

Watermarking Motion Data

Shuntaro Yamazaki

Digital Human Research Center

The National Institute of Advanced Industrial Science and Technology (AIST)

shun-yamazaki@aist.go.jp

<http://staff.aist.go.jp/shun-yamazaki/>

Abstract

This paper presents a method for a robust motion data watermarking mainly related to the human motion data. Our method can deal with the motion represented by joint angles or marker positions of an articulated link model. Motion data are usually supposed to undergo through considerable signal processing, which makes it difficult for the embedded watermark to survive adversary attacks. Moreover, the existing watermarking techniques lack the consideration of a major problem like its disappearance in the computer graphics (CG) animation.

Aiming to make the watermarking robust, we insert a watermark into the original motion data in the frequency domain, distributing it redundantly across different components of the motion data structure. We also propose a method of finding the watermark embedded in a set of motion data in order the copyright violation can be detected not only in their original format but also in a CG animation sequence which is generated using the watermarked motion data. Certain experimental results are presented for various motion capture data to show that the proposed watermarking method is imperceptible, while it is robust with respect to various types of possible attacks.

1 Introduction

Thanks to the rapid progress in motion capture (MoCap) technology [4], now is relatively easier to deal with this type of data as a basic component in a variety of digital media authoring. Furthermore, the necessity of watermarking this data type, stems from their wide use in the recent computer graphics (CG) related entertainment industry. The motion data (hereafter referred as motion) constitute a digital form of input for the abstract information, concerning everyday human activities, and provide the basis of a further digital processing. As the motion acquisition process including MoCap system requires an expensive setup and considerable work, the protection of the copyrighted data becomes an important issue.

We propose a method of robust motion watermarking¹

¹In this paper, we use the term of *robust* watermarking in the mean-

with the aim of the copyright protection of the obtained motion data. Watermarking is the process of slightly modifying the original data to hide into their structure a predefined message related to the contents. We extend the scope of this technique to motion data. Though in this paper we have applied this method only to human motion data for simplicity, it can easily be generalized to a broader scope.

The difficulties in motion watermarking is that motion data usually undergoes through a considerable signal processing, the latest including different free-form deformations, during the authoring process of multimedia contents. There are few chances for the embedded watermark to survive adversary's attacks. Moreover, the existing watermarking techniques lack the consideration of a major problem like its disappearance in the rendered CG images.

2 Related Work

2.1 Watermarking

The watermarking techniques have received increasing attention from both academic and industrial communities since the late 1990s. Early research was mainly focused on the techniques of so-called classical multimedia format such as images, sounds, and movies. The scope of watermarking has already spread over various types of digital contents, including texts, line drawings, 3D shape models, executable codes, and integrated circuits. This can be in details observed in different watermarking related literature [7, 16].

Watermarking is the method of associating some additional information within the original digital contents. To accomplishing this goal, there can be many ways including different trivial methods as inserting a message in the header of a digital file, etc. The advantage of watermarking over other methods resides with its imperceptibility and independence with respect to data representation. Owing to this property, watermarking has been used in such fields as proof of ownership (copyright protection), owner identi-

ing that the watermark is designed to survive the adversary's removal attempts, while some researchers refer such a watermark as a *secure* one [7].

cation, broadcast monitoring, transaction tracking, authentication and device control.

Our watermarking method is designed for proving ownership of human motion data. To be effective on ownership claims cases, the watermark must resist any possible attempt by an adversary to eliminate it. Such watermarks are referred to as *robust* watermarks, while *fragile* watermark is designed to be lost through any modification process and it mainly serves for proving the authentication of obtained original data. In order to maximize its robustness against different types of attacks, we adopt the scheme of *informed* watermarking in which a detector requires access to the original digital data [7], although *blind* watermarking advantage is that it can detect a watermark without gaining access to the original data.

One of the most successful methods of robust informed watermarking for digital signal is the spread spectrum approach proposed by Cox et al [6]. They inserted a set of pre-defined random numbers \mathbf{w} , which is regarded as a watermark, to the perceptually significant parts of the original data. The significance of the data is determined according to the amplitude of the frequency component. The spread spectrum approach has two characteristics that are important to robust watermarking. First, the watermark embedded into the low frequency components is difficult to remove without loss of data quality, since the coarse structure of the data is often essential. Second, that fact that a watermark can be dispersed over a large number of frequency components provides robustness to such signal distortions as band-pass filtering and additive random noise.

In the spread spectrum watermarking, the original data M is first transformed into the frequency domain by discrete cosine transformation (DCT). The largest coefficients except for DC component are then slightly modulated according to the watermark. The partially modified data is then transformed inversely to the original domain, which yields a watermarked data M' . We can distribute M' securely, while M and \mathbf{w} must be kept secret in a safe place. In the detection of the watermark, data which we suspect to have been attacked, are demodulated to extract a possible watermark \mathbf{w}' . The presence of watermark is claimed based on the statistical correlation between \mathbf{w} and \mathbf{w}' .

