# Fragile Watermarking of 3D Motion Data

R. C. Motwani, K. E. Bekris, M. C. Motwani, and F. C. Harris, Jr.
Computer Science and Engineering Department
University of Nevada, Reno
Reno, NV 89557

**Abstract**

With the increase in demand of motion planning solutions for digital actors in animation movies, 3D games and virtual reality, copyright protection of motion data becomes increasingly important. Likewise, a lot of time and effort is invested in generating motion capture data, which is susceptible to plagiarism due to its digital nature. To address this problem, watermarking techniques have been used for copyright protection of digital data. This paper proposes a fragile watermarking algorithm for 3D motion curves. An imperceptible watermark is inserted into the wavelet coefficients of the 3D motion data prior to publishing the 3D digital work that utilizes the motion data. The novelty of the proposed scheme lies in using wavelet transform for watermark insertion. To detect altered or plagiarized copies of the 3D motion data, the extracted watermark is compared to the originally inserted watermark. If the comparison results fall within a threshold value, the motion data is declared as authentic, else flagged as plagiarized or tampered. Experimental results indicate that the proposed watermarking scheme is imperceptible and resistant to affine transformations.

## 1 Introduction

Motion planning algorithms have applications in a wide range of domains such as robotics, animation, biomedicine etc. Motion capture and motion planning is used to produce compelling graphic animations used in entertainment and e-commerce applications [1, 2]. Artists spend a great deal of time on their animations and 3D media work and many have had their work stolen and claimed by others, with little way of proving the original owner of the concept. This problem exists mainly due to the nature of digital domain which allows copying and altering of the original data. Watermarking techniques have been used to solve such issues related to unauthorized copying and malicious modifications of multimedia. Multimedia watermarking embeds a hidden piece of information in the original content, without interfering with the intended use of the content. Typical watermarks carry ownership information such as a logo as an image, a copyright message string or a number. It is required that the embedded watermark should survive affine transformations, compression loss and noise due to the nature of the digital transmission channel and customized visualization needs of the end user. At the same time, the watermark should be perceptually invisible and should be embedded in such a way that a potential hacker or an adversary should be forced to make substantial changes in the multimedia content in order to destroy the watermark.

This paper explains a technique to watermark 3D motion data (generated by motion planning algorithms [3] used to compute feasible paths to move rigid bodies in a 3D space with obstacles). The novelty of this paper lies in inserting the watermark by modifying the wavelet coefficients of the 3D motion curve. An informed watermarking approach is used which requires the original unmarked data and the watermark for watermark extraction from the data that is believed to have been modified. The proposed technique generates a fragile watermark. Fragile watermarks find application in tamper detection and integrity proofing. Unlike robust watermarks, which are tolerant to a designated class of transformations (such as compression, cropping, scaling, and smoothing), fragile watermarks can not be detected even if the digital data has been modified only to a minor extent (such as additive noise).

## 2 Background

While research related to 3D trajectory watermarking is still in its infancy stage, a wide range of possibilities exist for watermarking motion data. The 3D motion path can be expressed as a curve in the 3D space. This curve can be slightly modified directly in the spatial domain by embedding a watermark (which could be a number, text, or a random sequence of numbers) into randomly selected points on the 3D curve. The authors in [4], describe a spatial domain technique to watermark 3D motion capture data. Most watermarking techniques [5] adopt a certain level of randomness in the algorithm to battle attacks on watermark removal by brute force approach. However, this is the simplest approach and has its drawbacks. Embedding the watermark directly in the spatial domain makes it vulnerable to removal or replacement attacks. It is preferred to transform the motion curve into frequency domain. This assures that the watermark is spread across the 3D curve such that removal or replacement of parts of the curve does not destroy the watermark completely. Yamazaki [6] proposes segmentation of the motion data, followed by a discrete cosine transform operation on each segment to embed the watermark. So far, the frequency domain based motion data watermarking algorithms have not explored the use of wavelet transform.

