforementioned two points characterize our compression approach properly when comparing with previous works. Actually, there are no conventional algorithms specialize the constraints in joints, i.e., precise reconstruction of the end effectors, and achieve efficient compression rate simultaneously.

To predict every next frame, an intuitive method is to utilize the last frame directly. Taking the difference $D$ between the current and last frames may be one of the simplest method. By this method, we can decode current value $\theta_{current}$ in decoder using equation $\theta_{current} = \theta_{last}+D$. Improving compression rate only depends on stepsize since $D$ is fixed. We have to explore a better prediction method which can provide the data closer to $\theta_{current}$.

The inverse kinematics gives a solution exactly. In our compression method, an encoder calculates the differences of position of end effectors between two sequential frames. More bits are assigned to them to keep the precision of the end effectors in quantization. Then these differences will be sent to decoder. Next, both in encoder and decoder, using these differences and the angles in previous frame we can calculate the change of rotation angle in each joint by IK (inverse kinematics) algorithm approximately. Suppose the difference between the value predicted by IK and the value in last frame is $D'$ and $\theta_{IKpredict} = \theta_{last}+D'$. We need transfer $\delta = D\text{-}D'$ to the decoder which may reconstruct current value $\theta_{current}$ by $\theta_{current} = \theta_{IKpredict}+\delta$. Obviously, the closer the $\theta_{IKpredict}$ to $\theta_{current}$, the smaller the $\delta$ is. In some cases, $\theta_{IKpredict}$ almost equals to $\theta_{current}$, thereby $\delta$ tends to zero.

This IK algorithm based compression procedure with prediction technique is shown in Fig.3.

## 5. EXPERIMENTAL

In our experiment we adopt the BVH file format [7]. The BVH file has two parts, a header section which describes any number of skeleton hierarchies and the initial pose of the skeleton by translational offsets of children segments from their parent; and a data section which contains the position of the root joint and the rotation information of motion of all joints in each frame. In the BVH format, the motion is described by a series of the three rotation matrices with respect to $y$, $x$, $z$ axes. We convert them to the two curves representing the rotations by $\phi$ and $\varphi$ in formula (1), and then compress them.

To evaluate the efficiency, we calculate the error of position of joint $i$ between the original motion and the compressed one by equation:

$$E_i = \frac{1}{N_f}\sum_{j=1}^{N_f}\sqrt{\left\|P_{oj}^i - P_{cj}^i\right\|^2}$$

and the error of all joints in all frames by:

$$E = \frac{1}{N_j}\sum_{i=1}^{N_j} E_i$$

where $P^i_{oj}$ and $P^i_{cj}$ are 3D position of joint $i$ in frame $j$ of original and compressed motions respectively in world coordinate. And $N_f$ is the number of frames, while $N_j$ is the number of total joints.

To compare our IK based prediction (IKBP) method with simply prediction by last frame (SPBLF), we give the RD curves of walking motion and ballet motion in Fig.4. In this figure, as the increment of entropy, the change of error of our method is smaller than the simple predicting method. It demonstrates that our proposed algorithm can get the more common corrections and save more bits than the general algorithm.

We also show the different error of position of each joint in a limb chain corresponding to different entropy value in Table.1 to demonstrate the advantage of our approach.

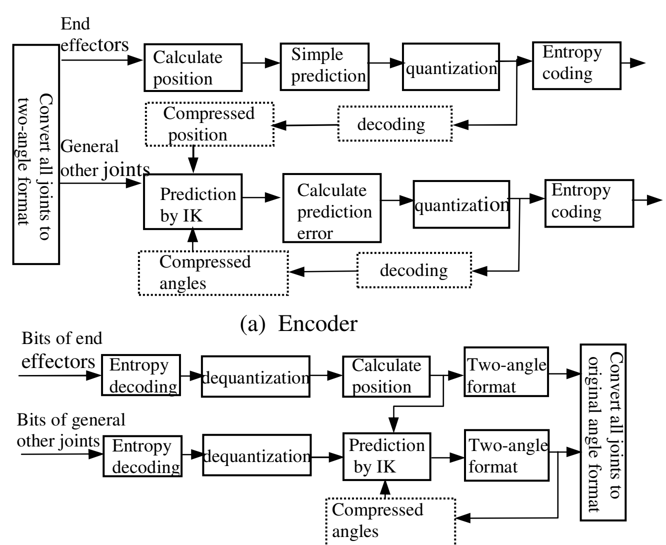



Fig. 3 Compression flow chart

The original data size of walking motion is 314kb. The number of joints is 23 and these joints belong to 5 motion chains respectively. The number of frames is 580 and the frame sampling rate is 0.00833s. Fig.5 shows series of samplings of the original walking motion and the decoded one correspondingly when entropy is 0.6679, where the serial numbers of the sampling frames are 1, 13, 23, 36, 46, 71, 83, 97, 109 and 128. These series of samplings present a period of the walking motion and it is difficult to discovery the difference between the original motion and the decoded one when entropy is larger than 1.1206. Another applied motion is the ballet motion with 141kb original size. There are 20 joints which belong to 5 motion chains respectively in ballet motion. The number of frames is 388 and the frame sampling rate is 0.04s.

