

# Robust Blind Watermarking Mechanism For Motion Data Streams

Parag Agarwal      Ketaki Adi      B. Prabhakaran  
Department of Computer Science, University of Texas at Dallas

MS EC 31, PO Box 830688, Richardson, TX 75083

Email {parag.agarwal, ketaki.adi}@student.utdallas.edu, praba@utdallas.edu

## ABSTRACT

The commercial reuse of 3D motion capture (Mocap) data in animation and life sciences raises issues with respect to its copyright. In order to improvise content protection of Mocap data, we devise a substitutive blind watermarking technique. This technique visualizes 3D Mocap data as a series of non-intersecting cluster of triangles. Bits are encoded inside the triangles by using an extended substitutive bit encoder in spatial domain. The encoding supports watermark imperceptibility and develops robustness against affine transforms (rotation, translation, scaling), noise addition, reordering and sample loss attacks. Security of the scheme can be enhanced by adding secret embedding distances between clusters, which are based on a secret key used for watermarking purpose.

## Categories and Subject Descriptors

H. 0 [Information Systems] General, Security, Copyright

## General Terms

Algorithms, Security

## Keywords

Watermarking, imperceptibility, blind, spatial, encoding, decoding

## 1. INTRODUCTION

The advent of Motion Capture systems such as Vicon [12] has brought in applications like animation (games, films & TV, education), and life sciences (biomechanical research, gait analysis, rehabilitation, posture, balance and motion control, sports performance). The above applications deal with motion analysis or reusability, and can benefit from having a large repository of 3D human motions. In cases where data is stolen, we incur losses in terms of effort, time and money. Theft of motion data can lead to copyright issues, and digital watermarking can be a potential solution to these problems.

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

*MM&Sec'06*, September 26–27, 2006, Geneva, Switzerland.

Copyright 2006 ACM 1-59593-493-6/06/0009...\$5.00.

3D Mocap data streams consist of motion information related to human joints, and can be represented by their translational and/or rotational information. The process of encoding and decoding motion data samples poses problems related to watermarking scheme properties, such as imperceptibility of the watermark, and robustness against data dependent attacks. This can be analyzed as follows:

**1) Distortions in meaning of data:** Addition of watermarks distorts the original data. This changes the meaning of the data set. The visible distortions due to change in the meaning of motion data, will fail the imperceptibility criteria of a watermarking scheme.

**2) Watermark loss due to motion editing:** Motion data can be subject to motion editing operations done intentionally (adversary) or unintentionally (user – an artist) by using motion editing software such as Motion Builder [1]. Editing operations result in loss of the watermark, which defeats the purpose of robustness. The motion data editing attacks can be categorized as follows:

- **Sample Loss:** Operations such as cropping and curve simplification can result in loss of data elements that were associated with the watermark. Cropping can be visualized as selection of sub-part of the data. The cropped data might contain only a sub-part of the watermark or no watermark in the worst case, resulting in loss of the watermark.
- **Value change of samples:** Since watermark is associated with the data set values, any change in values of the motion data set alters the watermark. Value changes can take place as result of additive noise, time scaling, smoothing, affine transforms (translation, scaling, and rotation), and format conversions (global to local format in Vicon IQ [12]). Time scaling involves scaling and format conversion resulting in change in co-ordinate systems, and thus can be categorized under affine transformations.
- **Reordering of samples:** The order of samples can be altered. For example, if we reverse the order of time series data, the action represented by the motion will change in direction. Since the decoding process is order dependent, a change in the order creates problems in the decoding process.

It has been observed [6, and 9] that imperceptibility, and robustness are in an inverse relation to each other. Replication is a mechanism of achieving robustness. However, replication can induce more distortions resulting in loss of meaning of the data. Therefore achieving both at the same time is difficult.

The amount of information required to verify the presence of watermark should not depend on the size of the data set. This would help the verification information stay minimal in cases

where the motion capture data set increases by large amounts. Such storage efficiency can be achieved by making the scheme blind, which requires only secret and watermark for verification purpose. Finding a storage efficient scheme that achieves robustness with minimal distortions is a challenge.

