SVD-BASED TAMPER PROOFING
OF MULTI-ATTRIBUTE MOTION DATA

Parag Agarwal   *Ketaki Adi*   *B. Prabhakaran*

Department of Computer Science,

The University of Texas at Dallas, TX 75083

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

ABSTRACT

Repositories of motion captured (MoCap) data can be reused for human motion analysis in physical medicine, biomechanics and animation related entertainment industry. MoCap data expressed as a matrix $M_{m \times n}$ can be subject to tampering from shuffling of its elements or change in element values due to motion editing operations. Tampering of archived motion data intentionally or due to machine/human errors, may result in loss of research, money and effort. The paper proposes singular value decomposition (SVD) based methodology for tamper proofing motion data. This tamper proofing methodology extracts reference patterns in the form of right and left singular vectors of motion data matrix M. These patterns are used to verify and trace the pattern of tampering. The use of first Eigen vectors for tamper detection reduces storage and computation complexities to $O\,(m + n)$ and makes the solution scalable.

1. INTRODUCTION

The advent of Motion Capture systems such as Vicon [16] has brought in applications like

- *Physical Medicine and Rehabilitation:* Analyzing different body segments/joints for different motions aid in better diagnosis of the problem(s) that a patient might be facing.
- *Biomechanics and Physiology*: Researchers investigating the interplay of bone and muscle in leg movement benefit from the 3D map of human body motions.
- *Reusability in Animation:* Motion captured data is reusable and it can help build entertainment related animations, by using software such as Motion Builder [8].
- *Quantifying the effects*: of certain diseases such as the effect of spasticity on knee movements.

These applications can benefit from having a large repository of 3D human motions. Motion data archived in a repository can be subject to tampering due to malicious actions or human/machine related faults. The tampering of motion data may result in loss, in terms of valuable information, money, and time spent for recording. Moreover, incorrect information can be misleading from the application's perspective.

MoCap data is multi-attribute, and can be described as $M_{m \times n}$ matrix (see Figure 1 and Figure 2) (for the comma separated value (*.csv*) format of Vicon IQ [16]), with columns representing rotational and positional data of skeleton joints, and rows representing the changing values over time. The varying lengths of frames make the data bulky. Adversaries can use motion editing techniques [16] such as motion cropping, mapping and concatenation to tamper data. These techniques alter the trajectory of joints by changing values or shuffling rows, columns, row elements, and column elements. There could be combination of these attacks or it could be a random attack. The tampering methodology should be capable of verifying and tracing the pattern of such kind of attacks.

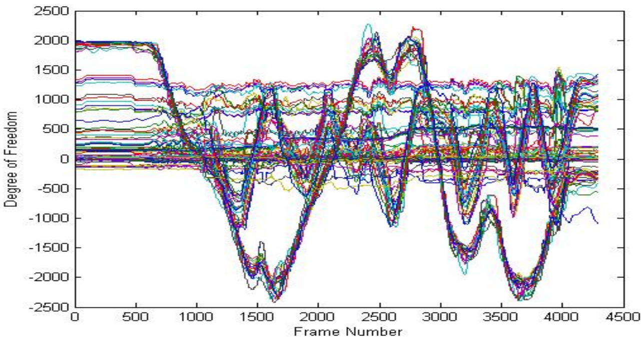
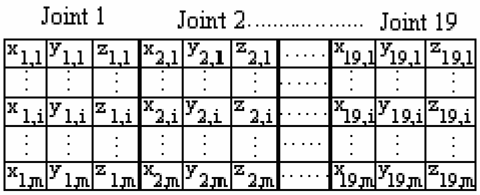


Figure 1. Motion Data representation



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

Figure 2. Motion Data Representation I matrix format with positional information

Tampering can be avoided by using tamper proofing mechanism, such as fragile watermarking. Fragile watermarking [2-13] can be achieved by embedding a watermark inside a target data. Tampering is recognized whenever during an extraction process, a sub-part of the embedded watermark is found corrupt. The sub-part points to the spatial location of corruption and serves as an evidence for tampering. Watermarking techniques alter the original data, resulting in distortion, which eventually can alter the meaning of the data. Recent research in watermarking motion data [14, and 15] uses private watermarking schemes for copyright protection. These schemes are private and are storage inefficient (not scalable), since the original data has to be stored for tamper verification. Therefore, in order to achieve tamper proofing, we need to design a novel scheme that can detect errors, without distorting the original data, and must be storage efficient.