## 6. CONCLUSTION

The compression of captured motion data is an important issue in motion storing, retrieval, editing and transmitting. For the motion compression, some specific types of joints such as end effectors often require higher precision than other general types of joints.

Inverse kinematics is a common approach to control the movement of the whole body. The position of end effector can be specified in a target position from preceding position. By position changes of the end effectors we may get variations of the motion of the entire body. The inverse kinematics, on the other hand, supports a prediction based compression. To predict motion in decoder, we only need to save the first frame and a series of small difference between real value and prediction gotten by the variations of the positions of the end effectors. Therefore, it can solve the problems of specific joints and achieve efficient compression of the motion data.

We applied the inverse kinematics based compression in our reduced two-angle format. Some experiment results of example motions demonstrate the advantage of our method compared with conventional method.

**Table1.** Error of Position of Each Joint in Limb Chain

| Method | Entropy | Error | | | |
|---|---|---|---|---|---|
| | | Root Joint | Child Joint | Grandchild Joint | End effector |
| IKBP | 0.6697 | 0.125 | 0.224 | 0.329 | 0.371 |
| SPBLF | 0.7269 | 1.645 | 2.717 | 6.468 | 6.338 |
| IKBP | 1.3413 | 0.076 | 0.139 | 0.204 | 0.229 |
| SPBLF | 1.3673 | 0.887 | 1.506 | 4.827 | 4.699 |
| IKBP | 2.3130 | 0.062 | 0.116 | 0.164 | 0.187 |
| SPBLF | 2.2147 | 1.270 | 1.821 | 1.085 | 1.470 |
| IKBP | 3.0346 | 0.038 | 0.069 | 0.102 | 0.117 |
| SPBLF | 3.1294 | 0.582 | 0.863 | 0.814 | 0.913 |
| IKBP | 3.9400 | 0.018 | 0.035 | 0.053 | 0.064 |
| SPBLF | 4.0860 | 0.542 | 0.666 | 0.637 | 0.609 |

## 7. REFERENCES

[1] A. Ahmed, A. Hilton and F.Mokhtarian "Adaptive compression of human animation data," *In Eurographics 2002,* Saarbrucken, Germany, 2002.

[2] Etou, H., Okada, Y. and Niijma, K. "Feature Preserving Motion Compression based on Hierarchical Curve Simplification", *Proc. of ICME2004*, Taipei, Taiwan, June 2004.

[3] Bruderlin, A., and Williams, L, "Motion signal processing," *Proceedings of SIGGRAPH 95. ComputerGraphics*, pp. 97—104, Aug. 1995.

[4] Vladimir M. Zatsiorsky, Kinematics of Human motion, Department of Kinesiology, The Pennsylvania State University, 1998.

[5] Lim, D. Thalmann , "Key-posture Extraction out of Human Motion Data by Curve Simplification", Proc. *EMBC2001, 23rd Annual International Conference of the IEEE Engineering in Medicine and Biology Society,* vol.2, pp.1167-1169, 2001.

[6] Shiyu Li, Masahiro Okuda, Shin-ichi Takahashi, "Hierarchical Human Motion Compression with Constraints on Frames", *47th MWSCAS*, Hiroshima, Japan, pp. I-253- I-256, Jul. 2004.

[7] M. Meredith and S.Maddock, "Motion capture file formats explained," Department of Computer Science, University of Sheffielf.

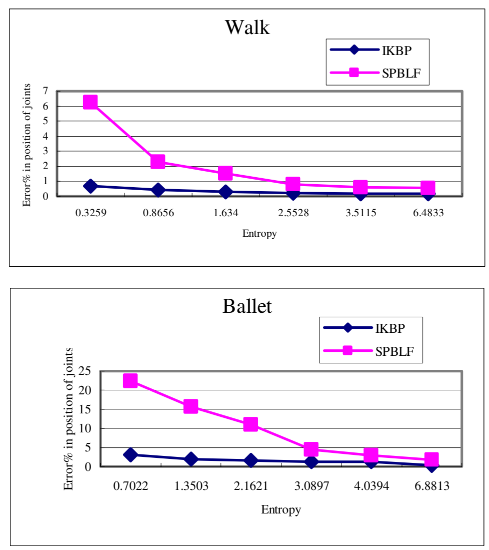
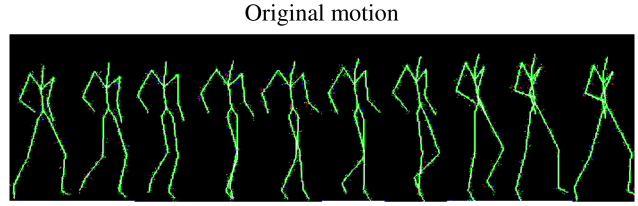



Fig. 4 RD curve

Original motion



Decoded motion

Fig.5 Series of samplings of walking motion