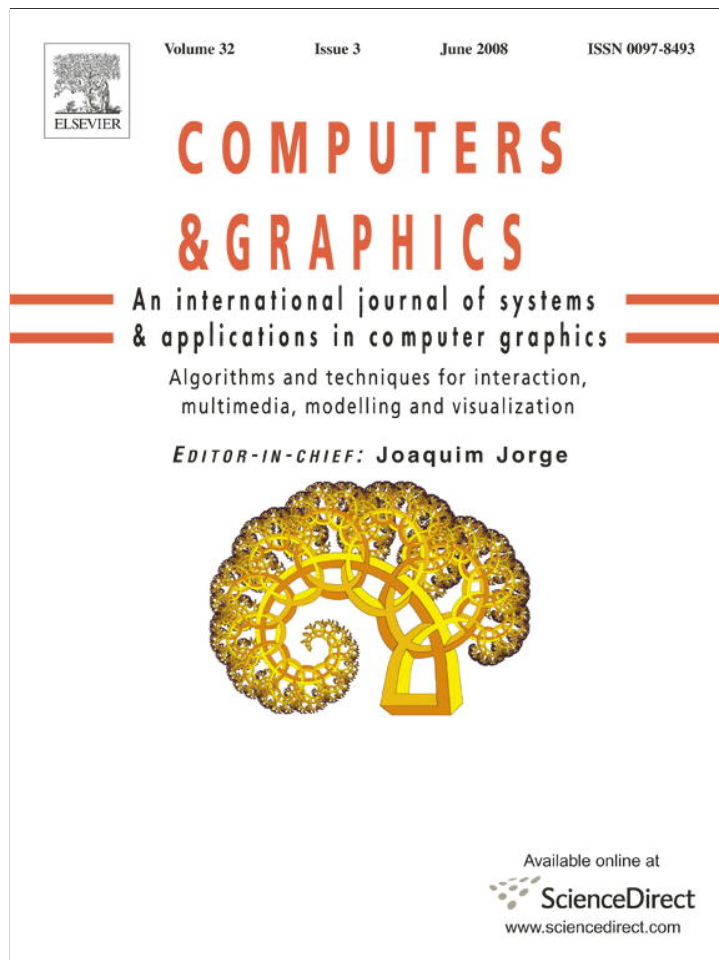


Provided for non-commercial research and education use.
Not for reproduction, distribution or commercial use.



This article appeared in a journal published by Elsevier. The attached copy is furnished to the author for internal non-commercial research and education use, including for instruction at the authors institution and sharing with colleagues.

Other uses, including reproduction and distribution, or selling or licensing copies, or posting to personal, institutional or third party websites are prohibited.

In most cases authors are permitted to post their version of the article (e.g. in Word or Tex form) to their personal website or institutional repository. Authors requiring further information regarding Elsevier's archiving and manuscript policies are encouraged to visit:

<http://www.elsevier.com/copyright>



ELSEVIER

Contents lists available at ScienceDirect

Computers & Graphics

journal homepage: www.elsevier.com/locate/cag

Technical Section

Robust watermarking motion data with DL-STDM

Xiaomao Wu^{a,b,*}, Lizhuang Ma^b, Zhuoqun Dong^b, Lionel Revéret^a^a Evasion/LJK, INRIA, France^b Department of Computer Science & Engineering, Shanghai Jiao Tong University, Shanghai, China

ARTICLE INFO

Article history:

Received 17 July 2006

Received in revised form

1 April 2008

Accepted 10 April 2008

Keywords:

Robust watermarking

Copyright protection

Motion capture data

QJM

STDM

ABSTRACT

In this paper, we propose a robust watermarking scheme called Double-Layer Spread-Transform Dither Modulation (DL-STDM) for watermarking motion data. We embed watermarks into the DCT domain of the quaternion logarithm image (QLI) representation of motion data. Experimental results demonstrate that our scheme is more robust than existing motion watermarking schemes against a wide variety of attacks, including noise addition, smoothing, cropping, attenuation, simplification, time warping, non-uniform scaling along amplitude axis and 2nd watermarking. Our scheme is useful for copyright protection and ownership identification of motion data.

© 2008 Elsevier Ltd. All rights reserved.

1. Introduction

Digital watermarking techniques have been widely applied to images, videos, 3D models and motion data. These techniques can be classified into two categories: robust and (semi-)fragile watermarking. In a robust watermarking scheme, the watermark is embedded into the original data. Any attempt to remove the embedded watermark will not succeed unless it corrupts the watermarked data too much. On the other hand, a (semi-)fragile watermark scheme is capable of detecting any change in the watermarked data and possibly the locations of the changed parts. The former technique is more challenging.

Unlike popular watermarking applications, watermarking motion data has been rarely explored. Use of motion capture (mocap) techniques is becoming increasingly popular for producing realistic character animation. Professional mocap data providers are emerging [1,2]. These companies provide customized mocap data for users. They need techniques to protect their data from unauthorized usage or illegal copying, and to claim ownership of the data when necessary. On the other hand, owners of freely available motion databases [3] may want to ensure that their data are not being used commercially. When they suspect someone using their data for commercial purpose, they need techniques to claim the ownership of their data.

* Corresponding author at: INRIA Rhone-Alpes, Evasion group, 55 Avenue de L'Europe, 38334 Grenoble, France. Tel.: +33 476 615445; fax: +33 476 615466.
E-mail address: xiaomao.wu@inria.fr (X. Wu).

Pioneering work on robust watermarking of motion data was done by Kim et al. [4]. Their algorithm has proven successful at resisting attacks such as noise addition, smoothing, cropping, time warping, simplifying, enhancement, attenuation and 2nd watermarking. However, they did not test with large data sets, and their framework cannot resist non-uniform scaling along amplitude axis (NUSA) attack. A typical NUSA operation is motion warping [5–7].

As Kim's scheme is based on Cox's method [8], the more recent and powerful spread transform Dither modulation (STDM) [9] scheme is expected to be more robust than Kim's scheme. Unfortunately, our experiments show that the STDM scheme performs worse than Kim's scheme (Table 4). We thus propose a robust watermarking scheme called Double-Layer Spread-Transform Dither Modulation (DL-STDM). This scheme is derived from STDM, but differs from traditional STDM scheme in two aspects: the dither generation method, and the number of layers. In our scheme, the watermark is embedded into the discrete cosine transform (DCT) domain of the quaternion logarithm image (QLI) (Section 4.1) of motions. We verified our method with 1000 motions selected from the CMU motion database [3]. Experimental results show that our scheme is more robust than Kim's against a wide variety of attacks.

The remainder of this paper is organized as follows. After a review of related work, we give an overview of our scheme in Section 3. Then we describe our approach in detail in Section 4, followed by the description and analysis of our experimental results. Finally, we give a conclusion and propose possible future work.

2. Related work

2.1. Watermarking images

Early image watermarking algorithms [10,11] employ a quantize-and-replace strategy. A simple implementation of such algorithms is low-bit modulation (LBM), where the least significant bits of the host signal will be replaced by the embedded signal. Additive spread-spectrum-based methods [8] embed the watermark into transformed domains such as the DCT domain. They are more robust than LBM against most common attacks, and can be conveniently integrated into the standard image processing pipelines such as JPEG compression. More recently, quantization index modulation (QIM)-based methods [9] have received considerable attention. QIM-based methods have been shown to be better than the above two kind of methods against a wide variety of attacks. Our approach is derived from STDM [9] which is an efficient implementation of QIM.