We developed our watermarking system based on the spread spectrum approach described above. The detail is explained in Section 4.

2.2 Motion Data

Motion data are given as time series of the parameters which describe the posture of human body represented in an articulated link structure. Motion data can be acquired in several ways. First, designers can create them manually or semi-automatically by using commercial tools for motion editing. Another second alternative are the MoCap systems which can be used to capture the motion of a real actor. Third, some techniques such as dynamics simula-

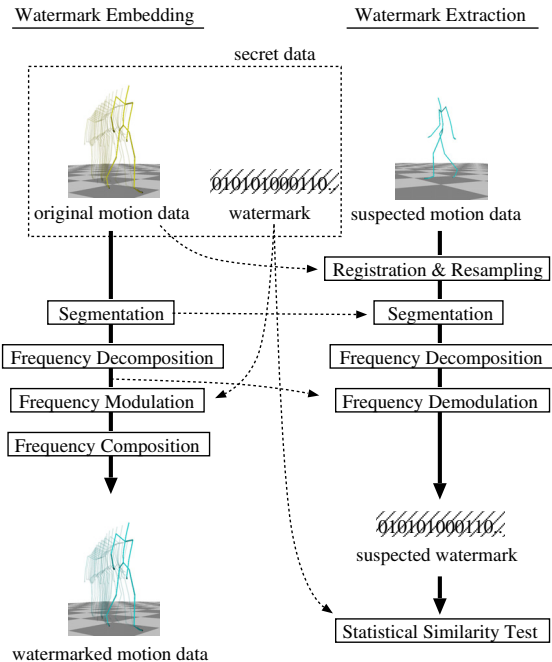


Figure 1: Overview of watermarking process

tion [19] or example-based motion synthesis [5, 18] can be used to generate motion data when combined with the data obtained with the other methods mentioned above.

One commonly used way of describing human motion is angular representation, in which the lengths of links are assumed to be fixed and the posture is defined by the 3D angles of joints. Another representation is the time-series of 3D positions. When the motion is captured from the images of an actor by tracking markers, a set of 3D positions is obtained. Our algorithm of watermarking can deal with both angle and position based motions. In this paper we represent motion as a time-dependent posture

$$M(t) = \{q_1(t), \dots, q_n(t)\} \quad (1)$$

where $q_i(t)$ are the time-series of either joint angles or marker positions. We suppose $q_i(t)$ are independent. n is the number of the independent parameters.

3 Process Overview

The overview of our watermarking process is illustrated in Figure 1.

Given an *original motion data* (or *reference motion data*) M to which we want to insert a watermark, a *segmentation* π and a *watermark* \mathbf{w} are first generated from M . Then the watermarked motion M' is obtained by embedding \mathbf{w} to M . The overview of watermark embedding process is as follows. The precise definitions of the symbols are described in Section 4.

1. **Watermark Generation:**

\mathbf{w} is generated using M to ensure that the watermarking is irreversible without gaining access to M .

2. **Motion Segmentation:**

M is temporally divided into a set of partial motions $\{M_i\}$ to increase resilience against partial deformation attack. This segmentation is referred to as π .

3. **Frequency Decomposition:**

Each 1D temporal data sequence m_i in M_i is decomposed to $\{f\}$ in frequency domain.

4. **Frequency Modulation:**

\mathbf{w} is embedded to the motion by modulating each $\{f\}$ to $\{f'\}$.

5. **Frequency Composition:**

$\{f'\}$ is transformed to a watermarked data sequence m'_i by frequency composition. Then $\{m'_i\}$ are combined into a watermarked motion M'

Once M' is generated, it can be distributed securely, while M , \mathbf{w} and π must be kept secret in a separate and safe place.

Given a *suspected motion data* M^* which we suspect to have been generated from M' without the author's permission, the copyright violation can be detected by proving the presence of watermark in M^* as follows. The precise definitions of the symbols are described in Section 5.

1. **Registration:**

M^* is transformed to the location of M in spatiotemporal space.

2. **Resampling:**

M^* is resampled so that M^* and M' has a one-to-one correspondence in the overlapping region.

3. **Motion Segmentation:**

M^* is temporally divided into partial motions $\{M_i^*\}$ according to π .

4. **Frequency Decomposition:**

Each 1D temporal data sequence m_i^* in M_i^* is decomposed to $\{f^*\}$ in frequency domain.

5. **Frequency Demodulation:**

A possible watermark \mathbf{w}^* is extracted by demodulating $\{f^*\}$ using $\{f\}$.

6. **Watermark Detection:**

The ownership of M^* is claimed by proving the similarity between \mathbf{w} and \mathbf{w}^* .

In our current implementation, we used well-known discrete cosine transformation (DCT) for the method of frequency composition and decomposition required in above watermarking process.