### 1.1 Proposed Approach and Contributions

The paper proposes a mechanism to blind watermark 3D time series motion data by visualizing time series data as a cluster of triangles separated by secret random distances. Watermarks are hidden inside the triangles by using extensions of a bit encoding process [5]. The contributions of the paper are listed as follows:

- **Reduction in distortions:** The encoding methodology results in similarity of watermarked and original motions. As a result, the technique achieves imperceptibility of the watermark.
- **Robustness against motion editing attacks:** The watermarking mechanism is shown to be robust against attacks based on editing operations on motion data, such as affine transforms (rotation, scaling, and translation), reordering of samples, sample loss, noise addition, and format conversion.
- **Storage efficiency:** The nature of the scheme is blind which results in storage efficiency, since for verification since it needs only a secret key and the watermark.

### 1.2 Related Work

Different blind watermarking schemes have been developed for multimedia data such as images, audio, video and 3D mesh objects. These can be categorized as spatial or transform domain techniques. Spatial techniques [6, and 9] operate on the data set without changing it to any other domain, as compared to transform domain techniques [6, 7, and 9] that use spread spectrum.

Techniques developed for 1D audio, 3D mesh objects, 2D images, and video are not generalized to handle 3D motion data. Previous work on watermarking motion data [10, and 11] are informed mechanisms, which require original data sets for verification purpose. As motion data files increase, this would result in information explosion, making the method storage inefficient.

The technique proposed in this paper is spatial in nature, and is storage efficient. In addition, the spatial technique has high replication (i.e. more robustness against attacks), and it tries to minimize the distortion to achieve watermark imperceptibility.

## 2. SCHEME DESIGN

Human body comprises of 19 joints (see Figure 1), and their motion can be captured using motion capture technology. Figure 2 shows an example motion of the hand joint represented by positional data (x, y, and z) that can be represented in logical time. This information varies in *logical time* (row or frames or samples ordered sequentially). The varying joint positional information represents the 3D motion data streams. We can represent this 3D motion data as a time series data, which can be given in a matrix (see Figure 3), where the motion of 19 joints of the human body is given by its varying positional information. Only positional information is considered for watermarking as angular information is dependent on the same. We propose to watermark

each joint separately, since it provides robustness against joint cropping attacks and motion editing operations where a particular body part may be selected.