1.1 Proposed Approach

The paper proposes singular value decomposition (SVD) based tamper proofing scheme. The scheme is not a watermarking methodology, as it does not incorporate information hiding. The idea is to extract the reference patterns using SVD, and use them for tamper proofing. The reference patterns are recognized as right and left Eigen vectors of the SVD of a motion data matrix *M*. During the detection process, we take the SVD of the target matrix *M'* and compare its left and right Eigen vectors with that of *M*. The proposed method is shown to verify and trace the pattern of attacks, such as row tampering, column tampering, row-column tampering, and random attack.

The reference patterns are Eigen vectors with non-zero Eigen values. This helps in reducing the size of information required to be stored for verification, and as a result makes the solution scalable. The use of Eigen vectors and thresholds help us determine the exact position in *M* that has been tampered. Experimental results aptly demonstrate the effectiveness of our approach.

The rest of the paper is organized as follows:
Section 2 discusses the attack patterns and methods developed using SVD to verify and trace these patterns. In addition, we suggest ways to optimize the methodology and advantages over watermarking are mentioned. Section 3 gives experimental proof of the attacks and helps visualize the advantage of the proposed technique. The paper ends with Sections 4 describing the future work and conclusion. Table 1, gives a list of notations used in the paper.

RT	Row tampering
RS	Row shuffling
RES	Row element shuffling
CT	Column tampering
CS	Column shuffling
CES	Column element shuffling
RCT	Row column tampering

Table 1. Table of Notations

2. TAMPER PROOFING METHODOLOGY

The tamper proofing methodology is applied on MoCap data (.csv format) acquired from Vicon IQ [16] (*120* frames/sec). As discussed earlier, this data can be expressed as a matrix $M_{m\ x\ n}$ ($m > n$), where columns represent the joints of the human skeleton. The joints are represented as rotational and translational information, with varying values per frame (row). The attacks on a matrix *M* can be categorized as follows:

- **Row tampering (RT) attacks:** This attack is restricted to row tampering only, such that column elements of *M* stay invariant. Mathematically, if *A* is modified to using row tampering then $A_{mxn} \neq B_{mxn}$ such that $\bigcup_{k=1}^{m} a\ (k,\ i) = \bigcup_{k=1}^{m} b\ (k,\ i)$, for all i ($1 \le i \le n$), and $\bigcup_{k=1}^{n} a\ (j,\ k) \neq \bigcup_{k=1}^{n} b\ (j,\ k)$, for j ($1 \le j \le m$). This attack can be realized either by column element shuffling (CES) or row shuffling (RS). CES exchanges the row elements, and does not alter the column element set. RS does not alter column elements, but shuffles the rows. These attacks are further categorized as combinations of {CES}, {RS}, {CES, RS}.
- **Column tampering (CT) attacks:** This attack is restricted to column tampering only, such that rows elements of *M* stay invariant. Mathematically, if *A* is modified to *B* using column tampering then $A_{mxn} \neq B_{mxn}$ such that $\bigcup_{k=1}^{n} a\ (i,\ k) = \bigcup_{k=1}^{n} b\ (i,\ k)$, for all i ($1 \le i \le m$), and $\bigcup_{k=1}^{m} a\ (k,\ j) \neq \bigcup_{k=1}^{m} b\ (k,\ j)$, for j ($1 \le j \le n$). This attack can be realized either by row element shuffling (RES) or column shuffling CS. RES exchanges the column elements, and does not alter the row element set. CS does not alter row elements, but shuffles the columns. These attacks are further categorized as combinations of {CS}, {RES}, {RES, CS}.
- **Combined row-column tampering (RCT) attacks:** This attacks results in tampering of row and column element set, such that element set of A and B are the

same. The tampering can be due to combinations of row and column tampering attacks.

- **Random Attacks:** This kind of attacks result by adding random noise signals to motion data. This attack is different from the above attacks, since the element set of *A* and *B* is not similar. In other words it is a combination of row and column tampering, such that element set of $A \neq$ element set of *B*.

The above attack patterns will result in joint motions following different patterns from their original. Since human body motion is described by motion information of joints, it will change as a consequence of the above attacks. The paper develops a singular value decomposition (SVD) based tamper proofing technique to handle such attacks.