4 Embedding Watermarks

Watermark Generation

A mark vector $\mathbf{w} = \{w_1, \dots, w_m\}$ is generated by repeatedly sampling a Gaussian distribution $N(0, 1)$. We used the Mersenne Twister method [13] as pseudo-random number generator (PRNG) in our current implementation. In order to prevent false ownership claiming mentioned by Craver et al. [8], a seed passed to the PRNG is generated from the original data M so that the watermarking process is irreversible without gaining access to M . We passed M to a cryptographic hash function SHA-1 [17] to generate the seed.

Adelsbach et al. [1] mentioned that most of proposed irreversible methods, including Craver's scheme we used, do not come with a satisfactory proof of security. If further security is needed, the protocol proposed by Li and Chang [12] can be used to generate a watermark, although it requires a secret key to initialize a cryptographically secure pseudo-random number generator (CSPRNG).

Motion Segmentation

The motion data is a set of values defined in both spatial and temporal space, there are two approaches to embedding watermark to a given motion data set.

A spatial watermarking embeds a mark to each posture frame, that is, each posture in the motion segment. A single frame in a motion data is a graph structure with known topology in 3D space. Regarded the posture frame as a 3D mesh, theoretically, we can achieve motion watermarking by applying the existing method of mesh watermarking [14] to each posture in the motion. In practice, however, this scheme is mostly infeasible for several reasons. First, the number of joint position data in one frame is too small for imperceptible watermarking. Moreover, per-frame modification of postures degrades the temporal smoothness to which human visual system is known to be sensitive.

Accordingly, we adopt a temporal watermarking in which a mark is embedded independently into each time series q_i . This is based on the following observations. First the trajectory of $q_i(t)$ in spatial domain forms a continuous curve (Figure 2), and some trajectories are likely unaffected when the body form is partially altered. Second, the number of frames in a single motion is often more than 1000, which is sufficient to apply frequency analysis. We deal with each free parameter of each joint q_i independently, in aiming to increase the resilience against modification such as body form alternation.

Before embedding watermark, the original motion M is temporally divided into partial motions $\{M_i\}$ to ensure that the embedded watermark can be extracted even when a part of the marked motion is removed or replaced with other partial motion data. This temporal segmentation π

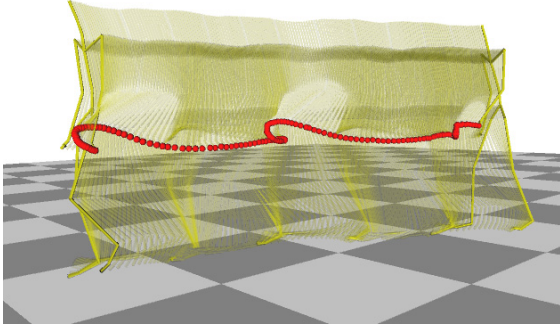


Figure 2: A trajectory of a joint in 3D space

is designed in such a way that watermark remains inside the motion data even after the attacker’s modification. We propose three approaches:

- **Feature-based segmentation of body motion:** The motion M is temporally divided according to the feature or semantics of the action. The consequent segmentation is represented as segmentation in temporal space. This can be carried out either manually or automatically [2].
- **Feature-based segmentation of joint motion:** The motion of each joint q_i is temporally divided according to the feature of the trajectory in 3D space. The number and position of consequent segments may differ for each joint.
- **Equally-spaced segmentation:** The motion M is equally divided into the segments with constant length N_π .

The number of frames in a frame must be larger than a minimum number that is required for reliable frequency analysis. For simplicity, in our experiments we used the equally-spaced segmentation with $N_\pi = 500$.

Frequency Modulation

Choose the m most significant components in $\{f_i\}$ except for DC component in the frequency domain, according to their amplitudes. They are marked as $\{\hat{f}_i\}$. Then, the watermark vector \mathbf{w} is inserted into the motion data by slightly altering \hat{f}_i according to w_i as

$$\hat{f}_i^* = (1 + \epsilon w_i) \hat{f}_i \quad (2)$$

where ϵ is a parameter which controls the intensity of watermarking.

5 Extracting Watermarks

Registration

Given a suspected motion data set M^* , a possible embedded watermark \mathbf{w}^* is extracted through the inverse operation of equation (2). Before the extraction, two motion data sets M and M^* have to be placed at the same position with respect to their parameter space.

5.1 Spatiotemporal Registration

When motion data is used to generate digital contents related to computer graphics animations, the suspected motion data is supposed to be obtained in a form which is the same as the original. In this scenario, what we have to do is bring the suspected motion data back to the same position of the original in the spatiotemporal space.

As mentioned in Section 4, the trajectory of a joint position in the motion data forms a continuous curve. Since the temporal changes of a motion data do not affect the shape of the trajectory, the registration process comes down to the estimation of similarity transformation in 3D space, by assuming that changes in the temporal space can be represented as a uniform scaling.