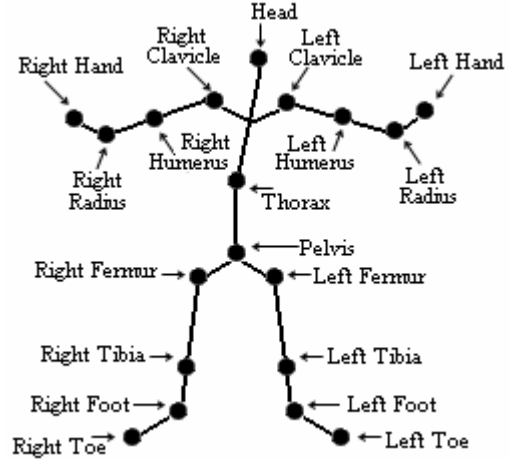


Figure 1. Human Body Representation with 19 joints

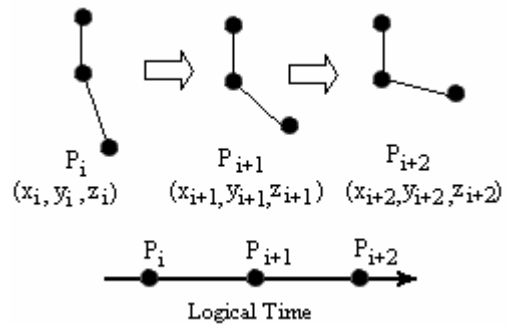


Figure 2. Motion Representation of Hand Joint

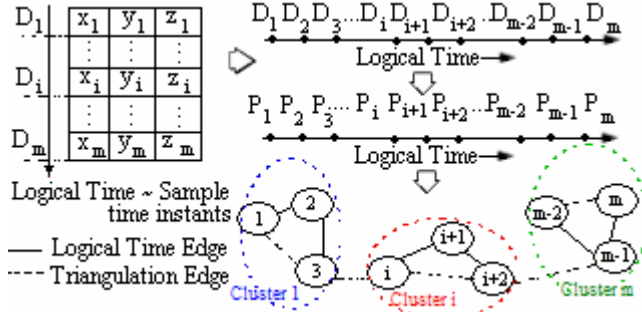
	Joint 1			Joint 2.....			Joint 19			
Logical time	$x_{1,1}$	$y_{1,1}$	$z_{1,1}$	$x_{2,1}$	$y_{2,1}$	$z_{2,1}$	.....	$x_{19,1}$	$y_{19,1}$	$z_{19,1}$
	...	...	...	...	...	...	.....	...	...	...
	$x_{1,i}$	$y_{1,i}$	$z_{1,i}$	$x_{2,i}$	$y_{2,i}$	$z_{2,i}$	.....	$x_{19,i}$	$y_{19,i}$	$z_{19,i}$
	...	...	...	...	...	...	.....	...	...	...
	$x_{1,m}$	$y_{1,m}$	$z_{1,m}$	$x_{2,m}$	$y_{2,m}$	$z_{2,m}$	.....	$x_{19,m}$	$y_{19,m}$	$z_{19,m}$

Figure 3. Matrix representation of 3D Joint motion data

Encoding inside each joint is done by visualizing it as a 3D triangle mesh as described below:

**Triangle Mesh Transform:** 3D joint data can be represented in matrix as  $(D_m \times 3) = [D_i]^T$ ,  $1 \leq i \leq m$ , where sample data-set  $D_i = \langle X, Y, Z \rangle$  and  $m$  – number of samples, satisfying the following properties 1)  $X_i_p \rightarrow X_i_q$  ( $\rightarrow$  ‘causality’), ( $p < q$ ), 2)  $X_i_p$  is correlated to  $X_i_q$ . It can be observed that  $D_i$  can be represented as point ( $P_i$ ) in 3D space (see Figure 4). Since  $D_i$  (s) are already totally ordered with respect to time, we can say that points  $P_i$  and  $P_j$  ( $i < j$ ) are adjacent to each other. As a result, the points in space can be converted into a 3D mesh, by joining  $P_i$  and  $P_{i+1}$ , where the order of occurrence of points is defined. Since the angle subtended between  $P_i$ ,  $P_{i+1}$  and  $P_{i+2}$  is not necessarily  $180^\circ$ , we

can join  $P_i$  and  $P_{i+2}$  using an edge, resulting in triangulation of the entire 3D time-series data. Similar edges can be obtained between points  $P_j$  and  $P_k$  ( $j > k$ ).

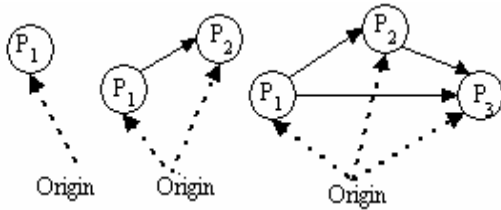


**Figure 4. Visualization of 3D motion data as triangle-mesh**

After transforming the 3D Mocap data into a 3D triangular mesh, we identify a series of non-intersecting clusters of triangles that can be watermarked using cluster based encoding. The bit information is embedded by identifying vertices of the triangle, whose displacement (warping) results in bit encoding. The decoding process is similar to encoding, and is discussed in Section 3. The following Sub-sections explain the idea in detail.

## 2.1 Cluster Based Encoding

As discussed above, clusters of points are used to encode data sets. These clusters are non-intersecting to localize the problem of robustness. These clusters are visualized as 3D triangular mesh, and are suited for affine transform invariant encoding. The following discussion gives the rationale behind the scheme.