2.2. Watermarking meshes

Existing mesh-watermarking algorithms can be classified into two categories: watermarking in spatial domain [12–15] and watermarking in different kinds of transformed domains [16–20]. Among them, Benedens [12] proposed the first watermarking technique for copyright protection of meshes. He subsequently proposed a scheme that can resist both affine transformations and mesh simplifications [13]. More recently, Zafeiriou et al. [14] proposed a blind robust mesh-watermarking scheme. Bors [15] used local moments for watermarking 3D mesh objects. In [16], the watermark is embedded using a spread-spectrum watermarking scheme. In [18], the watermark was embedded into the mesh spectral domain. Uccheddu et al. [19] embedded the watermark in the wavelet domain. Li et al. [20] used global spherical parameterization to parameterize meshes and embedded the watermark into the Fourier-frequency domain of the original mesh. Information hiding techniques have also been studied by researchers [21,22], and several fragile watermarking schemes for meshes have been proposed [23,24].

2.3. Watermarking motion data

Little work has been done on watermarking motion data. Among them, Kim et al. [4] first presented a robust watermarking scheme based on Cox's method [8]. They embed watermarks into the coefficients of the multiresolution representation of motions in the quaternion domain. Their algorithm is robust against a wide range of attacks. However, their algorithm cannot resist the attack of NUSA such as motion warping [5–7]. It is also sensitive to motion types, and has not been verified on large motion databases. Yamazaki [25] proposed a watermarking scheme that is also based on Cox's method [8]. They embed watermarks into the frequency domain of the Euler angles or alternatively the joint positions of motions. Although their scheme is robust against

noise addition, smoothing, down sampling, 2nd and 3rd watermarking, bone-length changing, and transformation, it is not coordinate-independent, and suffers from the same problem as Kim's method. Recently, Agarwal et al. [26] formulated joint positions of motions as triangular meshes, and used a macro-embedding procedure (MEP) for watermarking motion data. This algorithm was tested against dropping, noise addition, smoothing reordering and affine transformation attacks. However, other attacks such as simplification, cropping, NUSA, 2nd watermarking, enhancement and attenuation were not tested, and only one motion is tested in their experiment. In this paper, we present an algorithm that is robust against a wide range of attacks. Our algorithm is more robust than existing methods in most cases and has been tested with CMU's motion database [3].

3. Overview of our approach

The overview of our watermarking scheme is illustrated in Fig. 1.

The motion data to be watermarked is \mathcal{M} . The proposed DL-STDM scheme works as follows. First, embed the occurrence message \mathbf{m}_o and then the authentication message \mathbf{m}_a into \mathcal{M} . These messages are embedded into the DCT domain of the QLI (Section 4.1) using a modified STDM method [9]. We thus obtain the watermarked motion \mathcal{M}^s . The according dithers $\mathcal{D}_o = [d_0, d_1]$ and $\mathcal{D}_a = [d'_0, d'_1]$, along with the index matrix \mathcal{I}_o and \mathcal{I}_a , and the random matrix \mathcal{U}_o and \mathcal{U}_a , are all saved to the database of the owner as these data are important for the watermark extracting process. After that, the motion is distributed to users and it may be changed by intentional attacks, turning into \mathcal{M}^y . In order to extract the embedded watermark from a suspect motion \mathcal{M}^y , we input \mathcal{M}^y into the watermark extracting block. If the watermarked motion has been attacked by time warping or cropping, then the attacked motion should be resampled or registered before the watermark extracting process. When both \mathbf{m}_o and \mathbf{m}_a are successively extracted from the suspect motion, we can claim the ownership of that motion.

4. The DL-STDM scheme

In this section, we first describe the QLI representation for motion data, and explain why we use this representation for our scheme. Then we briefly introduce the traditional STDM model, followed by a detailed description of the proposed DL-STDM scheme.

4.1. Representation

A motion of an articulated figure can be described by a sequence of poses. A pose is specified by the translation and orientation of the root joint, along with the orientation of the other joints [27,28]. Suppose that the pose at time t is represented by $\mathbf{m}(t) = (\mathbf{p}(t), \mathbf{q}_1(t), \dots, \mathbf{q}_n(t))^T$, where $\mathbf{p}(t) \in \mathbb{R}^3$ and $\mathbf{q}_1(t) \in \mathbb{S}^3$

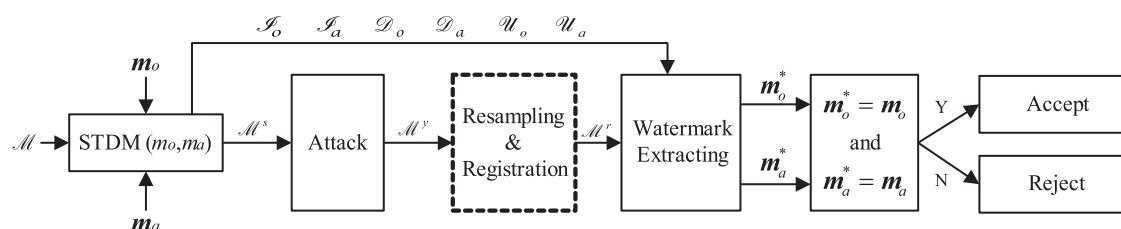


Fig. 1. Overview of double-layer STDM scheme.

represent the translation and orientation of the root joint, respectively. $\mathbf{q}_i(t) \in \mathbb{S}^3$ represents the orientation of the i -th joint for $2 \leq i \leq n$, where n is the number of joints.

In order to obtain the correspondence between four-component quaternions and three-component RGB images, we propose the QLI representation. QLI is obtained as follows: each unit quaternion $\mathbf{q}_i = [w_i, \mathbf{v}_i]$ can be converted to an equivalent form $\mathbf{q}_i = [\cos(\theta), \mathbf{v}'_i \sin(\theta)]$, where $\cos(\theta) = w_i$, $\sin(\theta) = |\mathbf{v}_i|$ and $\mathbf{v}'_i = \mathbf{v}_i/|\mathbf{v}_i|$. Here, we simply describe $\mathbf{q}_i(t)$ as \mathbf{q}_i . The logarithmic representation of \mathbf{q}_i is then defined by $\log(\mathbf{q}_i) = [0, \theta\mathbf{v}'_i]$ [28–30]. If we treat the three components of $\theta\mathbf{v}'_i$ as the RGB components of a pixel and construct an image with these components, we can obtain a QLI. An example of QLI is illustrated in Fig. 2. Each horizontal line of pixels represents the three quaternion logarithmic components of one joint. Each vertical line of pixels represents the quaternion logarithmic components of all joints in one frame. The QLI has n rows and k columns, where n and k denote the number of frames and the number of joints of a motion, respectively.

The data type of each pixel in QLI is floating point. The data range of the value of each component of each pixel is $[-\pi, \pi]$. The QLI is stored in physical memory during the watermark embedding and extracting process.

Our DL-STDm is built upon the QLI representation. Other representations including Euler angles, quaternions, and coefficients of the multiresolution representation of quaternions [4,31], have also been considered. A watermarking scheme that embeds watermarks into Euler angles is more sensitive to noise than a scheme that embeds watermarks into the QLI. Embedding watermarks in the quaternion domain is undesirable, since the four components of a quaternion have a redundant degree of freedom. The DCT transformation of quaternions may produce lots of zeros. As these zeros are redundant, they may decrease the performance of a watermarking system. We have also considered embedding the watermark in the coefficients of the multiresolution representation. However, this representation is not fully reversible during the watermarking process. For example, if two components of the watermark are located side by side, we cannot extract the watermark precisely, even if the motion data were not modified. This property is harmless to Kim's method which relies on a statistical similarity analysis to detect the watermark, but is not suitable for our scheme which explicitly extracts each bit of the watermark one by one.