The similarity transformation T_s is composed of seven parameters (uniform scaling s , 3D rotation \mathbf{R} , and 3D translation \mathbf{t}). Then the relationship between an original motion M and the transformed motion M' is as follows.

$$M' = s\mathbf{R}M + \mathbf{t} \quad (3)$$

The estimation of similarity transformation is a well-studied problem in the field of computer vision. Given a reasonably good initial estimation, the seven parameters can be retrieved simultaneously, for example, by the iterative closest point (ICP) method [3]. We have extended the original ICP method so that it can recover the uniform scaling, as well as rotation and translation, based on the closed-form solution proposed by Horn [10].

Given a suspected motion M^* and the original motion M , we try to estimate the transformation T_s^i that brings M_i to the spatiotemporal position of M^* . Suppose the modification of M' is mostly caused by changes of different parameters constituting the human body, then some of the partial motions M_i are likely fit to the corresponding partial motion of M^* . Thus, T_s^i are candidates of the transformation T_s wished to be estimated. Based on this observation, once T_s^i is obtained, the whole original motion M is first transformed by T_s^i . Then, the registration error e_i between M^* and $T_s^i(M)$ is then calculated as

$$e_i^s = \sum_t \sum_i \rho(q_i^*(t) - T_s^i(q_i(t))) \quad (4)$$

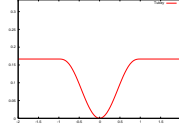
where $q_i^*(t)$ and $T_s^i(q_i(t))$ denote the position of joints in M^* and $T_s^i(M)$ respectively. ρ is one of the M-estimators [9]

that represent functions designed to reject outliers aiming stable statistical analysis. In our case, ρ decreases the influence of the joints poorly registered. We used an M-estimator based on the Tukey's biweight function

$$\rho(x) = \begin{cases} \frac{c^2}{6} \left(1 - \left(1 - \frac{x^2}{c^2}\right)^3\right) & \text{if } |x| \leq c \\ \frac{c^2}{6} & \text{otherwise} \end{cases} \quad (5)$$

where c is the parameter that controls the rate of outliers. The proper value of c depends on the scale of the motion data to be processed. After the estimation of T_s^i for all partial motion M_i , we adopt a transformation whose error e_i^s is minimum as the appropriate transformation T_s .

Note that the registration process must be performed between the suspected motion M^* and the original motion data M . We have to use M as a target motion to which M^* is transformed, since M' has already the watermark embedded.



Tukey's function

5.2 Projective-Temporal Registration

When the watermarked motion has already been used to synthesize a CG character animation, it is desirable for a detector to extract a watermark only from the 2D image sequence (i.e. movie). We tackle this problem by reconstructing the motion which was used to synthesize the images, by fitting the original motion that has not been watermarked yet.

The process of CG animation synthesis using the watermarked data can be summarized as follows.

1. The watermarked motion M^* is attached to the character that acts in the movie,
2. the scale and position of the motion is changed into the scene by similarity transformation T^s ,
3. a character figure and other decorations are attached to the motion,
4. finally the character is rendered into an image sequence by projective transformation T^p onto the rendered scene.

For simplicity, we assume that the articulated motion M^\dagger in the image sequence has been extracted in the same link structure order as that of M , either manually or automatically by existing motion tracking methods, for instance, particle filtering [11].

In order to retrieve M^* from the image sequence, we must perform the transformation of the images in a reverse order. Generally, this problem has no unique solution without some additional constraints, since the projective transformation T^p is irreversible. In our case, the original motion M can be used as the reference motion that needs to

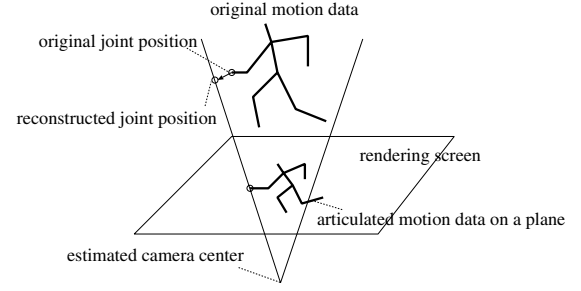


Figure 3: Reconstruction of a motion in 3D space from a 2D image sequence.

be reconstructed. Thus the above ill-posed reconstruction problem can be regarded as the fitting of 2D motion data into 3D motion data in the projective space.

The transformation from M^* to M^\dagger can be represented by nine parameters consisting of focal length f , camera center u, v , rotation \mathbf{R} and translation \mathbf{t} as

$$M^\dagger = P_{f,u,v}(\mathbf{R}M^* + \mathbf{t}), \quad (6)$$

where $P_{f,u,v}$ is the operator of projective transformation. The scaling parameter s in similarity transformation is eliminated due to the ambiguity in projective transformation.