**Figure 5. Finding Minimum points to encode**

**Finding minimal points for affine invariant encoding:** In order to encode bit information inside a data set, we need to identify affine transform invariant scalar quantities ( $q_1, q_2$ ), where  $q_1$  and  $q_2$  can be derived from a subset of the data set elements. Equations (1) and (2) represent the bit information using these scalar quantities.

$$\text{Bit}(q_1, q_2) = 0, q_1 \leq q_2 \quad (1)$$

$$\text{Bit}(q_1, q_2) = 1, q_1 > q_2 \quad (2)$$

It can be observed that ratio  $q_1/q_2$  must be invariant to affine transformations. We need to find the minimum number of points that can be used to encode the bit information. A single point  $P_1$  in 3D space can be visualized as a vector (see Figure 5) which results in only one scalar – magnitude, which is variant to affine transformation. Considering two points ( $P_1$  and  $P_2$ ) we get a new vector that is a resultant of the two points. The scalar corresponding to this vector is its magnitude, which cannot be

compared to itself. When considering three points ( $P_1, P_2$  and  $P_3$ ), we can have two vectors such that one point is common among these vectors. The magnitudes of these vectors give us scalars, whose ratio stays invariant to affine transformations. Points ( $P_1, P_2$  and  $P_3$ ) form a triangle or can even lie on a straight line. A scheme to embed information can be derived using scalar quantities, such as the Euclidian distances between these points.

**Bounds on Cluster size:** From the above observation we see that three points can be used to encode bit information, hence the minimum number of points necessary from the data set to form the cluster is  $CSize_{low} = 3$ . The number points used to encode can be increased by using a cluster of larger size. Therefore, the upper bound on the cluster size is  $CSize_{upper} = m$ , where ‘ $m$ ’ is total number of points in the data-set.

**Robustness against reordering:** A sequence of bits is stored in clusters by choosing a set of points. This sequence must be invariant to change in order of points i.e. sequence of bits stored must be invariant to point ordering attacks. To handle this problem, equations (3) and (4) need to be satisfied.

$$\text{Invariant set: } \{P_i, P_j\} = \text{function}(\text{cluster } C) \quad (3)$$

$$P_i \prec P_j: P_i \neq P_j, \prec \text{ is a relation, } P_i \text{ and } P_j \in C - \{P_i, P_j\} \quad (4)$$

The invariant set (see equation 3) based points ( $P_1$  and  $P_3$  in Figure 5) can be chosen on the basis of a heuristic, which is a function of the cluster ‘ $C$ ’. For encoding purpose, we select ‘encoding’ point(s) ( $P_2$  in Figure 5) in the order as given by the equation (4). After encoding we must make sure that equations (3) and (4) are still satisfied, so that during decoding process we get the same sequence of bits that were encoded. The choice of the function ‘function’ and relation ‘ $\prec$ ’ determines the ability to withstand reordering attacks.

**Handling Pathological Cases:** In cases with a contention between points for invariant set (see equation 3), we warp some points to make sure only one pair is chosen. For example, the heuristic identifies the invariant set on the basis of maximum distant pair of points. In such a case, we can visualize contention in a cluster with size = 3, such that the points form an equilateral triangle or an isosceles triangle. There can be cases where points in the data set have same values i.e. points  $P_i$  and  $P_j$  belong to the same cluster and have equal values. For such cases, we have to choose among these points for encoding, which may result in contention. This situation can be avoided either by excluding these points from the encoding process or by perturbing them to achieve an encoding.

The encoding inside clusters is done using a bit encoding scheme as described in the following Sub-section.

### 2.1.1 Bit Encoding Scheme

The bit encoding scheme for three points is implemented by using a substitutive modulation technique. We extend the idea of macro embedding procedure (MEP), proposed in [5] for 3D triangles. MEP has been applied in [3, and 4], and is based on minimizing distortion using quantization index modulation [2].