In our scheme, the translation components of all joints and the orientation of the root joint are not used. This means our scheme can resist global translation and rotation attacks, as well as segment scaling attacks such as motion retargeting [32].

4.2. The STDm model

The QIM model proposed by Chen and Wornell [9] has been shown to be more robust against a wide range of attacks than traditional spread-spectrum methods such as Cox's method [8]. STDm is a special class of the implementation of QIM [9].

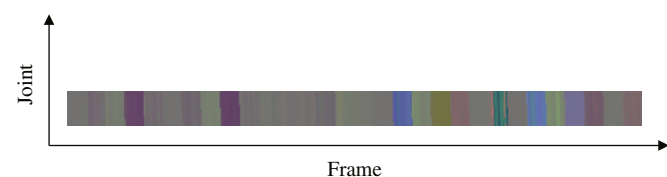


Fig. 2. The QLI of a motion clip containing 512 frames and 31 joints. The RGB components of each pixel in the image represent the according three components of the quaternion logarithmic components of one joint at one frame.

In the QIM model, information is embedded into the host signal by modulating a sequence of indices with the embedded information and then quantizing the host signal with the associated quantizers. Quantizers are generated with parameter Δ_k [9]. Fig. 3 illustrates the QIM technique. In this case, two quantizers are used, denoted by $\times(0)$ and $\circ(1)$. The information that we want to embed into the host signal, denoted by \mathbf{m} , is chosen to be binary data. If $\mathbf{m}_i = 0$, the host signal is quantized with the \times -quantizer, i.e., the \times closest to the original signal will be chosen. If $\mathbf{m}_i = 1$, the \circ quantizer will be chosen. The quantizing function has the property $\mathbf{s}(\mathbf{x}; \mathbf{m}) \approx \mathbf{x}$, where \mathbf{x} is the original signal. In such a way, the quantized signal is not perceptibly different from the original signal.

The minimum-distance decoder can be utilized to decode the embedded information from the quantized signal:

$$\hat{\mathbf{m}}(\mathbf{y}) = \arg \min_m \|\mathbf{y} - \mathbf{s}(\mathbf{y}; m)\| \quad (1)$$

where \mathbf{y} is the watermarked signal, $\mathbf{s}(\mathbf{y}; m)$ is the quantizing function.

A low-complexity realization of QIM is coded binary dither modulation, and STDm is a special case of this, in which only projections of the host signal along certain orthogonal vectors are quantized. Quantizing only a subset of host signal components has the advantage of low-signal-to-noise distortion [9].

4.3. The DL-STDm scheme

4.3.1. Motivation

The STDm scheme has been shown to be powerful for robust watermarking of images, but our experimental results show that if we directly apply the STDm algorithm to motion data, the performance is worse than Kim's method (Table 4). This is mainly because the human visual system is more sensitive to artifacts of motions than that of images. When we apply STDm to motion data, it is very difficult to choose a good dither generation parameter Δ_k [9]. If Δ_k is small, the original motion will not be changed too much, but the watermarked motion becomes fragile against intended attacks. On the other hand, if Δ_k is bigger, the watermarked motion will be more robust against intended attacks, but the original motion will be changed too much, producing artifacts such as trembling and sliding. It is difficult to achieve a good tradeoff. We have tested several parameters and selected the best one. Unfortunately it still does not give us satisfactory results. This fact leads us to propose a new scheme called DL-STDm that is based on STDm, but adapted to motion data.

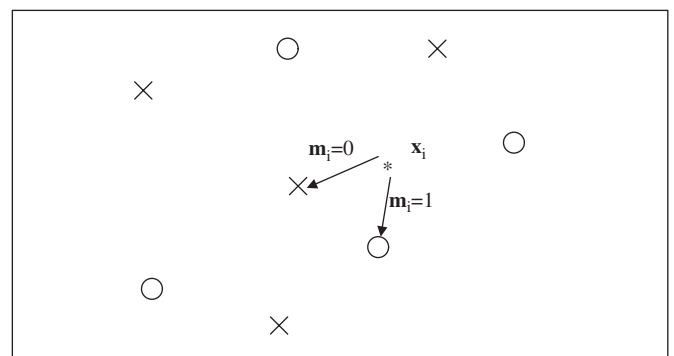


Fig. 3. QIM for information embedding [9]. The points marked with “ \times ” and “ \circ ” belong to quantizer-0 and quantizer-1. \mathbf{x}_i is quantized to the nearest “ \times ” or “ \circ ” according to the bit of \mathbf{m}_i .

4.3.2. The double-layer structure

The double-layer structure is designed to obtain a good balance between the robustness of the watermarking scheme and the distortion that the watermarking process will impose on the original motion.

In order to decrease distortion, we change the dither generating process of the standard STDM scheme. Specifically, the dither is generated with a new procedure (Eq. (4)) in the DL-STDM scheme, while in standard STDM, the dither is generated randomly. This strategy decreases the distortion that the watermarking process will impose on the original motion to a low level (Eq. (5)), while still maintaining the robustness of the watermarking system.

The potential problem with the modified dither generation approach is the increase of the false positive rate (FPR). We thus add an additional layer called *the occurrence layer*, to resolve this problem. One binary bit is embedded into the occurrence layer in order to identify whether a motion is watermarked or not. This means a motion that is not watermarked will not be detected to be watermarked, thus the FPR is decreased to a low level. The other layer corresponding to the occurrence layer is the *authentication layer* into which the 64-bit watermark will be embedded.

To embed a watermark, we first embed the occurrence layer with a 1-bit message \mathbf{m}_o , and then embed the authentication layer with a 64-bit message \mathbf{m}_a . The superposition of two layers is not a problem since the occurrence layer causes negligible change to the original motion (Table 3).

4.4. Implementation

The implementations of the two layers of the DL-STDM scheme are similar. We describe them together here. The only difference between them is the watermark length l is 1 for the occurrence layer and 64 for the authentication layer.

In the following part of this subsection, we introduce the concept of the active data set (ADS) which is very important for our scheme, and the watermark embedding and extracting processes.

4.4.1. Active data set

The watermark is embedded into the ADS of a motion clip in our scheme. The ADS is obtained with the following operations in the DCT domain of the QLI (Fig. 4):

1. Obtain the QLI by using the algorithm proposed in Section 4.1. The three components of the QLI corresponding to the RGB components of an image are illustrated in the top row of Fig. 4. Each one of the three components is an n -by- $(k-1)$ matrix, where n is the number of frames, and k is the number of joints. Only $k-1$ joints are used because the root joint is not included in the QLI.
2. Reshape each of the three matrices into an $n(k-1)$ -by-1 matrix row by row, then randomly rearrange the elements. The randomness is helpful for resisting sequential attacks such as motion warping and cropping. The original position of each element in the matrix is recorded into the index matrix represented by \mathcal{S}_o for the occurrence layer and \mathcal{S}_a for the authentication layer. \mathcal{S}_o and \mathcal{S}_a have the same size $n(k-1)$ -by-1. They are necessary for the roll back process during watermark extraction.
3. Assemble blocks according to the forward error correction (FEC)-coded watermark length L (Section 4.4.2) and the user-defined parameter f , where f is the number of blocks

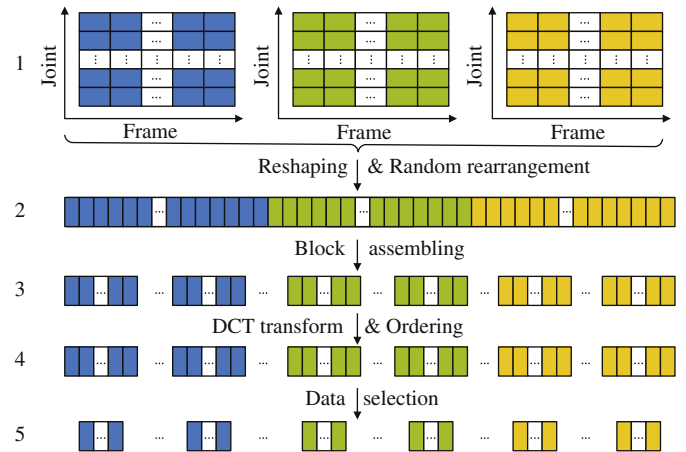


Fig. 4. Data structure for the DL-STDM scheme.

corresponding to 1-bit of the watermark. f is set to be 2 in our experiment. Thus, the number of blocks is $2L$.