Alternatively, the 3D motion data can be decomposed into a series of motion vectors in the spatial domain and the phase or magnitude of the most important vectors can be modified by the watermarking algorithm. Motion vectors are ranked in importance based upon their proximity to obstacles (in the context of motion planning data) and the obtuseness or acuteness of the angles with successive motion vectors. Motion vector manipulation approach has been utilized by several video watermarking algorithms [7, 8, 9]. In [10], watermark values are embedded in the less phase angle changed components and the large valued motion vectors. While, a motion vector based approach is employed for video watermarking, this approach has not been explored thus far for 3D trajectories and is worth experimenting. However, certain modifications must be applied before directly adopting this approach from the video domain. Since the points on the 3D motion curve are very close to each other, successive motion vectors do not vary significantly. Gradient variation computation is required at successive points to derive a series of cumulative motion vectors which vary by a measurably large amount. Another way is to use Beneden's idea [11] to partition the curve into several parts and then change each part. In this way, we spread the watermarking all over the motion sequence.

Moreover, the authors in [12], propose a progressive watermarking scheme for 3D motion capture data that uses frame decimation. A robust, blind 3D motion capture data watermarking algorithm for human motion animation is proposed in [13] that is cluster-based and uses quantization techniques.

## 3 Proposed Method

Our approach implements a prototype in spread spectrum domain by using a Haar [14] wavelet transform on the 3D data and alters the wavelet coefficients that represent the fluctuation of data. Haar wavelets are chosen because these are efficient in capturing sharp discontinuities in the data, are very simple to understand, easy to implement and serve as a prototype for studying more sophisticated wavelets such as Daubechies and Gabor wavelets. Wavelet [15] transform is preferred over Fourier or Discrete Cosine transforms because it captures not only a notion of the frequency content of the input, by examining it at different scales or resolutions, but also temporal content, i.e. the times at which these frequencies occur. If we look at a signal with a large window, we notice gross features [16]. Similarly, if we look at a signal with a small window, we notice detailed small features. The benefit of wavelet analysis is the ability to see both the gross and detailed versions of the signal at the same time, in addition to multi-resolution representation of the data since wavelet transform down samples the data.

The algorithm design criteria are as follows: i) The motion data to be watermarked should be independent of the motion capture technique or motion planning algorithm (RRT, PRM [3] etc.) chosen to determine the path. This goal is achieved by watermarking the motion data after the motion planning algorithm has computed the entire motion path. ii) Watermark should be uniformly distributed across the motion data to avoid hacking using brute force techniques. This can be achieved in several ways and one approach would be to use spatial-frequency domain technique. iii) Watermark should be content-based. The watermark is inserted as noise in the original data and the level of noise, that can be used to avoid distortion, is decided by the content of the data where the watermark needs to be inserted. iv) Watermark should be imperceptible. The changes caused in the motion path due to addition of watermark should not distort the motion data to an extent that the human eye can perceive the difference.

A prototypical approach is implemented using the Haar wavelet transform which is applied to each of the $x$, $y$ and $z$ component of 3D motion data. Block diagram for the algorithm is shown in Fig. 1.

Given a discrete signal $f$ of length $2^N$, with equally spaced samples, a decomposition of the signal into two sub-signals of half its length is constructed.

$$f = f_1, f_2, \ldots, f_N \tag{1}$$

The first sub-signal (known as scalar co-efficient) represents the average of the original signal and is computed by the following equation:

$$c_m = \frac{(f_{2m-1} + f_{2m})}{2} \tag{2}$$

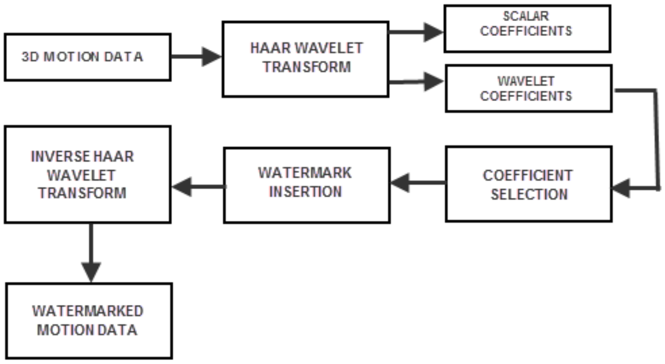


Figure 1: Block Diagram

where m$= 1, 2, 3, \dots, N/2$

The second sub-signal (known as wavelet co-efficient) represents the difference (i.e. fluctuation) between the subsequent components of the original signal and is denoted by:

$$d_m = \frac{(f_{2m-1} - f_{2m})}{2} \tag{3}$$

where m$= 1, 2, 3, \dots, N/2$

Magnitudes of the fluctuation sub-signal $d$ are often significantly smaller than those of the original signal. A visual representation of Haar wavelet decomposition into scalar ($c$) and wavelet ($d$) co-efficients is shown in Fig. 3, 4, and 5. The above equations are the basis of 1-level Haar transform. The inverse transform derives the original signal from the two sub-signals using the following equation:

$$\begin{aligned} f \quad = \quad & (c_1 + d_1)/\,2, (c_1 - d_1)/\,2, \dots, \\ & (c_{N/2} + d_{N/2})/\,2, (c_{N/2} - d_{N/2})/\,2 \end{aligned} \tag{4}$$

### 3.1 Implementation

A motion planning query is solved using the OOPSMP package [17, 18] and the solution path co-ordinates are exported to *MATLAB* for further processing. Representation of the motion data in *MATLAB* is shown in Fig. 2.

#### 3.1.1 Watermark Insertion

Haar transform equations are applied to the $x$, $y$ and $z$ co-ordinates separately. As stated earlier, Fig. 3, 4, and 5 show the disintegration of the transformed 3D motion data on each of the three axes.

The wavelet coefficients of each axis are compared to an experimental threshold value of $\lambda = 0.002$ which should be less than the distance metric used by the motion planning algorithm. If the absolute magnitude

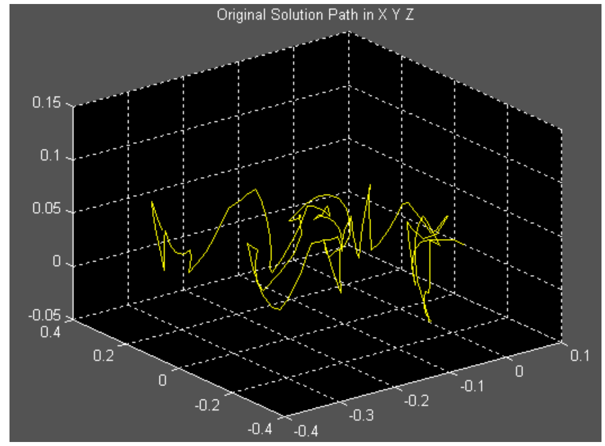


Figure 2: Original Motion Data plotted in MATLAB

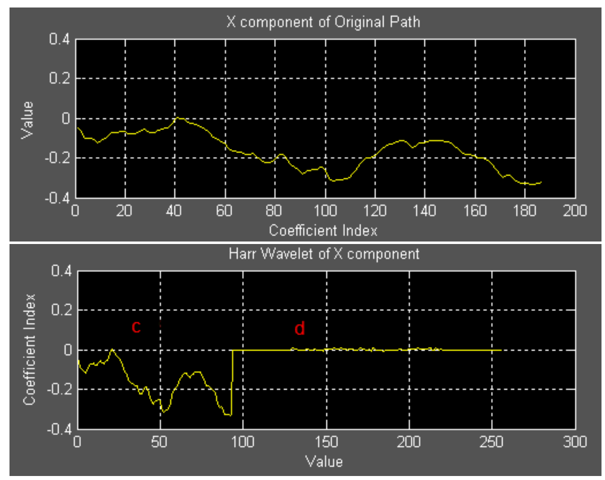


Figure 3: X-Component of the Motion Data and 1-Level Haar Wavelet Transform of the X co-ordinates