2.1 Background on SVD

As proved in [1], any real $m \times n$ matrix *M* has a SVD (M) = $U.S.V^T$, where $U = [u_1, u_2 \ldots u_m]$ є $R_{m\times m}$ and $V = [v_1, v_2 \ldots v_n]$ є $R_{n\times n}$ are two orthogonal matrices, and S is a diagonal matrix with diagonal entries being the singular values of M: $s_1 \geq s_2 \geq \ldots \geq s_{\min(m,n)} \geq 0$, where s_1 is significantly larger than other Eigen values. Column vectors u_i and v_i are the i^{th} left and right singular vectors of *M* respectively. The left singular vectors have length equal to the number of time frames, that vary with each individual motion data file. The right singular vector has a constant length depending on the number of joints considered. The singular values of matrix *M* are unique, and the singular vectors corresponding to distinct singular values are uniquely determined up to the sign, or a singular vector can have opposite signs.

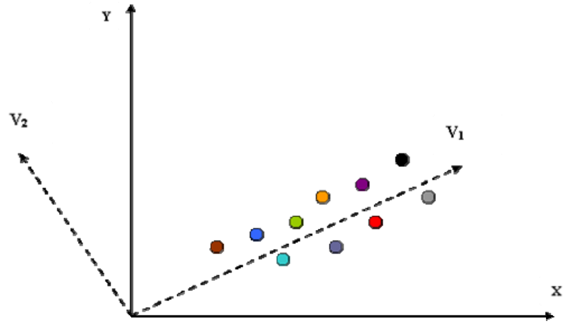


Figure 3. Geometric structure of a matrix exposed by its SVD

SVD exposes the geometric structure of a matrix *M*. It has orthogonal bases. It transforms the matrix from one vector space to another. The components of SVD quantify the resulting change between the underlying geometry of these spaces. Along the direction of the first right singular vector, the row vectors in *M* have the largest variation, and along the second right singular vector direction, the point variation is the second largest, and so on. The singular values reflect the variations along the corresponding right singular vectors. Figure 1 illustrates a 10 x 2 matrix (2D (x, y)) with first and second right singular vectors V_1 and V_2 (orthogonal vectors). The 10 points in the matrix have different variations along different directions; hence have the largest variation along V_1. We observe that elements of the matrix *M* can be expressed using a set of linear combination of elements of the matrices *U*, *V* and *S*.

The SVD (M) = $U.S.V^T = U.(S.V^T) = \{M(i, j)\}$ where $1 \leq i \leq m$, $1 \leq j \leq n$, $m > n$. $M(i, j)$ can be expressed as follows:

$M(i, j) = [U(i, 1)\ U(i, 2) \ldots U(i, m)]\ [S1.V(j, 1)\ S2.V(j, 2) \ldots Sn.V(j, n)\ 0\ 0 \ldots 0]^T$

The tamper proofing mechanism consists of *extraction phase* and *detection phase*. During the extraction phase reference pattern are extracted from matrix *M*. The second phase is the detection phase, where attacks are detected. We take SVD (M) = $U.S.V^T$, and for non-zero Eigen values *S* store the corresponding left and right Eigen vectors U and *V*. Here *V* and *U* are the reference patterns that will be used to identify and trace the attack pattern.

2.2 SVD Based Detection

Once the attack has been identified it traces the pattern used for the attack. The tamper detection process can be realized as follows:

```
Step1: SVD (M') = U'.S'.V'^T
Step2: DiffU = U' – U, DiffV = V' – V
Step3: if DiffU and DiffV are zero matrices
           No tampering;
         Else If (DiffV is zero matrix only)
           RS attack or CES attack or {RS, CES} attack;
         Else If (DiffU is zero matrix only)
           CS or RES or {CS, RES};
         Else
           Random attack or RCT;
```

In the above checking process we consider vector till *k*, since first *k* Eigen values are non-zero. The remaining *(n- k)* Eigen values are zero and their contribution to *M* is insignificant. The process of tamper detection is done by taking considering the difference (*DiffU* and *DiffV*) between Eigen vector matrix of *M* and *M'*.

The above steps can be understood as verification of attack and tracing the attack pattern. The following subsection discusses the reasons behind the verification and tracing of attacks:

2.2.1 Verification and tracing of Row Tampering (RT) attacks

The following theorem, an extension of the theorem mentioned in [1], helps us prove that row tampering is related to change in left Eigen vector only.