The MEP is based on the idea of treating a triangle as a 2 state object (number of bits to be encoded = number of states). As

shown in Figure 6, we take three points ( $P_1$ ,  $P_2$  and  $P_3$ ) that form a triangle, and choose the side ( $P_1, P_3$ ) = ' $a$ ', and divide the side into equally spaced ' $2p$ ' intervals, where ' $p$ ' is the order of MEP. The intervals can be numbered as 0 or 1. The interval set of 0(s) is identified by  $S_0$  and 1(s) by  $S_1$ . In order to embed the bit inside the triangle, the following rules [5] can be observed:

Let  $Pos(P_2)$ : position of perpendicular from  $P_2$  on side ' $a$ '. In order to encode bit  $i$ , where  $i \in \{0, 1\}$ ,

- $Pos(P_2) \in S_i$ : No modifications required
- $Pos(P_2) \notin S_i$ :  $Pos(P_2)$  has to be shifted to a  $Pos(P_2')$  so that  $Pos(P_2') \in S_i$

The order ( $\geq 1$ ) of the MEP scheme can control and minimize the distortion [2, and 5] produced. More bits can be encoded by changing the interval labeling that supports  $2^s$  states ( $s = l$  in Figure 6). This increases the hiding capacity, which eventually contributes to robustness by supporting replication of the watermark. However, this increases the distortion [5] since point  $P_2$  has to be displaced more (*number of states*  $> 2$ ).

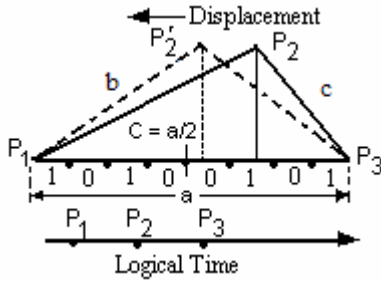


Figure 6. MEP applied to triangle of points  $P_1, P_2, P_3$

We observe that for MEP, the ratio of the length of projected point  $P_2$  on side ' $a$ ' to the length of side ' $a$ ' is invariant to affine transformations. Thus, it satisfies equation (1) and (2) and can be used to blind watermark using minimal number of points.

### 2.1.1.1 Extensions to MEP

We extend MEP to handle the following cases for our scheme.

**1) Line case:** If points  $P_1, P_2$  and  $P_3$  lie on a line, largest side ' $a = b+c$ ', where point  $P_2$  lies on the line joining  $P_1$  and  $P_3$ . We observe that the encoding rules do not change in such cases (as shown in Figure 7, we can handle the case by placing point  $P_2$  on side ' $a$ '). This case satisfies equations (1) and (2) since the ratio of  $a, b, c$  is invariant to affine transformation.

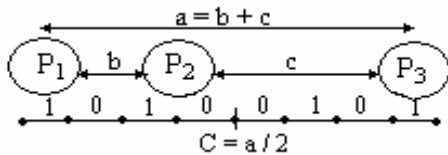


Figure 7. Points  $P_1, P_2,$  and  $P_3$  on a line

**2) Robustness against reordering:** If bits corresponding to each interval are labeled from the center in opposite directions (see Figure 6) starting with 0 or 1, we get a palindrome of 0s and 1s. The advantage of a palindrome scheme over the scheme suggested in MEP [5] is robustness against reordering attack. In case, the order of  $P_1, P_2$  and  $P_3$  is changed, the encoded bit information remains unaltered. This can be visualized from Figure 8, where the bit encoding is lost by changing the order of  $P_1$  and  $P_3$  in time

series data. However, as shown in Figure 9 the encoding does not change due to choice of the palindrome scheme. For our scheme, the numbers of intervals are  $2^{\text{order}}$ , where order  $\geq 2$ . Minimum order = 2, since we need at least 4 intervals to achieve a palindrome scheme.

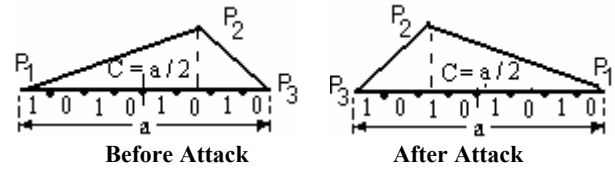


Figure 8. MEP [5]. Before reorder attack, for MEP [5] bit = 0, and after Attack bit = 1

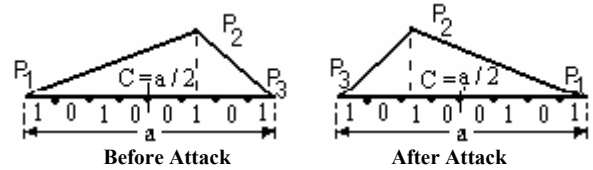


Figure 9. Extended MEP, no change in bit information

The following discussion gives the impact on the imperceptibility of the watermark by using the above scheme, and varying the cluster size.