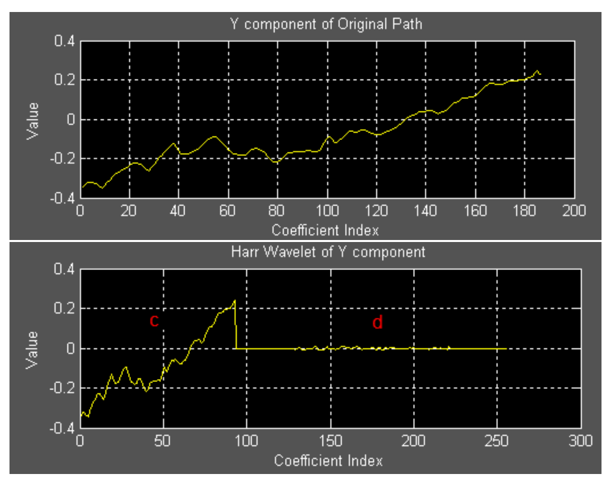


Figure 4: Y-Component of the Motion Data and 1-Level Haar Wavelet Transform of Y co-ordinates

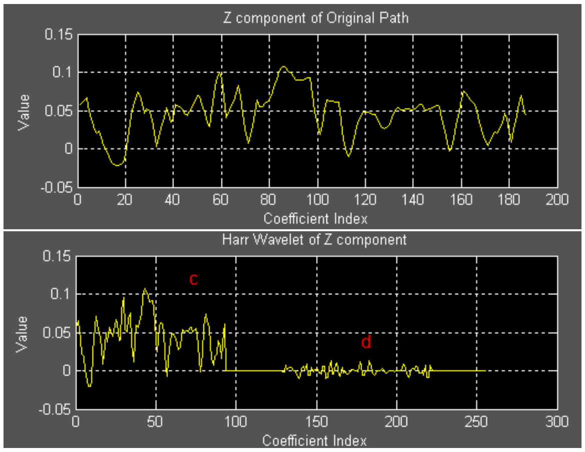


Figure 5: Z-Component of the Motion Data and 1-Level Haar Wavelet Transform of Z co-ordinates

of the coefficient exceeds the threshold, it is selected for watermark insertion and a watermark value $w$ is added to this coefficient. The index of this coefficient and the watermark $w$ is stored to enable watermark retrieval during the extraction phase.

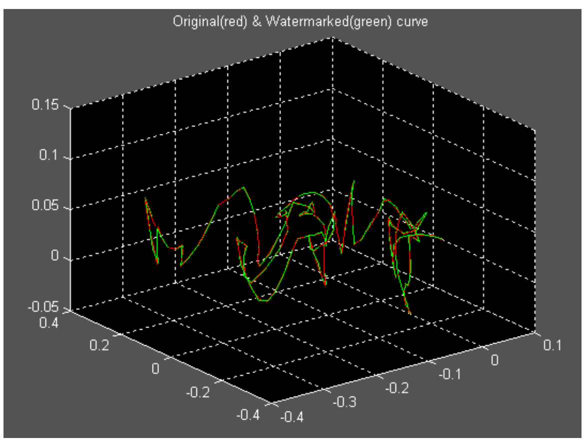


Figure 6: Original (red) and Watermarked (green) Motion Data

Inverse transform applied to the unmodified scalar coefficients and the modified wavelet coefficients yields the watermarked motion data as shown in Fig. 6. The amount of information inserted and the places where it is inserted in the motion data forms the watermark.

#### 3.1.2 Watermark Extraction

To detect if a motion data has been modified, wavelet domain representation of the original path is subtracted from the wavelet domain representation of the suspected path. Correlation of the subtraction result with the original watermark determines if the path is authentic or an illegal modified copy of the original.