Theorem 1: Given matrix $A_{mxn} \neq B_{mxn}$ such that $\bigcup_{k=1}^{m} a(k, i) = \bigcup_{k=1}^{m} b(k, i)$, for all i ($1 \leq i \leq n$), and $\bigcup_{k=1}^{n} a(j, k) \neq \bigcup_{k=1}^{n} b$

(j, k), for j $(1 \leq j \leq m)$, then SVD $(A) = U_1.S.V^T$ and SVD $(B) = U_2.S.V^T$.

Proof: Given SVD $(A) = U.S.V^T$, the right singular vector V can be determined from $A^T.A = V.S^2.V^T$. Let $C = A^T.A$, where $c\ (i, j) = \sum_{k=1}^{m} a\ (k, i).a\ (k, j)$ => the condition $\bigcup_{k=1}^{m} a\ (k, i) = \bigcup_{k=1}^{m} b\ (k, i)$, for all i $(1 \leq i \leq n)$, and $\bigcup_{k=1}^{n} a\ (j, k) \neq \bigcup_{k=1}^{n} b\ (j, k)$, for j $(1 \leq j \leq m)$ makes no difference to C. As a result, V and S are same for A and B. However since $A \neq B$ => SVD $(A) = U_1.S.V^T$ and SVD $(B) = U_2.S.V^T$

As a consequence of the above result, we can assume that whenever there is a row tampering attack, there is a change in left Eigen vector of matrix M. The following theorem helps us realize that it is possible to trace the rows of M where the tampering has occurred.

Theorem 2: If B is derived from A, by row tampering a set of rows $\{r_i: 1 \leq i \leq m\}$, then rows of left Eigen vectors of A and B are different by the same set $\{r_i: 1 \leq i \leq m\}$.

Proof: For a given matrix A, with SVD $(A) = U.S.V^T$, U can be determined from $A.V = U.\ S$. It can be easily be shown that $u\ (i, j) = s_j^{-1} \sum_{k=1}^{n} (a\ (i, k).\ u\ (k, j))$. By Theorem 1, we know that $V_1 = V_2$ and $S_1 = S_2$. If rows of A $\{r_i: 1 \leq i \leq m\}$ are tampered => Set of rows $\{r_i: 1 \leq i \leq m\}$ will be different for the left vectors of A and B.

So, by theorem 2 we can say that the non-zero rows of *DiffU* $= |U - U'|$ will indicate the set of rows $\{r_i: 1 \leq i \leq m\}$ that were tampered. Since rows of U change in the same pattern as modified M, it is possible to trace the presence of RS. *DiffU* will point out the rows that have been shuffled. The shuffle pattern can be identified by sorting the left Eigen vectors, and observing the mapping between the rows of the sorted vectors. In case of a CES, *DiffU* will point out the rows, but not the exact shuffled column elements. Hence, it is not possible to trace a CES or {CES, RS} as opposed to RS.

2.2.2 Verification and tracing of Column Tampering (CT) attacks

The following theorem helps us prove that column tampering is related to change in right Eigen vector only.

Theorem 3: Given matrix $A_{mxn} \neq B_{mxn}$ such that $\bigcup_{k=1}^{n} a\ (i, k) = \bigcup_{k=1}^{n} b\ (i, k)$, for all i $(1 \leq i \leq m)$, and $\bigcup_{k=1}^{m} a\ (k, j) \neq \bigcup_{k=1}^{m} b\ (k, j)$, for j $(1 \leq j \leq n)$, then SVD $(A) = U.S.V_1^T$ and SVD $(B) = U.S.V_2^T$.

Proof: Given SVD $(A) = U.S.V^T$, the right singular vector U can be determined from $A.A^T = U.S^2.U^T$. Let $C = A.A^T$, where $c\ (i, j) = \sum_{k=1}^{n} a\ (i, k).a\ (j, k)$ => the condition $\bigcup_{k=1}^{n} a\ (i, k) = \bigcup_{k=1}^{n} b\ (i, k)$, for all i $(1 \leq i \leq m)$, and $\bigcup_{k=1}^{m} a\ (k, j) \neq \bigcup_{k=1}^{m} b(k, j)$, for j $(1 \leq j \leq n)$ makes no difference to C. As a result, U and S are same for A and B. However since $A \neq B$ => $SVD\ (A) = U.S.V_1^T$ and SVD $(B) = U.S.V_2^T$.