The estimation of joint projective and similarity transformation can be solved through the minimization of the objective function derived from equation (6). To the best of our knowledge, there is no linear method to solve this problem. Hence, we formulate the objective function and minimize it through a nonlinear minimization method, such as Levenberg-Marquardt [15].

Similar to the case of similarity transformation, we estimate the whole transformation by finding a consensus between the transformations estimated using partial motions. First we divide the motion M into partial motions M_i according to π , and then calculate registration error e_i^p between transformed M and M^\dagger . Finally the best transformation with minimum error e_i^p is chosen as the transformation from M to M^\dagger . Again, M^\dagger must be registered with M , not M' .

Once T^p has been estimated, M^* is generated from M^\dagger . Since the projective transformation is irreversible, M^* cannot be reconstructed uniquely from M^\dagger . Instead, we reconstruct M^* from M^\dagger by fitting M^* to M through a least square minimization of the Euclid distances between joint positions, considering the constraint of equation (6). This process is illustrated in Figure 3.

Resampling

Once the spatiotemporal position of the suspected motion data is registered to that of the original motion data, the

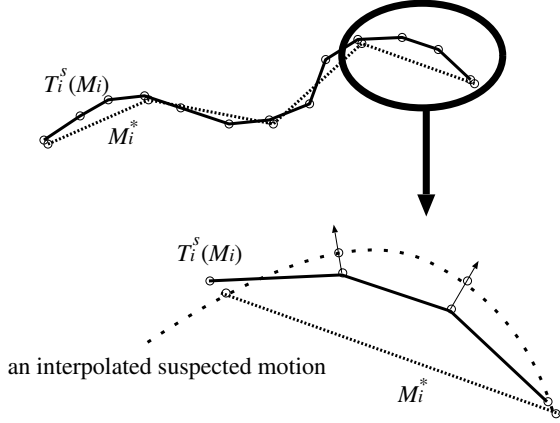


Figure 4: Resampling of motion data

data is resampled so that M^* and M' has a one-to-one correspondence in the overlapping region. First, we interpolate the suspected motion by cubic spline interpolation. Then, the closest point to every sample points in M on the spline curve are resampled, resulting in the resampled suspected motion (Figure 4).

Watermark Detection

Given a suspected motion data M^* , we can extract a possible existing watermark $\{w_i^*\}$ by first converting this data into the frequency domain and inverting the operation defined in equation (2). While the extracted watermark $\{w_i^*\}$ may not be the same as the watermark $\{w_i\}$ originally embedded to M , the existence of watermark can be detected by statistically testing the similarity between $\{w_i\}$ and $\{w_i^*\}$ [6]. We use a linear correlation coefficient as the metric of the similarity,

$$r = \frac{\sum_{i=1}^m (w_i - \bar{w}_i)(w_i^* - \bar{w}_i^*)}{\sqrt{\sum_{i=1}^m (w_i - \bar{w}_i)^2 \cdot \sum_{i=1}^m (w_i^* - \bar{w}_i^*)^2}} \quad (7)$$

where \bar{w}_i and \bar{w}_i^* are the mean of \mathbf{w} and \mathbf{w}^* respectively. Similar to the detection process in the mesh watermarking method proposed by Praun et al [14] we discard the watermark w_i whose absolute value exceeds a pre-defined threshold w_{thresh} as outliers.

When watermarking is used to prove ownership claims, the system should be designed to minimize the probability of a false positive detection, that is, the probability that the system detects a watermark from the data which are not watermarked. The false positive probability can be calculated from the correlation coefficient r by Fischer's z -transformation [15]. When $\{w_i\}$ and $\{w_i^*\}$ are randomly sampled from a normal distribution $N(0, 1)$, it is known that

$$z = \frac{1}{2} \ln \left(\frac{1+r}{1-r} \right) \quad (8)$$

Table 1: Probabilities of false positive detection. The shaded cells correspond the cases that the watermark could not be detected properly.

Attack	walk	run	dance	boxing
A. No attack	0	0	0	0
B. Noise 1%	10^{-16}	10^{-11}	10^{-21}	10^{-2}
C. Noise 2%	10^{-5}	10^{-4}	10^{-6}	10^{-1}
D. Smooth	10^{-7}	10^{-12}	10^{-15}	10^{-5}
E. Downsample 1/8	10^{-71}	10^{-98}	10^{-2}	10^{-4}
F. Downsample 1/32	10^{-2}	10^{-14}	10^{-3}	10^{-2}
G. Double watermark	10^{-13}	10^{-9}	10^{-13}	10^{-6}
H. Triple watermark	10^{-7}	10^{-6}	10^{-9}	10^{-5}
I. Modify 50%	10^{-8}	10^{-3}	10^{-3}	10^{-2}
J. Modify 90%	10^{-2}	10^{-2}	10^{-2}	10^{-2}
K. Similarity trans.	0	10^{-320}	0	0
L. Projective trans.	10^{-16}	10^{-16}	10^{-34}	10^{-15}
M. CG rendering	10^{-2}	10^{-1}	10^{-3}	10^{-1}
N. D+H	10^{-5}	10^{-5}	10^{-5}	10^{-2}
O. B+E	10^{-3}	10^{-39}	10^{-8}	10^{-2}
P. B+E+K	10^{-2}	10^{-9}	10^{-8}	10^{-2}
Q. B+K+L	10^{-2}	10^{-1}	10^{-3}	10^{-1}