**Impact on Imperceptibility (Distortions):** For cluster size = 3, only 1 point is altered. For a cluster of size ' $CSize$ ' the number of bits being encoded can be estimated on the basis of encoding points, since they are used to hide bit information. For our scheme since the number of invariant points is ' $2$ ', the number of encoding points is given by ' $CSize - 2$ '. As a result, larger size clusters can encode more bits by perturbing more number of encoding points. It can be noted that use of more points to encode implies more distortion, which harms the imperceptibility of the watermark. Therefore, the idea of having cluster size  $> 3$  impacts imperceptibility.

## 2.2 Embedding Distance – Security Enhancement

In cases where clusters are chosen in a known fashion 'sequentially', an adversary can attack the watermarked data by reordering of samples, down sampling, and noise addition. In order to handle this problem we use the concept of embedding distance. The clusters have to be chosen in a random fashion, where randomness is a secret. As a consequence, the clusters are separated by a random number of points or samples, termed as embedding distance. The use of random embedding distance reduces the chance of attacks, since the adversary finds it difficult to search for watermarks. For ' $n$ ' samples, the adversary's search space is  $2^n$ , which is huge if ' $n$ ' is large. By introducing embedding distance we may reduce the number of clusters that can be embedded. This implies that it impacts replication of watermark, and the robustness. Therefore, embedding distance is a price we might have to pay for robustness. In addition, decrease in number of clusters will result in less points being encoded, which will reduce the distortions as well.

### 3. IMPLEMENTATION AND SIMULATION

The watermarking scheme has been implemented in Matlab 7.0.4. For encoding and decoding purpose, we use the same secret key and watermark. The secret key corresponds to the private key of the user. The watermark was detected on the basis of detection ratio given as follows:

**Detection Ratio (DR):** This ratio (see equation 5) measures the similarity between original watermark  $W$ , and detected watermark  $W'$ . It is used during the detection process, and is measured on the basis of a threshold.

$$DR = \frac{\text{Number of Similar Bits}}{\text{Watermark Length}} \quad (5)$$

We always assume threshold for detection ratio is 1, in order to cover the worst case of analysis.

#### 3.1 Experiment Description and Metrics

The experiments were done on a motion matrix, which is a dance sequence with (4286 frames captured at 120 frames/sec for 19 joints = 81434 points/samples). The model is justified for performance analysis, since it consists of joints moving in varying directions at different time intervals, thus giving diversity in motion data. In order to analyze distortion (or similarity between original data  $M$  and watermarked data  $M'$ ), we subjected the file to 1000 random watermarks of size 10 bits. The number of bits encode per sample/point = 1. This measurement is taken as mean value for the following metrics.

**Similarity Metric [8]:** Based on singular value decomposition of matrix  $M$ , and  $M'$ , such  $u_1$  and  $v_1$  are first right singular vectors of SVD of  $M$  and  $M'$  respectively, with corresponding Eigen values -  $\gamma$  and  $\lambda$ , and with  $f = 0.9$  constant (see equation 6).

$$SM(M, M') = |u_1 \cdot v_1| \cdot \frac{\lambda \cdot \gamma - f}{1 - f} \quad (6)$$

**Signal to Noise Ratio (SNR)** can measure the distortion produced. Higher the distortion less imperceptible is the watermark. SNR for 3D motion data is calculated in equation (7) (RMS ~ root mean square).

$$SNR(M, M') = 20 \text{Log}_{10} \frac{RMS(M)}{RMS(M - M')} \quad (7)$$

During robustness analysis, we assume random attacks given as a percentage ('*attack percentage*') of samples perturbed, which are chosen randomly. The randomness covers the unpredictable nature of the adversary. Table 1 gives the list of attack used for experiments.