As a consequence, we can assume that whenever there is a column tampering attack, there is a change in right Eigen vector of matrix M. The following theorem helps us realize that it is possible to trace the columns of M where the tampering has occurred.

Theorem 4: If B is derived from A, by column tampering a set of columns $\{c_i: 1 \leq i \leq n\}$, then rows of right Eigen vectors of A and B are different by the same set $\{c_i: 1 \leq i \leq n\}$.

Proof: For a given matrix A, with SVD $(A) = U.S.V^T$, V can be determined from $U^T.\ A = S.V^T$. It can easily be shown that $v\ (i, j) = s_i^{-1} \sum_{k=1}^{n} (u\ (k, j).\ a\ (k, i))$. By Theorem 3, we know that $U_1 = U_2$ and $S_1 = S_2$. If a set of columns of A $\{c_i: 1 \leq i \leq n\}$ are tampered => similar set of rows $\{c_i: 1 \leq i \leq m\}$ will be different for the right Eigen vectors of A and B.

So, by theorem 4, we can say that the non-zero rows of *DiffV* $= |V - V'|$ will indicate the set of rows $\{c_i: 1 \leq i \leq m\}$ that were tampered. Since rows of V change in the same pattern as modified M, it is possible to trace the presence of *CS*. *DiffV* will point out the rows that have been shuffled. The shuffle pattern can be identified by sorting the right Eigen vectors, and observing the mapping between the rows of the sorted vectors. In case of a RES, *DiffV* will point out the rows, but not the exact shuffled row elements. Hence, it is not possible to trace a RES or {RES, CS} as compared to CS.

2.2.3 Verification and tracing of Row-Column Tampering (RCT) attacks

As seen from Theorem 1 and 3, left and right Eigen vectors help us realize the presence of attacks. Therefore, we can intuitively say that a combination of row and column tamper attacks will affect the right and left Eigen vectors of A. As a result we get non-zero *DiffU* and *DiffV* and can verify the presence of a row-column attack.

Row and column tampering can occur in any order and any number of times. Say, we have row tampering (RT) order $<rt_1, rt_2 ... rt_n>$ and column tampering (CT) order $<ct_1, ct_2 ... ct_n>$. Row column tampering occurs such that order of rt_i and ct_i is not compromised. For such cases, theorem 5 helps us prove that it is possible to determine final outcome of $<rt_1, rt_2 ... rt_n>$ and $<ct_1, ct_2 ... ct_n>$.

Theorem 5: Given row tampering (RT) and column tampering (CT) pattern, the order of application of tampering to a matrix is independent of the resultant matrix.
Proof: Given matrix A where SVD (A) = $U_1S_1V_1^T$. By theorem 1 RT on A results in B where SVD (B) = $U_2S_1V_1^T$. By theorem 3, CT on A results in matrix D where SVD (D) = $U_1S_1V_2^T$. We have two orders of attack <RT, CT> and <CT, RT>.
Case <RT, CT>: By theorem 1, RT on matrix A will result in B, such that SVD (A) = $U_1S_1V_1^T$, SVD (B) = $U_2S_1V_1^T$. Since the row information is invariant, then by theorem 3 CT on matrix B will result in matrix C, such that SVD (C) = $U_2S_1V_2^T$
Case <CT, RT>: By theorem 3, CT on matrix A will result in B, such that SVD (A) = $U_1S_1V_1^T$, SVD (B) = $U_1S_1V_2^T$. Since the column information is invariant, then by theorem 1 RT on matrix B will result in matrix C, such that SVD (C) = $U_2S_1V_2^T$. Since both the cases give the same result, we can say that the order of application of tampering to a matrix is independent of the resultant matrix.

As observed above, the order of application of RT and CT are independent of each other, the net resultant left and Eigen vectors of the final matrix are same as those corresponding to row tampering order $<rt_1, rt_2... rt_n>$ and column tampering order $<ct_1, ct_2 ... ct_n>$ applied to M. If RT and CT corresponded to row and column shuffling only, then it is possible to predict the shuffling pattern. In other cases we restricted to finding the columns and rows where attacks took place. In such cases we assume that the attack to be a random attack and can trace the pattern of attack as shown in subsection 2.2.4.