is approximately distributed as $N(0, \frac{1}{\sqrt{m-3}})$. Accordingly, the probability of false-positive detection can be calculated by

$$P_{fp} = \text{err} \left(\frac{|z| \sqrt{m-3}}{\sqrt{2}} \right) \quad (9)$$

where

$$\text{err}(x) = \frac{2}{\sqrt{\pi}} \int_x^{\infty} e^{-t^2} dt. \quad (10)$$

6 Results

We have implemented the algorithm described above on a standard IBM-compatible PC equipped with Pentium4 3.4GHz and 1GB memory. The motion data used in our experiments are captured by MoCap measurements. Each motion data is a sequence of 1024 posture frames composed of 31 joints. Three of the motion data sets, “walk”, “run” and “dance”, are represented by 3D positions of the joints, while the other, “boxing”, is given in angular representation.

In the following experiments, the length of a watermark vector \mathbf{w} is set to be 50. The scale ϵ of watermark embedding is set to be 0.001. The threshold w_{thresh} for watermark detection is set to be 3.

6.1 Resilience

Table 1 shows the result of watermark detection in the presence of various adversary attacks. The figures in the table show the probability of false-positive detection. That is to

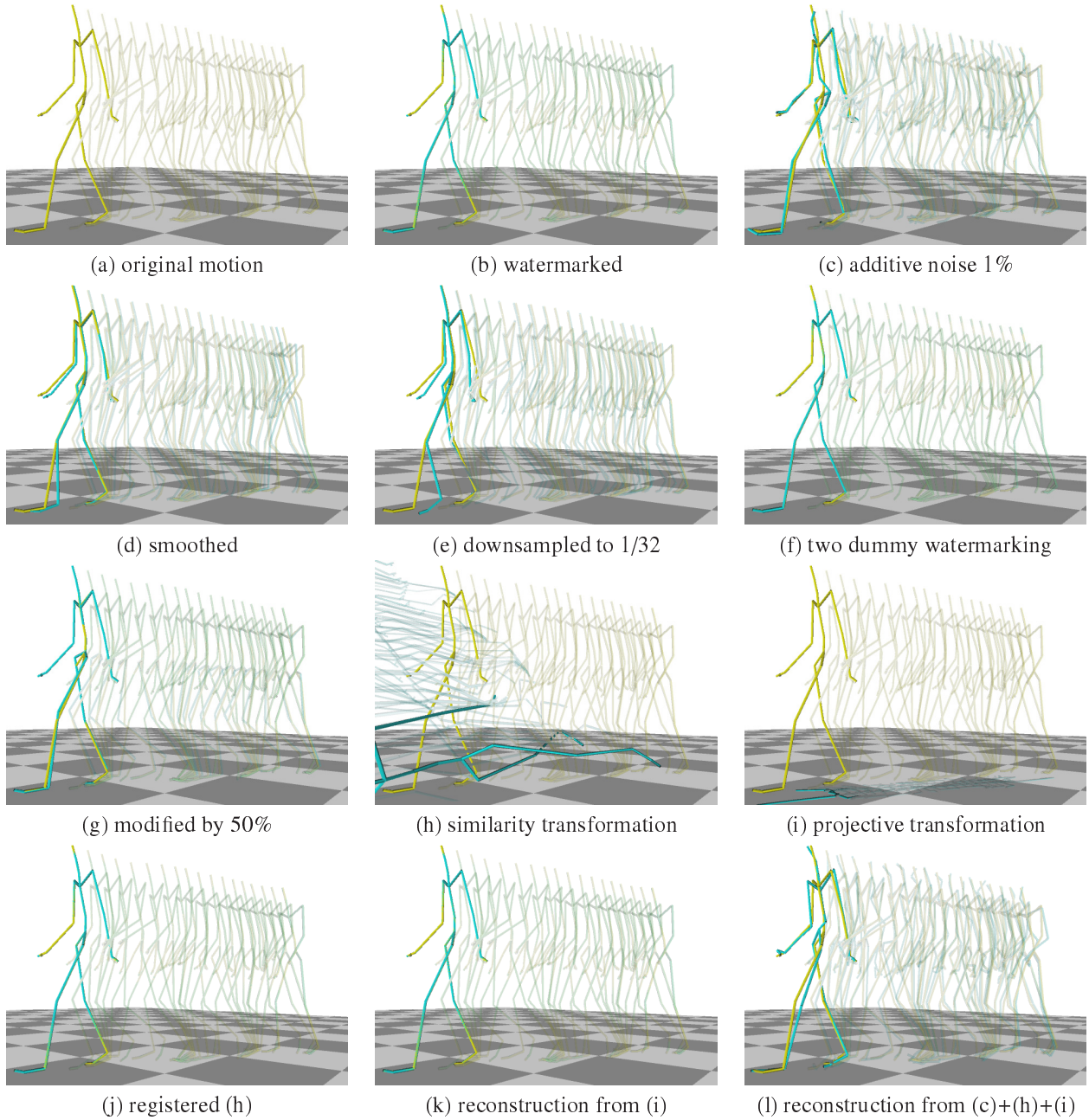


Figure 5: Motion data used in our experiments. The original and watermarked motion data are shown in (a) and (b) respectively. We simulated various attacks to (b), which are presented in (c)-(h). The figures (j), (k) and (l) are the results of registration proposed in Section 5.

say, when the probability is small, the embedded watermark is likely present in the motion data. As shown in the first row in Table 1, the false-positive probability is equal to 0 when the extracted watermark is exactly the same as that embedded to the original motion.

Random Noise: In Table 1, the rows B-C show the resilience of the watermark under additive random noise. We added random noise to each joint parameter with three difference amplitudes. The percentage in the table is the ratio of the noise to the joint parameters. The disturbed motion is shown in Figure 5 (c). Since our algorithm is based on spread spectrum watermarking, and human motion consists mostly of low frequency components, our watermarking method is robust to such high frequency disturbance as additive noise.

Smoothing and Downsampling: The row D shows the resilience against temporal smoothing. We replaces each joint parameter in motion data with the mean value of 20 adjacent parameters in the temporal domain.

The rows E-F also show the resilience against temporal smoothing. But in this experiments, the motion data is first downsampled in the temporal domain, then interpolated by cubic spline interpolation, and finally resampled at the same rate as that of the original motion data. We performed this experiment at two different downsampling ratio as shown in the table.

Although the attacked motion data sets in this experiments do not resemble the original motion any more due to the large scale modification (Figure 5 (d) and (e)), the embedded watermarks were successfully detected.