4. Perform DCT transformation for each block, and sort the DCT coefficients in each block in descending order.
5. Cut off and discard the biggest and smallest part of each block. This is done because elements in each block change rapidly (Fig. 5). If we embed information into these parts, the watermarked motion is likely to produce artifacts. The coefficients of each block are 3D points. For each block, if we connect its DCT coefficients one by one from the biggest to the smallest, we obtain a 2D curve with block elements along the X-axis and DCT coefficients along the Y-axis. We define the tangent vector of the curve as the first discrete derivative of it. The normalized tangent vector of the curve is denoted by T , and its vertical component which is along the DCT-coefficient axis is denoted by T_v . The cut-off ratio depends on $|T_v|$. We examine $|T_v|$ of each block coefficient from the largest to the smallest. When we find all $|T_v|$ s of each block are below a threshold $\tau = 0.1$, we mark this position as A . Then continue searching, until we find that all $|T_v|$ s are above the threshold τ , we mark this position as B . The coefficients before A and after B are cut off. Here the threshold $\tau = 0.1$ is set by experiments.
6. Concatenate the remaining elements in the middle part of each block consecutively into a vector \mathcal{X} of length td , where t is the number of blocks and d is the number of elements remaining in each block. We define \mathcal{X} as the ADS.

4.4.2. Watermark embedding

In this section, we describe the watermark embedding processes for the two layers in the DL-STDM scheme. We define a watermark as a binary code of length l . The following description is suitable for both layers. As mentioned in the previous section, the difference between them is the watermark length l which is 1 for the occurrence layer and 64 for the authentication layer. Therefore, we uniformly describe the watermarks of the two layers as \mathbf{m} , the index matrices as \mathcal{S} , the dithers as \mathcal{D} , the quantizing steps as δ and the random matrices as \mathcal{U} .

1. Forward-error-correction (FEC) coding the watermark \mathbf{m} . Suppose the watermark is $\mathbf{m} = \{b_1, b_2, \dots, b_l\}$, then the FEC-coded watermark $\mathbf{m}' = \{b'_1, b'_2, \dots, b'_L\}$ has the length of $L = l/r$ if the code rate is r [33].
2. ADS segmentation. The ADS \mathcal{X} obtained in the 6th step in Section 4.4.1 is divided into consecutive segments according to

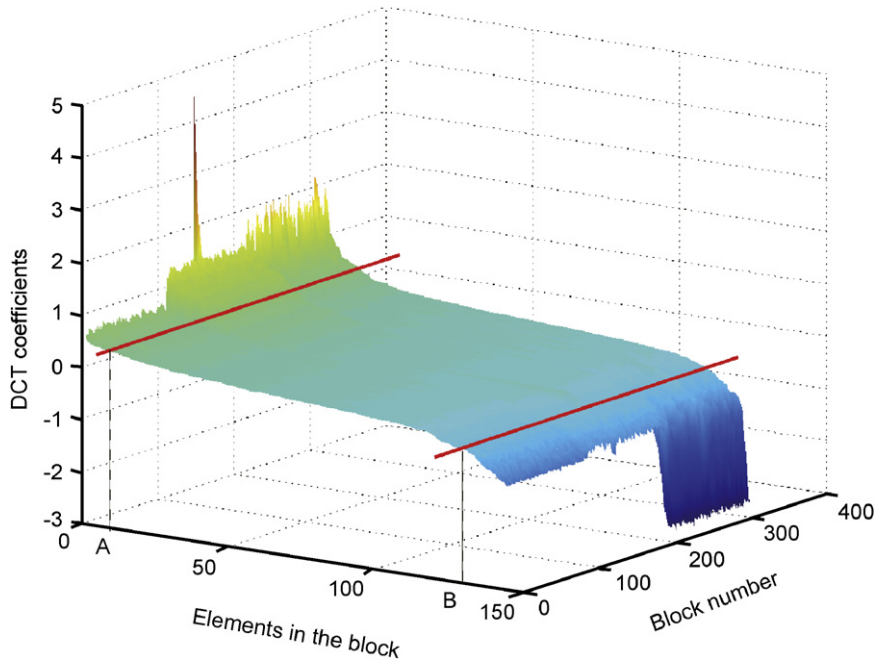


Fig. 5. DCT coefficient of 320 sorted blocks of a motion. Three hundred and twenty blocks are illustrated. Each block contains 144 sorted DCT coefficients. The middle part between the two red lines is selected for embedding watermark information.

the length L of FEC-coded watermark. The i -th segment is denoted by \mathbf{x}^i for $1 \leq i \leq L$. Thus the i -th bit of the FEC-coded watermark \mathbf{m}' will be embedded into \mathbf{x}^i . When the length of \mathcal{X} does not divide exactly by L , the remaining elements are discarded.

3. Dither generation. For binary watermarks, two dithers are needed [9], one for bit “0” and another for bit “1”. In Chen’s scheme [9], the two dithers are generated randomly. In our scheme, the dithers are generated in a different way. The reason is given in Section 4.3.2 and the implementation detail is described in Section 4.4.4. Suppose the two dithers we obtained are \mathbf{d}_0 and \mathbf{d}_1 . Their length is the same as the FEC-coded watermark length L . These two dithers should be saved as \mathcal{D} , because they are important for the watermark extraction process.
4. Obtaining matrix \mathcal{U} . This process is the same as that of the standard STDM scheme [9]. $\mathcal{U} = [\mathbf{u}_1, \mathbf{u}_2, \dots, \mathbf{u}_L]$ contains L vectors, each vector has the same length as that of \mathbf{x}^i . Each \mathbf{u}_i is generated pseudo-randomly. \mathcal{U} should be saved into the database of the owner for watermark extracting process.
5. Quantizing. The quantizing process handles one bit of the watermark and one segment of the ADS each time. Here we introduce how we embed the i -th bit of \mathbf{m}' into the i -th segment \mathbf{x}^i of the ADS \mathcal{X} . First, we project \mathbf{x}^i onto \mathbf{u}^i that is generated in the previous step, producing a scalar x_p^i . Then the j -th element of segment \mathbf{x}^i can be quantized with the following equation:

$$s_j^i(l) = (q(x_p^i + \mathbf{d}_l[l], \delta) - \mathbf{d}_l[l] - x_p^i) \mathbf{u}_j^i + x_p^i \quad (2)$$

where s_j^i is the watermarked j -th element of the i -th segment. $x_p^i = \mathbf{x}^i \cdot \mathbf{u}^i$. $\mathbf{d}_l[l]$ is the i -th value of dither l , l is the binary value of FEC-coded watermark \mathbf{m}' , $l \in \{0, 1\}$. δ is the quantizing step which is set to be 0.1 for the occurrence layer and 3.2 for the authentication layer. q is a function defined as $q(a, b) = \lfloor a/b \rfloor$. \mathbf{u}_j^i and \mathbf{x}_j^i are the j -th element of \mathbf{u}^i and \mathbf{x}^i , respectively.