2.2.4 Verification and tracing of Random Attack

Values of M (i, j) are changed randomly. By equation 1, any change in the j^{th} column elements of M is reflected in the j^{th} row of V, and any change in the i^{th} row of M is reflected in the i^{th} row of U. As a result, we have both *DiffU* and *DiffV* non-zero. Therefore, the indication that *DiffU* (i, k) and *DiffV* (j, p) change is non-zero, points that M (i, j) has changed due to random attack. The elements changed in M will give us the random attack used to tamper motion data.

2.3 Optimizations

The number of non-zero Eigen values determines the number of computations involved in tamper detection and information required for tamper detection. This will be significant while considering the case for scalability. The following discussion describes the optimizations that can be considered to aid scalability and reduce computations.

It can be observed from equation 1, the contribution of left U_k and right V_k vectors to the matix M is determined by their corresponding Eigen value s_k. As observed in section 2.1 the Eigen values can be ordered as $s_1 \geq s_2 \geq . . . \geq s_{min (m, n)} \geq 0$, where s_1 is significantly larger than other values. This implies the contribution of V_1 and U_1 is significant as compared to other vectors. Therefore any change ($M' – M$) will be reflected in $DiffU_1$ and $DiffV_1$. If we keep only the first left and right Eigen vectors, we can save computations and storage as follows:

- **Computation Reduction:** Initially we had n left and right Eigen vectors for comparison. We have m elements in left vector and n elements in right vector. As a result, we have O (n ($m + n$)) comparisons. By restricting it to first vectors, we now have O ($m + n$) comparisons. Therefore we save computations are O ((n -1) ($m + n$)).
- **Storage Reduction:** When we are storing U and V for n Eigen vectors, then storage required is O (n ($m + n$)). However, once we use first Eigen vectors storage is reduced to O ($m + n$). The reduction is identified as O (($n – 1$) ($m + n$)).

2.4 Advantages

The advantages of the scheme are described as follows:

2.4.1. Computational and Storage Advantages over Private Watermarking

Private fragile watermarking schemes use the original matrix M to extract the watermark. This will require the O ($m.n$) original data to be stored in the databases and during computations analysis of O (mn) elements in ($M – M'$). The proposed methodology used O (m) first left Eigen vector and O (n) first right Eigen vector. As a result it used only O ($m + n$) space and requires analysis of O ($m + n$) elements. Therefore, we save space and computations by O ($m.n – m – n$). Since $m > n$ (see section 2.1), for motion data we can have large reduction in space.

2.4.2 Better Accuracy over Existing Error Detection Methods

Error detection is a well studied topic with techniques [10] such as cyclic redundancy check (CRC), parity bit checking, and checksums. Schemes such as CRC and checksum are storage efficient and faster than our proposed scheme, as they do not require O ($m + n$) space. However, they are not capable of locating the errors in the matrix. Therefore, the proposed scheme is more efficient in terms of accuracy of error detection.

3. EXPERIMENTS AND DISCUSSION

All the experiments are carried out on angular motion data (Euler angles). This data was obtained in a *.csv* format created by Vicon IQ [16] MoCap system (Motion Capture facility at University of Texas at Dallas). The data consists of joint information of a skeleton, expressed in terms of rotational (Euler angles) and translational (co-ordinates) data. The motion clip used in these experiments can be expressed as *225 x 57* matrix, where frames = *225* and *19* skeleton joint rotational data (3 Euler angles) values.

3.1. Analysis of Attack Patterns

Attack patterns can be detected in automatically by our technique, and this detection can be visualized in this subsection. In Figure 4, CS and RES attack can be perceived on a joint. It shows the original trajectories for a joint, and also the attacked data for the same. It can be seen that one of the angles is shuffled with some other column, resulting in totally different data. This causes the joint to behave abnormally. The spikes in the figure depict the row element shuffling attacks, which can be seen in case of all the three angles. Figure 6 shows the effect on first right singular vector due to CS and RES attacks. The circled parts are the change in values due to row element shuffle (RES) attack. The other changing values are due to a column shuffle attack. The left singular vectors remain unchanged in these 2 cases as seen in Figure 8.

Figure 5 shows one Euler angle over the entire time duration before and after a RS and CES attack. The spikes (circled values) represent the CES attack. Figure7 shows the first left singular vectors before and after the attack (circled spikes represent CES values). The first right singular vectors do not change during a RS and CES attack, as observed in Figure 9.