**Table 1. Attacks used for simulation purpose**

Nature	Attack type	Impact on sample
Random	Drop	Loss
Random	Affine	Value change
Random	Noise	Value change
Random	Smoothing	Value change
Random	reordering	Order change

Robustness is captured as mean value for 1000 iterations of attacks, using the following metrics:

**Failure Rate (FR):** This is measured as the ratio of number of failed attempts to the expected number of successful attempts to identify a watermark in a given data set. Higher the rate, less robust is the scheme against a given attack. This corresponds to the false negative probability.

**Bit Error Rate (BER):** This is measured as the ratio of number of bits flipped to the expected number of bits encoded in a given data set. Higher the rate, less robust is the scheme against a given attack.

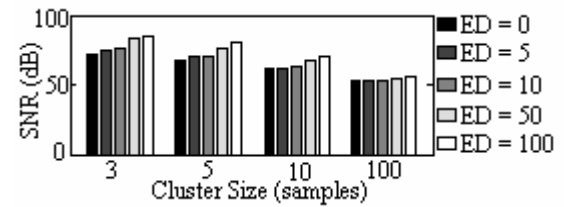
Sample losses are simulated as random sample drop attacks since dropping of samples results in loss of information. For smoothing attack we use an averaging filter of unit radius (see equation 8), as given in [11].

$$H(q(i)) = \frac{q(i-1) + q(i) + q(i+1)}{\|q(i-1) + q(i) + q(i+1)\|} \quad (8)$$

#### 3.2 Performance Analysis

The following gives a distortion analysis of the scheme:

**Distortion:** The impact analysis is done using a worst case, which is a smooth and less motion of a stable joint, the pelvis joint. A stable joint is one with less action, and is more vulnerable to distortions, as compared to a joint that have more motion. Figure 10 shows that the increase in cluster size results in more distortion (see Sub-section 2.1.1), since more points are altered for encoding purpose. By increasing the embedding distance (see Sub-section 2.2) the number of clusters being encoded decreases, resulting is less distortion i.e. higher SNR. In all the cases where distortion is measured according to SM (see equation 6), we found 100% similarity.



**Figure 10. Distortion (SNR) Vs Cluster size and embedding distance (ED), averaged over 1000 watermarks**

Robustness can be analyzed as follows:

**Failure Rate:** Overall failure rate (false negative probability) of the scheme can be visualized in Figure 11, which shows varying percentage of attacks and failure rates. The scheme is least robust against random sample drop attack, since loss of information is incurred when data set size is reduced, as compared to other attacks where data size does not change. The remaining attacks are done without reduction in size of the data set. The scheme shows the best performance against affine attack, due to its affine invariant embedding methodology. Robustness against reordering, smoothing and noise attack lies in between, and their order is evident – affine > reorder > smoothing > noise > sample drop. Thresholds can be identified on the basis of attack percentage till

which failure rate  $\sim$  (threshold value, 0 in our case). These thresholds determine the robustness to attacks (affine – 100%, reorder – 50%, smoothing – 30%, noise – 20% drop – (<) 1%). Attacks done at the observed thresholds will result in loss of meaning of data, which would make it worthless for the adversary. Therefore, the scheme is robust against attacks that do not alter the meaning of data.

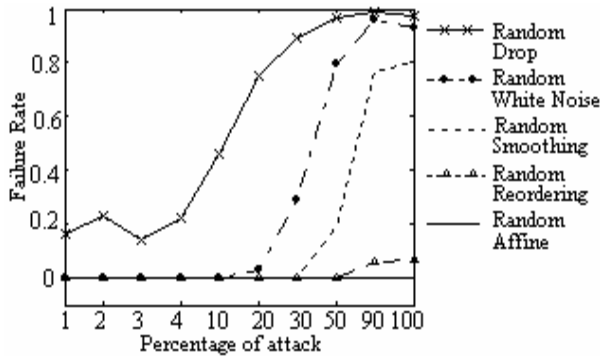


Figure 11. Failure Rate for varying attacks

**Robustness against reordering:** It can be observed from Table 2 that the bit error rate of MEP [5] is significantly large as compared to the extended MEP, for different percentage of reordering attack. Therefore, our scheme is more robust, as suggested in Sub-section 2.1.1.1