6. Roll back. Since the watermark is embedded into the ADS, we should apply this change back into the original motion. This can be achieved by a roll back operation. First, substitute the

DCT coefficients that have been selected into the ADS with the quantized ones. The recorded information \mathcal{S} are utilized to get their original positions, because they have been randomly rearranged in Section 4.4.1. Then, apply the inverse DCT transformation. Finally, convert the new QLI to quaternions with an exponential map function [30].

4.4.3. Watermark extracting

The watermark extracting process is the opposite of the embedding process, and is the same for the two layers. We first establish the QLI of the suspect motion. Then obtain the ADS of it and segment the ADS. These three steps are identical to the watermark embedding process described previously in Sections 4.4.1 and 4.4.2. The only difference is that the target is changed from the original motion to the suspect motion.

After segmenting the ADS, we load the index matrix \mathcal{S} and the dithers \mathcal{D} that are saved in ADS generating process and the watermark embedding process. Then extract each bit of the FEC-coded watermark with the following equation:

$$\widehat{\mathbf{m}}_i = \arg \min_{l \in \{0,1\}} [\mathbf{y}_p^i - q(\mathbf{y}_p^i + \mathbf{d}_l[l]) + \mathbf{d}_l[l]]^2 \quad (3)$$

where $\mathbf{y}_p^i = \mathbf{y}^i \cdot \mathbf{u}^i$, \mathbf{y}^i is the counterpart of \mathbf{x}^i , except that it is the i -th segment of the ADS of the suspect motion, instead of the original motion. q is the floor function. \mathbf{d}_1 and \mathbf{d}_2 are the recorded dithers in the encoding process.

Finally, we extract the embedded watermark \mathbf{m}_i with FEC decoding [33]. If both the extracted watermarks of the occurrence layer and the authentication layer are the same as the embedded watermark, we can claim the ownership of the suspect motion.

4.4.4. Dither generation

The dithers for the two layers are uniformly generated by the following formulation in our scheme:

$$\mathbf{d}_l[l] = q(x_p^i) - x_p^i - \varepsilon, \quad l = 0, 1 \quad (4)$$

where q is the floor function, $\mathbf{d}_l[i]$ and \mathbf{x}_p^i are defined the same as Eq. (2). If we substitute the above formulation into Eq. (2), we obtain that:

$$\mathbf{s}_j^i = \varepsilon \mathbf{u}_j^i + \mathbf{x}_j^i \quad (5)$$

This means that the distortion to the original motion is $\varepsilon \mathbf{u}_j^i$, which is negligible if ε is small enough.

5. Experimental results

In this section, we demonstrate that our watermarking scheme is effective for resisting various attacks. We first introduce the parameters that we use in the experiments. Then we present results for several kinds of attacks and compare them with the results of the standard STDM scheme and Kim's algorithm [4].¹ Finally, we give results of the FPR of our scheme, the difference between the watermarked motion and the original motion and the typical running time.

5.1. Parameter setting

Parameter setting for Kim's method: The parameters for Kim's method [4] are set to be the same as in their original paper except for the watermark length. It is 64 in our test and 20 in the original paper. The scaling parameters for pelvis and other joints are 10^{-2} and 10^{-4} , respectively. When we remove the outliers of an extracted watermark, we use 2.5 times its variance.

Parameter setting for DL-STDM : The parameters we used for testing our DL-STDM scheme are listed in Table 1. δ_o is set to be 0.1, however, $0.02 \leq \delta_o \leq 1.0$ works well in our experiment. δ_a is set to be 3.2 according to Table 2 and we use this value in order to make the maximal change ratios of Kim's method and ours similar (Table 3) for a fare comparison; $2.5 \leq \delta_a \leq 3.5$ also works well. The number of check digits e for FEC coding is set to be 48. As the number of check digits increases, the robustness increases to a peak level, and then decreases; $40 \leq e \leq 48$ achieves similar results. The number of blocks that 1 bit of the watermark will be embedded into is set to be 2 which works well in all our experiments. The dither generation parameter ε is set to be 0.01, but $0.01 \leq \varepsilon \leq 0.1$ achieves similar results in our experiments. The cut-off threshold τ is set to be 0.1, which works well for all the motion types in all our experiments.

5.2. Robustness of DL-STDM

We have selected 1000 motion clips from the motion capture database of CMU graphics lab [3], in order to test the efficiency of our watermarking scheme. These clips contain various motion types including locomotion, physical activities and sports, interaction with environments, situations and scenarios and human interaction. The number of frames range from 97 to 22 948.

The test results are listed in Table 4. In this table, we list FNR and FPR under various attacks. In this table, we only list a subset of each attack type, more detailed results for different parameters of each attack type are illustrated in Fig. 6.

The threshold for Kim's method is set to be 0.1, i.e., a watermark is detected successfully if the result of the student's t -test is less than 0.1. Yamazaki [25] used the same threshold. For the STDM scheme and the DL-STDM scheme, a watermark is detected from a motion only if the extracted watermark is the

Table 1
Parameter setting for the DL-STDM scheme

Parameter	Value	Description
δ_o	0.1	Quantizing step for the occurrence layer
δ_a	3.2	Quantizing step for the authentication layer
e	48 bits	Check digits for FEC coding
f	2	Number of blocks that 1 bit will be embedded into
ε	0.01	Dither generation parameter which is used in Eq. (5)
τ	0.1	Cut-off threshold

Table 2
The false positive rate (FPR) of DL-STDM

Dither parameter δ_a	0.5	1.5	2.5	3.0	4.0	4.5	5.0
FPR (10^{-4})	0	0	0	0	12	51	101

Table 3
The distortion measurements of Kim's method compared to that of the DL-STDM scheme

Motion clip (subject-trial)	r_{\max} (10^{-4} mm)		$RSME$ (10^{-4} mm)			
	KIM	DL-STDM	KIM		DL-STDM	
	Occ. Aut.		Occ.		Aut.	
3-01	1350	8	1254	140	3	167
23-04	563	11	551	82	5	94
78-34	1512	20	1413	210	9	176

The first column shows the subject-trial IDs of the motion clips in CMU motion database. Occ. and Aut. represent the occurrence and authentication layer, respectively.

Table 4
Comparison of the false negative rate (FNR) and false positive rate (FPR) according to different types of attacks

Attack type	Kim		STDM		DL-STDM	
	FNR	FPR	FNR	FPR	FNR	FPR
1. Noise 0.2%	0.15	0	0.22	0	0.09	0
2. Noise 1%	0.24	0.001	0.33	0	0.15	0.001
3. Cropping 10%	0.10	0	0.18	0	0.03	0
4. Smoothing Lee's method	0.16	0	0.23	0	0.04	0
5. Smoothing Euler angles	0.04	0	0.08	0	0.03	0.001
6. Average smoothing	0.30	0	0.38	0	0.01	0
7. Enhancement 1.5	0.24	0.001	0.35	0	0.21	0
8. Attenuation 0.8	0.24	0	0.36	0	0.13	0
9. Warping 5%	0.07	0	0.11	0	0.02	0
10. Warping 10%	0.22	0	0.22	0	0.04	0
11. Uniform scaling 1.05	0.23	0	0.29	0	0.20	0
12. NUSA	0.70	0	0.49	0	0.19	0
13. Uniform time warping	0.05	0	0.14	0	0.04	0
14. Non-uniform time warping	0.10	0	0.15	0	0.08	0
15. Simplifying	0.08	0	0.24	0	0.07	0
16. 2nd Watermark	0.02	0	0.13	0	0.01	0

Thousand samples were tested.

same as the original watermark. The detailed parameter setting is described in Section 5.1.