Random attack affects left and right Eigen vectors. These effects can be visualized Figure 10 and Figure 11. Figure 10 corresponds to the comparison of first right Eigen vectors. Figure 11 shows first left Eigen vectors of original and attacked data. The dissimilarity between the Eigen vectors serve as evidence for tamper detection.

4. CONCLUSION

The paper proposed singular value decomposition (SVD) based technique to detect tampering of archived motion data. It has been shown that this method is capable of verifying and tracing the attack patterns on motion data matrix $M_{m \times n}$. The proposed scheme supports addition of copyright based watermarks. The information and computation required for tamper detection for a $(m \times n)$ matrix is reduced from $O(m.n)$ to $O(m + n)$, making the solution scalable.

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.

REFERENCES

[1] Chuanjun Li, B. Prabhakaran and S.Q. Zheng, *Similarity Measure for Multi-Attribute Data,* Proceedings of the 30th IEEE International Conference on Acoustics, Speech, and Signal Processing (ICASSP), Philadelphia, PA USA, pp. II-1149 – II-1152, March 2005

[2] Chung-Ping Wu, C.-C. Jay Kuo, Fragile Speech Watermarking for Content Integrity Verification (2002), citeseer.ist.psu.edu/wu02fragile.html

[3] C.S. Collberg, and C. Thomborson, "Watermarking, Tamper-Proofing, and Obfuscation—Tools for Software Protection," *IEEE Trans. Software Eng.,* Aug. 2002, pp. 735-746.

[4] C.Kailasanathan, R.Safavi Naini, and P.Ogunbona, "Fragile Watermark on Critical Points", International Workshop on Digital Watermarking Seoul, Korea, (Springer-Verlag), November 21-23, 2002.

[5] D. Kundur and D. Hatzinakos, "Digital watermarking for telltale tamper proofing and authentication", Proceedings of the IEEE, vol. 87, no. 7, pp. 1167--1180, July 1999

[6] F. Cayre, O. Devillers, F. Schmitt and H. Maître, Watermarking 3D Triangle Meshes for Authentication and Integrity, INRIA Research Report RR-5223, Jun. 2004

[7] H. Golub and C.F.V.Loan. Matrix Computations. The Johns Hopkins University Press, Baltimore, Maryland, 1996

[8] H.T. Wu and Y.M. Cheung, A Fragile Watermarking Approach to 3D Meshes Authentication, Proceedings of the 7th Workshop on Multimedia & Security (ACM'05), pp. 117-123, 2005.

[9] Ingemar Cox, Matthew Miller, Jeffrey Bloom, Mathew Miller, Digital Watermarking: Principles & Practice (The Morgan Kaufmann Series in Multimedia and Information Systems)

[10] E.T. Lin and E.J. Delp, "A review of fragile image watermarks", in Proc. of ACM Multimedia & Security Workshop, Orlando, 1999, pp. 25—29

[11] J. Fridrich, M. Goljan, and A. C. Baldoza, "New fragile authentication watermark for images," in Proc. IEEE Int. Conf. Image Processing, Vancouver, BC, Canada, Sept. 10--13, 2000.

[12] Larry L. Peterson, Bruce S. Davie, Computer Networks: A Systems Approach, 3rd Edition (The Morgan Kaufmann Series in Networking)

[13] M. M. Yeung and B.-L. Yeo, "Fragile watermarking of three dimensional objects," Proc. 1998 Int'l Conf. Image Processing, ICIP98, volume 2, pp. 442--446. IEEE Computer Society, 1998.

[14] Shuntaro Yamazaki, "Watermarking Motion Data", In Proc. Pacific Rim Workshop on Digital Steganography (STEG04), pp.177-185, Nov 2004

[15] Tae-hoon Kim, Jehee 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).