Table 2. Bit Error Rate, for varying reorder attack, MEP [5] Vs New MEP, MEP order = 4

Attack %	1%	2%	5%	7%	10%
New MEP	0	7.5E-4	6.6E-4	6.7E-4	8.7E-4
MEP [5]	0.0607	0.0865	0.0865	0.0901	0.0972

#### 4. CONCLUSION AND FUTURE WORK

The paper presents a robust blind watermarking technique for motion data streams. The watermark embedding procedure makes the watermarks imperceptible, with a 100% similarity measure and high signal to noise ratio (SNR). The scheme is shown to be robust against affine transforms (scaling, rotation, and translation), noise addition, reordering, smoothing and sample losses. Based on the experiments, the scheme was found to be robust against random attacks as follows: affine > reorder > smoothing > noise > sample loss (drop).

As a future initiative, we would investigate the hiding capacity provided by the scheme. In addition, we would like to see the possibility of multiple watermarking. Since the scheme was least robust against random loss attacks, further enhancement of the scheme is necessary.

#### 5. ACKNOWLEDGEMENTS

The work is supported in part by US Army Research Office grant 48645-MA and National Science Foundation under Grant No. 0237954 for the project CAREER: Animation Databases.

#### 6. REFERENCES

- [1] Motion Builder - AutoDesk, <http://www.autodesk.com>
- [2] B. Chen and G. W. Wornell, "Quantization index modulation: A class of provably good methods for digital watermarking and information embedding," *IEEE Trans. on Information Theory*, vol 47, pp. 1423-1443, May 2001
- [3] C. Wang, Y. Cheng, "An Efficient Information Hiding Algorithm for Polygon Models", *EuroGraphics 2005*, Volume 24 Page 591 - September 2005
- [4] F. Cayre, O. Devillers, F. Schmitt and H. Maître, *Watermarking 3D Triangle Meshes for Authentication and Integrity*, INRIA Research Report RR-5223, Jun. 2004
- [5] F. Cayre, B. Macq, "Data hiding on 3-D triangle meshes", *Signal Processing*, *IEEE Transactions on*, Volume: 51 Issue: 4 Apr 2003 , Page(s): 939- 949
- [6] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Mathew Miller, *Digital Watermarking: Principles & Practice* (The Morgan Kaufmann Series in Multimedia and Information Systems)
- [7] I. Cox, J. Kilian, F. Leigton, T. Shamoan, "Secure Spread Spectrum Watermarking for Multimedia," *IEEE Trans. Image Processing*, 1996, vol. 6, pp. 1673-1687. Lampert, L. *LaTeX User's Guide and Document Reference Manual*. Addison-Wesley, Reading, MA, 1986.
- [8] Li C, Peng Zhai, S. Q. Zheng and B. Prabhakaran, *Segmentation and Recognition of Multi-Attribute Motion Sequences*, *Proceedings of the ACM MM '04*, New York, NY USA, pp.836-843, October 2004.
- [9] Michael Arnold, Stephen D. Wolthusen, Martin Schmucker, 'Techniques and Applications of Digital Watermarking and Content Protection', Artech House Publishers (July 2003)
- [10] Shuntaro Yamazaki, "Watermarking Motion Data", In *Proc. Pacific Rim Workshop on Digital Steganography (STEG04)*, pp.177-185, Nov 2004
- [11] Tae-hoon Kim, Jehhee Lee, Sung yong Shin, *Robust Motion Watermarking based on Multiresolution Analysis*, *Computer Graphics Forum*, Vol. 19 No. 3, pp. 189-198, 2000 (*Proc. EUROGRAPHICS '2000*).
- [12] Vicon, <http://www.vicon.com>