Noise: Rows 1–2 show the detection results under the addition attack of white noise. The percentage represents the noise amplitude as a fraction of the largest Euler angle of the motion.

Cropping: Row 3 demonstrates the resilience of the watermark under cropping attack. Ten percent of the central part is cut out. A registration process [4] is carried out for each cropped motion.

¹ We have not compared our algorithm with Yamazaki's [25] because their algorithm and Kim's algorithm are both based on Cox's method [8]. Yamazaki's algorithm achieves similar results as that of Kim's.

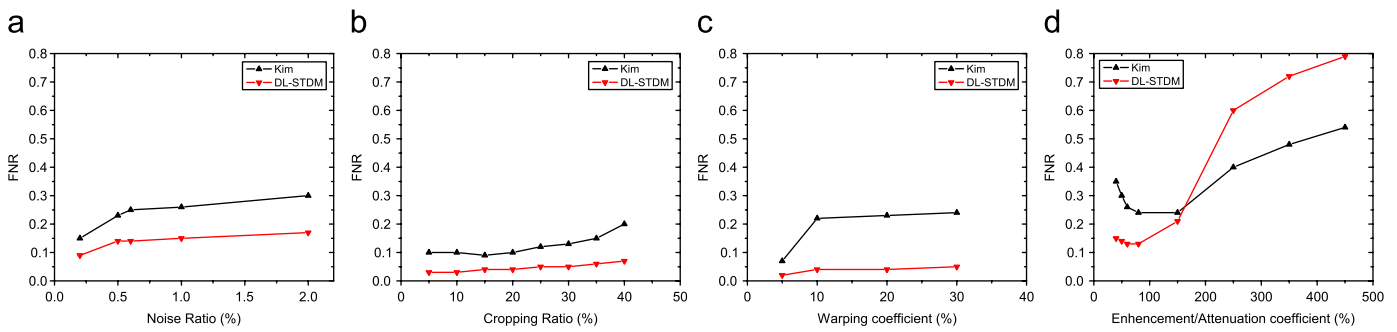


Fig. 6. The robustness of DL-STDM against four typical attacks with different scales. The results of Kim's method are shown as black lines, and ours as red lines. (a) Noise addition, (b) cropping, (c) motion warping and (d) enhancement and attenuation.

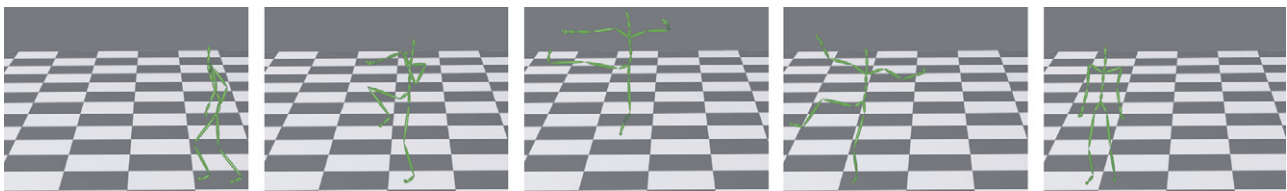


Fig. 7. The jump-kick motion we use for demonstrating signal changes after watermarking and different attacks that are illustrated in Fig. 8.

The robustness of the watermarking scheme depends on the registration process. Our experimental results show that the registration process works perfectly under cropping attack when the cut-off ratio is below 30%.

Smoothing: Rows 4–6 address three kinds of smoothing attacks. For row 4, we use the smoothing algorithm proposed by Lee et al. [4,31]. For row 5, we apply an average smoothing filter for Euler angles, the filter has the size of 1-by-3. For row 6, the smoothing process is carried out in quaternion domain by: $\mathcal{H}^A(\mathbf{q}_i) = (\mathbf{q}_{i-1} + \mathbf{q}_i + \mathbf{q}_{i+1}) / \|\mathbf{q}_{i-1} + \mathbf{q}_i + \mathbf{q}_{i+1}\|$.

Enhancement and attenuation: Rows 7–8 show the results of enhancement and attenuation attacks. For the enhancement attack, we multiply each of the detail coefficients at the coarsest level and its next finer level of the multiresolution representation by a constant factor of 1.5 [4,31]. For the attenuation attack, we use a constant factor of 0.8.

Motion warping: Rows 9–10 demonstrate the robustness against the motion warping [5,31] attack. We change the position of the *leftfoot* joint of the central frame, and solve the configurations of the related joints with inverse kinematics (IK). Then a multi-level B-spline fitting scheme [6] is utilized to propagate the changes to the nearest 20 frames.

Scaling along amplitude axis: Rows 11–12 address to two kinds of scaling attacks. For row 11, a uniform scaling for Euler angles is applied. The percentage represents the ratio of the scaled amplitude to the original amplitude. For row 12, a non-uniform scaling is carried out. The largest 10% and the smallest 10% of the Euler angles of each joint are scaled by 0.95, and the middle 20% part are scaled by 1.05.

Time warping: Rows 13–14 shows the results of time warping attacks. The attack of uniform scaling with a factor of 0.8 is presented in row 13, that of the non-uniform scaling is shown in row 14. A motion-signal alignment algorithm [4] is applied in order to align the time warped motion signal with the original one.

Simplifying: Row 15 demonstrates the results for a simplifying attack. In such an attack, the finest level and its nearest coarser level are eliminated.

2nd watermarking: Row 16 shows the robustness of our scheme under 2nd-watermarking attack. In such attack, a watermark is

inserted into a watermarked motion. The newly inserted watermark is different from that for the original motion.

Examples of the above attacks are illustrated in Fig. 8. In these examples, we use the signals of the *leftfoot* joint of the jump-kick motion shown in Fig. 7. The attacks are imposed on the watermarked signals (Fig. 8)).

The experimental data in Table 4 demonstrates that in 15 of 16 cases the proposed DL-STDM scheme outperforms Kim's method and the standard STDM scheme. There is a significant improvement for the NUSA attacks (row 12). The ability to resist NUSA attacks is important for motion data, because NUSA attacks can be carried out by motion warping which is a well-known motion editing method. As Kim's method is based on Cox's method, which extracts the watermark with correlation measurement, it cannot resist attacks of NUSA, since this kind of attack can destroy the correlation between the signals in which the watermark is embedded. However, DL-STDM can efficiently resolve this problem because it is based on STDM which has good tolerance against NUSA attack. The double-layer structure we designed further improves the robustness.

Table 4 only gives a subset of the experimental results. More results for the attack of noise addition, enhancement and attenuation, motion warping and cropping are illustrated in Fig. 6. These results further demonstrate that the proposed DL-STDM is more robust than Kim's against the above attacks, except for enhancement attacks that are relatively large. When the enhancement attack is larger than 210%, the performance of DL-STDM decreases to be worse than Kim's approach. This is because in Kim's approach, the watermark is embedded into the multi-resolution coefficients, and is detected with a student's *t*-test that relies on the similarity between the detected and the original watermarks. Our approach, however, needs to detect the "exact" watermark. In our approach, the watermark is embedded into the DCT domain of the QLI. When the enhancement attack destroys the motion too much, the DCT coefficients are heavily influenced.