Watermark Overwriting: Technically speaking, what the spread spectrum watermarking does is modulate the frequency components of motion data. Accordingly, the disturbance of the modulated frequency components affects the probability of the watermark detection. Such attack can be achieved by applying another watermark to the motion data by the same embedding algorithm (Figure 5 (f)).

The rows G-H show the result of watermark detection in the presence of 2nd and 3rd watermarks which have no relation with the original watermark. As shown in the table, the detection rate tends to decrease with the increase of the number of watermarking.

Body Form Modification: The rows I-J show the watermark resilience against the modification of the parameters constituting the human body. This process is referred to as retargeting in the field of motion data processing. We simulated the retargeting process by randomly changing the length of links in an articulated structure. Observing the detailed process of watermark detection, most of the wa-

Table 2: Timing (in seconds) for insertion and extraction of watermark. In the first column, “3D-3D reg.” and “2D-2D reg.” indicate respectively spatiotemporal and projective-temporal registration.

Process	walk	run	dance	boxing
Embedding	0.1	0.1	0.1	0.1
3D-3D reg.	2.2	3.0	2.3	3.3
2D-2D reg.	8.0	10.2	6.8	8.7
Extraction & Detection	0.3	0.3	0.3	0.3

termarks extracted from the joints on the modified links were discarded as outliers (Figure 5 (g)).

Transformation: The row K shows demonstrates the resilience of the watermark under the attack of similarity transformation. We simulated the attack by randomly determining seven parameters in the transformation and applied the transformation to watermarked motion data (Figure 5 (h)). In the watermark detection process, we first estimate the transformation by the method described in Section 5.1 (Figure 5 (j)).

The watermark detection from CG animations is demonstrated in the rows L-M. We simulated this attack as follows. First, a watermarked motion data is attached to a CG character by the Curious Labs Posed software. In the experiment shown in the row L, the character is an articulated links as illustrated in Figure 5 (i). On the other hand, in the experiment presented in the row M, a fully skinned human body is attached to the motion data. The character is then properly placed in a scene, and rendered into a CG animation. In order to extract the watermark from the animation, we first manually specified the articulated link structure of the character in the rendered scene, at every frame of the animation. Using the extracted motion data in the rendered scene, a motion data in the same form as the original motion is generated by the registration process described in Section 5.2 (Figure 5 (k)). As shown in Table 1, some of the results indicate that the watermark could not detected appropriately. This is mainly because of wrong extraction of articulated link structures in the rendered scenes.

Combination: The resilience of the watermark against the combination of above-mentioned attacks is demonstrated in the rows N-Q (Figure 5 (l)).