[16] Vicon, www.vicon.com

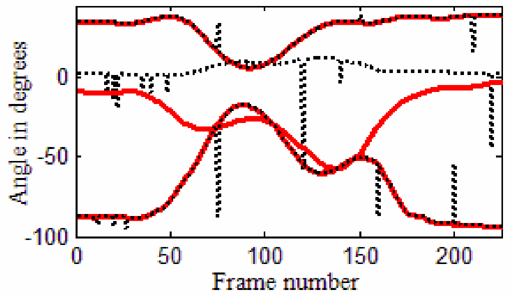


Figure 4: Effect on trajectories of a joint due to CS and RES attack

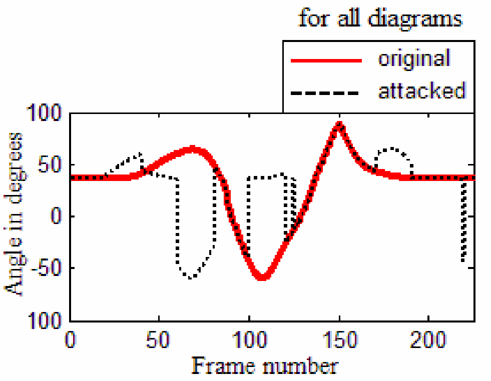


Figure 5 : An Euler angle value over the entire duration upon RS and CES attack

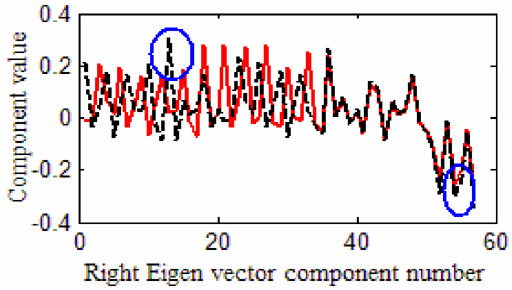


Figure 6: First right singular vector upon a CS and RES attack

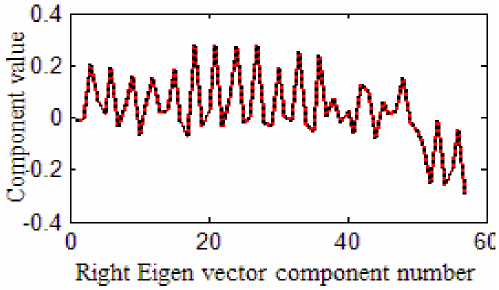


Figure 7: First left singular vectors upon RS and CES attacks

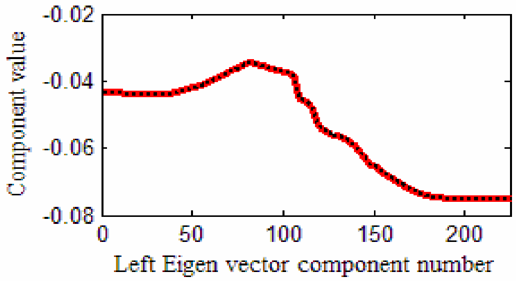


Figure 8: First left singular vector upon a CS and RES attack

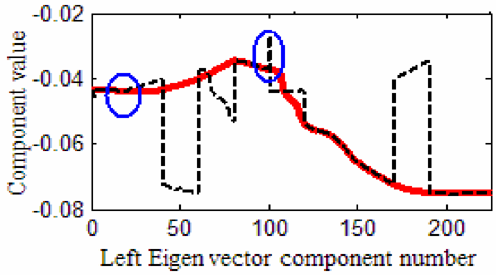


Figure 9: First left singular vectors upon a RS and CES attack

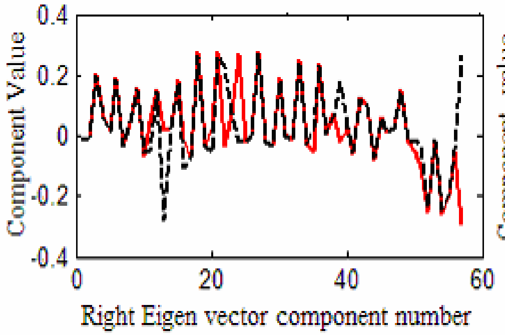


Figure 10: First right singular vector upon a random attack

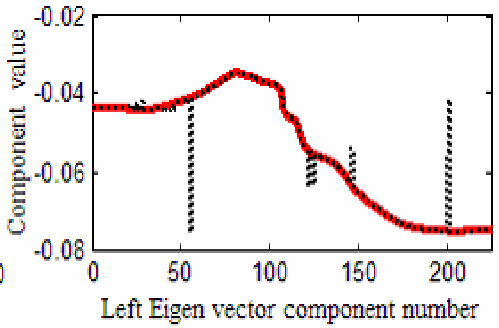


Figure 11: First left singular vector upon a random attack