The FPR of the DL-STDM scheme is illustrated in Table 2. The FPR is very low when the dither parameter δ_a is below 3.5.

We have also tested the distortion that our scheme will impose on the original motion. We use a maximum change ratio r_{\max} and root-mean-square error (RMSE) for evaluating the changes on the

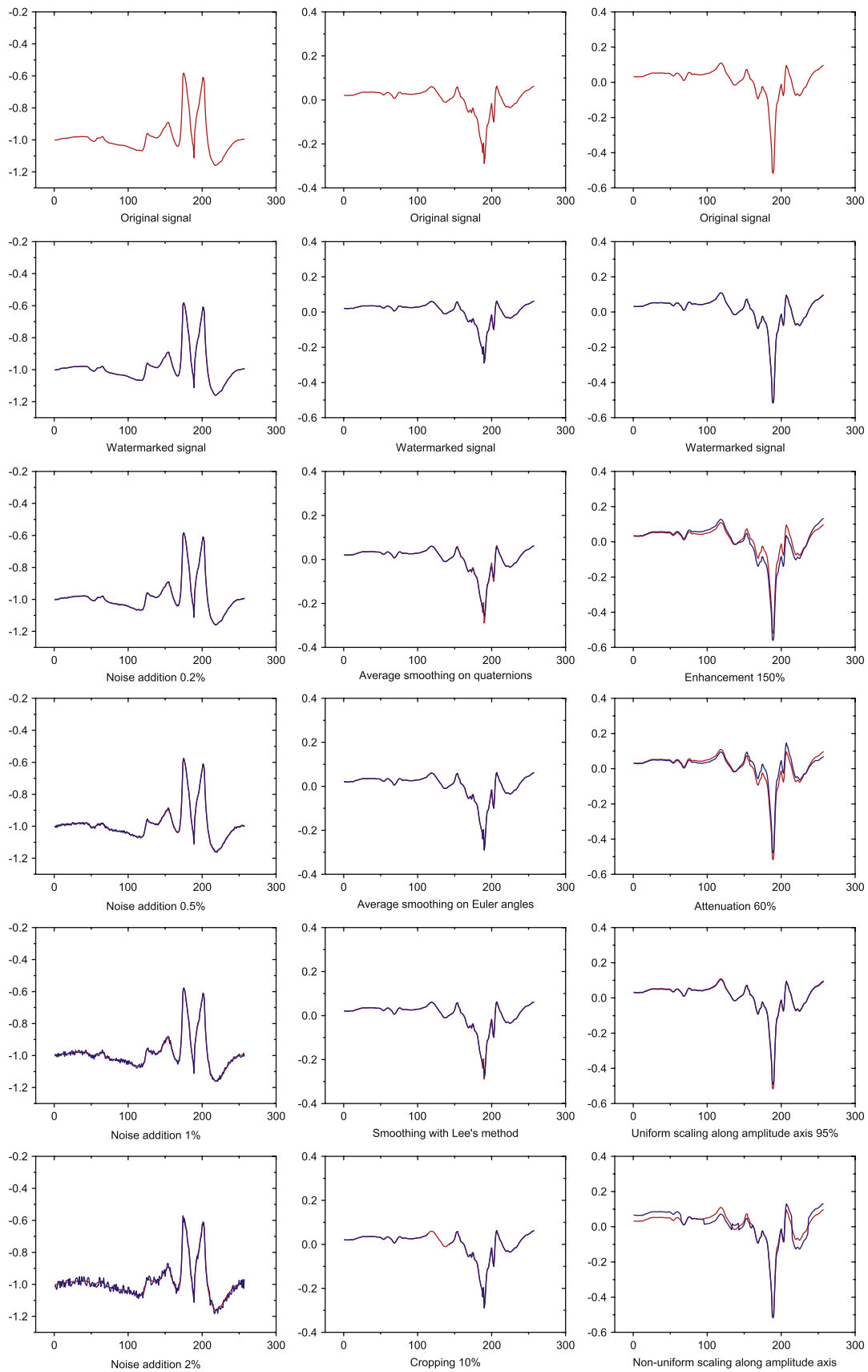


Fig. 8. Original and watermarked motion signals (top two rows), and the signal after various attacks. The original signal is shown in red, and the watermarked and attacked signals are shown in blue. The left, middle and right columns demonstrate the x, y and z components of the quaternion logarithm of the *leftfoot* joint of the motion shown in Fig. 7, respectively.

Table 5
The false negative rate of DL-STDM with different number of frames

Attack	Number of frames						
	50	100	200	400	600	800	1000
Noise 0.2%	0.20	0.14	0.13	0.11	0.10	0.09	0.08
Smooth Lee	0.14	0.07	0.06	0.05	0.04	0.03	0.03
Enhancement 1.5	0.32	0.25	0.23	0.22	0.21	0.20	0.19
Warping 10%	0.13	0.06	0.05	0.05	0.04	0.04	0.03
NUSA	0.30	0.23	0.21	0.20	0.19	0.18	0.18
Simplifying	0.19	0.14	0.12	0.09	0.08	0.07	0.07

Table 6
Running times of watermark embedding and extracting processes for six motions

Number of frames	Watermark embedding (ms)		Watermark detection (ms)	
	Kim	DL-STDM	Kim	DL-STDM
	129	195	90	132
257	347	193	314	72
513	762	458	522	149
1025	1601	1147	796	262
2049	3298	2450	1245	673
4097	9389	5542	3527	1714

quaternion logarithmic domain. The maximum change ratio is defined as: $r_{\max} = \max\{|\mathbf{p}_i - \mathbf{p}'_i|\}$. And the RMSE is defined as $e = \sqrt{(\sum_{i=1}^n |\mathbf{p}_i - \mathbf{p}'_i|^2)/n}$. Here, \mathbf{p} and \mathbf{p}' represent the joint positions of the original motion and that of the watermarked motion, respectively. n is equal to the number of joints multiplied by the number of frames. The test results are listed in Table 3. Both Kim's method and ours give negligible modification to the original motion.

To investigate the influence of the number of frames, we test six types of attacks with 1000 motions. The results are listed in Table 5. The more frames the motion has, the more robustness the DL-STDM will be.

The running time for five motion clips is listed in Table 6. Experimental results show that our approach runs faster than Kim's method. This is partially because we use Intel[®] integrated performance primitives 5.0 to calculate the DCT transformation for the DL-STDM. The running time of both Kim's approach and the proposed DL-STDM algorithm increase linearly with the number of frames. The time for the DL-STDM includes the time for the occurrence layer plus that of the authentication layer. The experiments were done on a desktop PC with P4 1.8 GHz CPU and 768M physical memory. The test program was developed in Visual Studio.net 2003.

6. Discussion and conclusion

Robustness of DL-STDM : Robustness requirements vary with application. The proposed DL-STDM scheme is robust against a wide variety of attacks listed in Table 4, but is not guaranteed to be robust against all kinds of attacks.

Weakness of DL-STDM : DL-STDM is designed mainly for authentication purposes, it is not as general as Kim's method. Another weakness is that extra data needs to be saved.

Cut-off positions for block coefficients: We have tested various motions presented in CMU motion database, and find that the

middle parts of the ordered DCT-coefficient blocks are suitable for watermarking. The test results also prove that the cut-off position that is specified by the vertical component of the normalized tangent vectors works well for watermarking motion data.