6.2 Timing

Table 2 shows the typical running times for various operations in our watermarking process. Note that our current implementation is not optimized for execution time.

7 Conclusion

In this paper, we have proposed a method of robust, informed watermarking for the motion data of those related to human motion. In order to achieve the robustness against both ordinary signal processing and adversary attacks, we extended spread spectrum watermarking to motion data. We also proposed the methods for comparing a suspected motion data with the original. Our method allows the detector to extract watermark from a CG animation that may be generated using the original motion data, as well as the ordinary motion data in the same form as that of the original.

As mentioned in Section 1, the major difficulties in motion watermarking comes from the flexibility of the data modification, including different free-form deformations, during the authoring process of multimedia contents. We tackled this problem by temporal partitioning of motion data, but the watermark detection fails in some cases when the watermarked motion data undergo a radical modification, as presented on shadowed cells in Table 1. Even in these cases, the probability of the detection can be improved by applying a strong watermark. However, the fidelity of the watermarked motion data tends to degrade. Basically, this is a fundamental trade-off relationship in watermarking problem. It may be improved by taking into account the human perception mechanism and semantics of the motion, which remains as future work.

Acknowledgement

The motion data used in this paper was obtained from the CMU motion capture database. The database was created with funding from NSF EIA-0196217.

References

- [1] Stefan Katzenbeisser André Adelsbach and Ahmad-Reza Sadeghi. On the insecurity of non-invertible watermarking schemes for dispute resolving. In *Proc. International Workshop on Digital Watermarking*, pages 355–369, 2003.
- [2] Jernej Barbič, Alla Safonova, Jia-Yu Pan, Christos Faloutsos, Jessica K. Hodgins, and Nancy S. Pollard. Segmenting motion capture data into distinct behaviors. In *Proc. Graphics Interface 2004*, pages 185 – 194, 2004.
- [3] Paul J. Besl and Neil D. McKay. A method for registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.
- [4] Bobby Bodenheimer, Chuck Rose, Seth Rosenthal, and John Pella. The process of motion capture – dealing with the data. In *Proc. the Eurographics Workshop on Computer Animation and Simulation*, pages 3–18, 1997.
- [5] Armin Bruderlin and Lance Williams. Motion signal processing. In *Proc. SIGGRAPH '95*, pages 97–104, 1995.
- [6] Ingemar Cox, Joe Kilian, Tom Leighton, and Talal Shamoosh. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing*, 6(12):1673–1687, 1997.
- [7] Ingemar Cox, Matthew Miller, and Jeffrey Bloom. *Digital watermarking*. Morgan Kaufmann Publishers Inc., Pine Street, San Francisco, 2002.
- [8] Scott Craver, Nasir Memon, Boon-Lock Yeo, and Minerva M. Yeung. Resolving rightful ownerships with invisible watermarking techniques: Limitations, attacks, and implications. *IEEE Journal on Selected Areas in Communications*, 16(4):573–586, 1998.
- [9] Frank R. Hampel, Elvezio M. Ronchetti, Peter J. Rousseeuw, and Werner A. Stahel. *Robust Statistics: The Approach Based on Influence Functions*. John Wiley, 1986.
- [10] Berthold K. P. Horn, Hugh M. Hilden, and Shahriar Negahdaripour. Closed-form solution of absolute orientation using orthonormal matrices. *Journal of the Optical Society of America A*, 5(7), July 1988.
- [11] Michael Isard and Andrew Blake. Condensation – conditional density propagation for visual tracking. *International Journal of Computer Vision*, 29(1):5–28, 1998.
- [12] Qiming Li and Ee-Chien Chang. On the possibility of non-invertible watermarking schemes. In *Proc. 6th Information Hiding Workshop*, 2004.
- [13] Makoto Matsumoto and Takuji Nishimura. Mersenne twister: A 623-dimensionally equidistributed uniform pseudorandom number generator. *ACM Transactions on Modeling and Computer Simulation*, 8(1):3–30, January 1998.
- [14] Emil Praun, Hugues Hoppe, and Adam Finkelstein. Robust mesh watermarking. In *Proc. SIGGRAPH '99*, pages 49–56, 1999.
- [15] William H. Press, Brian P. Flannery, Saul A. Teukolsky, and William T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.
- [16] Fabien A. P. Petitcolas Stefan Katzenbeisser, editor. *Information Hiding Techniques for Steganography and Digital Watermarking*. Artech House Books, January 2000.
- [17] US Department of Commerce, National Institute of Standards and Technology (NIST). *Secure Hash Standard*, April 1995.
- [18] Andrew Witkin and Zoran Popovic. Motion warping. In *Proc. SIGGRAPH '95*, pages 105–108, 1995.
- [19] Katsu Yamane and Yoshihiko Nakamura. Dynamics filter - concept and implementation of on-line motion generator for human figures. *IEEE Transactions on Robotics and Automation*, 19(3):421–432, 2003.