Amount of storage for extra data: Some extra data need to be stored for the DL-STDM scheme, which makes the DL-STDM scheme as a non-blind watermarking scheme. These data include: the index matrices \mathcal{I}_o and \mathcal{I}_a , the dithers \mathcal{D}_o and \mathcal{D}_a , and the random matrices \mathcal{U}_o and \mathcal{U}_a . These data can be stored in the database of the owner of the motion data. \mathcal{I}_o , \mathcal{I}_a and \mathcal{U}_o , \mathcal{U}_a have the same maximum size of $3n(k-1)$, and \mathcal{D}_o and \mathcal{D}_a have the same maximal size of $2l$. Suppose each integer or floating number occupies 4 bytes, then the amount of storage for them is $2 \times [3n(k-1) + 2l + 3n(k-1)] \times 4 = 48n(k-1) + 16l$, where l is the length of the watermark, n is the number of frames and k is the number of joints. Suppose we have a motion with 1000 frames and 30 joints. If the length of the watermark of the authentication layer is 64, the maximal amount of storage is approximately 1.4 MB.

To conclude, this paper addresses the problem of robustly watermarking motion data. We propose the QLI representation for motion data and present a practical scheme called DL-STDM based on this representation. Our scheme is proven to be more robust than Kim's against various attacks, including noise addition, smoothing, cropping, motion warping, simplifying, attenuation, 2nd watermarking and NUSA.

Our plans for future research include testing different possible attacks, especially combinations of the tested attack types. Since the DL-STDM scheme is still non-blind, designing a blind yet robust watermarking scheme for motion data is an open problem. Finally, the proposed scheme may have potential applications for watermarking of other data types such as mesh, audio and videos.

Acknowledgements

We thank Ke-sen Huang and Professor Ingemar J. Cox's for their helpful suggestions at the beginning of this research. We are extremely grateful to all the reviewers whose tough and detailed comments on manuscripts greatly contribute to the quality of the paper. Thank Jamie Wither and Midori Hyndman for proof reading. Our work was supported by the National 973 Plan (Grant no. 2006CB303105), the National Natural Science Foundation of China (Grant no. 60573147) and partially supported by Project ANR-KAMELEON of France. 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] House of Moves Inc. (<http://www.moves.com/stockdata.htm>).
- [2] Animazoo UK Ltd. (<http://www.animazoo.com>).
- [3] CMU graphics lab motion capture database. (<http://mocap.cs.cmu.edu>).
- [4] Kim TH, Lee J, Shin SY. Robust motion watermarking based on multiresolution analysis. *Computer Graphics Forum* 2000;19(3):189–98.
- [5] Witkin AP, Popovic Z. Motion warping. In: *Proceedings of SIGGRAPH '95*, 1995. p. 105–8.
- [6] Lee J, Shin SY. A hierarchical approach to interactive motion editing for human-like figures. In: *Proceedings of SIGGRAPH '99*, 1999. p. 39–48.
- [7] Bruderlin A, Williams L. Motion signal processing. In: *Proceedings of SIGGRAPH'95*, 1995. p. 97–104.
- [8] Cox JJ, Kilian J, Leighton T, Shamoon T. Secure spread spectrum watermarking for multimedia. *IEEE Transactions on Image Processing* 1997;6(12):1673–87.
- [9] Chen B, Wornell GW. Quantization index modulation: a class of provably good methods for digital watermarking and information embedding. *IEEE Transactions on Information Theory* 2001;47(4):1423–43.
- [10] Tanaka K, Nakamura Y, Matsui K. Embedding secret information into a dithered multi-level image. In: *1990 IEEE military communications conference*, 1990. p. 216–20.
- [11] Swanson M, Zhu B, Tewfik A. Data hiding for video-in-video. In: *1997 IEEE international conference on communications, vol. 2*, 1997. p. 676–79.

- [12] Benedens O. Geometry-based watermarking of 3D models. *IEEE Computer Graphics and Applications* 1999;19(1):46–55.
- [13] Benedens O, Busch C. Towards blind detection of robust watermarks in polygonal models. *Computer Graphics Forum* 2000;19(3).
- [14] Zafeiriou S, Tefas A, Pitas I. Blind robust watermarking schemes for copyright protection of 3D mesh objects. *IEEE Transaction on Visualization and Computer Graphics* 2005;11(5):596–607.
- [15] Bors AG. Watermarking mesh-based representations of 3-D objects using local moments. *IEEE Transactions on Image Processing* 2006;15(3):687–701.
- [16] Praun E, Hoppe H, Finkelstein A. Robust mesh watermarking. In: *Proceedings of SIGGRAPH '99*, 1999. p. 49–56.
- [17] Yin K, Pan Z, Shi J, Zhang D. Robust mesh watermarking based on multiresolution processing. *Computers & Graphics* 2001;25(3):409–20.
- [18] Ohbuchi R, Mukaiyama A, Takahashi S. A frequency-domain approach to watermarking 3D shapes. *Computer Graphics Forum* 2002;21(3).
- [19] Uccheddu F, Corsini M, Barni M. Wavelet-based blind watermarking of 3d models. In: *Proceedings of the 2004 multimedia and security workshop on multimedia and security*, 2004. p. 143–54.
- [20] Li L, Zhang D, Pan Z, Shi J, Zhou K, Ye K. Watermarking 3D mesh by spherical parameterization. *Computers & Graphics* 2004;28(6):981–9.
- [21] Yeo B-L, Yeung MM. Watermarking 3D objects for verification. *IEEE Computer Graphics and Applications* 1999;19(1):36–45.
- [22] Wang C-M, Cheng Y-M. An efficient information hiding algorithm for polygon models. *Computer Graphics Forum* 2005;24(3):591–600.
- [23] Lin H-YS, Liao H-YM, Lu C-S, Lin J-C. Fragile watermarking for authenticating 3-d polygonal meshes. *IEEE Transactions on Multimedia* 2005;7(6):997–1006.
- [24] Wu H-T, Cheung Y-M. A fragile watermarking scheme for 3D meshes. In: *ACM multimedia and security workshop*, vol. 7, 2005. p. 117–23.
- [25] Yamazaki S. Watermarking motion data. In: *Proceedings of the pacific rim workshop on digital steganography (STEG04)*, 2004. p. 177–85.
- [26] Agarwal P, Adi K, Prabhakaran B. Robust blind watermarking mechanism for motion data streams. In: *Proceeding of the 8th workshop on multimedia and security*, 2006. p. 230–35.
- [27] Meredith M, Maddock S. Motion capture file formats explained. (<http://www.dcs.shef.ac.uk/mikem/fileformats/mocapff.pdf>).
- [28] Park SI, Shin HJ, Shin SY. On-line locomotion generation based on motion blending. In: *SCA '02: Proceedings of the 2002 ACM SIGGRAPH/Eurographics symposium on computer animation*, 2002. p. 105–11.
- [29] Lee J, Shin SY. General construction of time-domain filters for orientation data. *IEEE Transaction on Visualization and Computer Graphics* 2002;8(2).
- [30] Dam EB, Koch M, Lillholm M. Quaternions, interpolation and animation. Technical Report DIKU-TR-98/5:87–105, University of Copenhagen; 2001.
- [31] Lee J, Shin SY. A coordinate-invariant approach to multiresolution motion analysis. *Graphical Models* 2001;63(2):87–105.
- [32] Gleicher M. Retargeting motion to new characters. In: *Proceedings of SIGGRAPH '98*, 1998. p. 33–42.
- [33] Lin S, Costello DJ. Error control coding: fundamentals and applications. Englewood Cliffs, NJ: Prentice-Hall; 1982.