#### 3.1.3 Similarity Measure

For measures of association between two sets of data, the most widely used linear correlation coefficient is Pearson's $r$ coefficient [19]. Given pairs of quantities (i.e. two sets of motion data $x$ and $y$) $(x_i, y_i)$, where $i = 1, \ldots, N$ and $\bar{x}$ is the mean of all $x_i$'s and $\bar{y}$ is the mean of all $y_i$'s, $r$ is given by the formula:

$$r = \frac{\sum_i (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_i (x_i - \bar{x})^2}\sqrt{\sum_i (y_i - \bar{y})^2}} \quad (5)$$

The value of $r$ lies between -1 and 1, inclusive. It takes on a value of 1 when the data points lie on a perfect straight line with positive slope, with $x$ and $y$ increasing together. The value 1 holds independent of the magnitude of the slope. If the data points lie on a perfect straight line with negative slope, $y$ decreasing as $x$ increases, then r has the value -1. A value of $r$ near zero indicates that the variables $x$ and $y$ are uncorrelated.

## 4 Experiments

Results for various attacks illustrated in Figs. 7, 8, 9, 10, 12, and 13 are listed in Table 1. Since our scheme is fragile, the watermark is invariant to affine transformations. The algorithm is also resistant to double watermarking attacks that embed a watermark into an already watermarked motion data. Any other attacks, such as additive Gaussian noise, cropping (i.e. removal of parts of data on the 3D curve), modification or replacement of original data, destroy the watermark. The plots of the motion data for the modify and replacement attacks dominantly impact the $x$ and $y$ axes, therefore the correlation value for the $z$ dimension are high.

| Attacks | X | Y | Z |
|---|---|---|---|
| No Attack | 100% | 100% | 99.99% |
| Translation | 99.99% | 99.99% | 99.99% |
| Rotation | 100% | 100% | 99.99% |
| Scaling | 99.19% | 100% | 99.37% |
| Noise | 6.22% | -6.76% | 14.34% |
| Cropping | 19.44% | 38.48% | 22.11% |
| Modify | -4.7% | 5.50% | 98.30% |
| Replacement | 13.99% | -6.54% | 99.99% |
| Double Watermark | 100% | 100% | 99.99% |

Table 1: Correlation Results

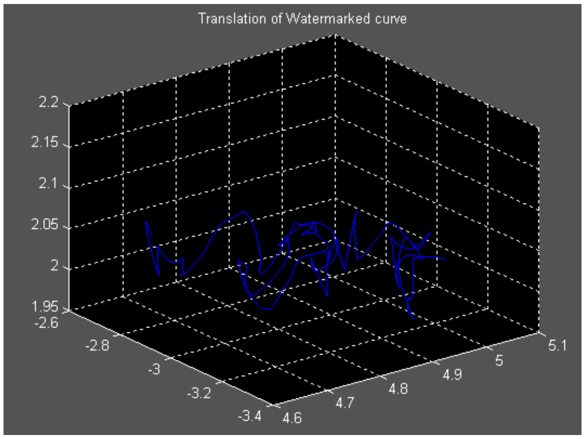


Figure 7: Translation of Watermarked Curve

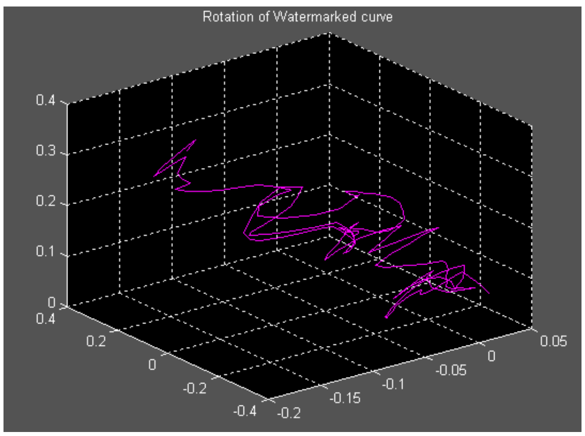


Figure 8: Rotation of Watermarked Curve

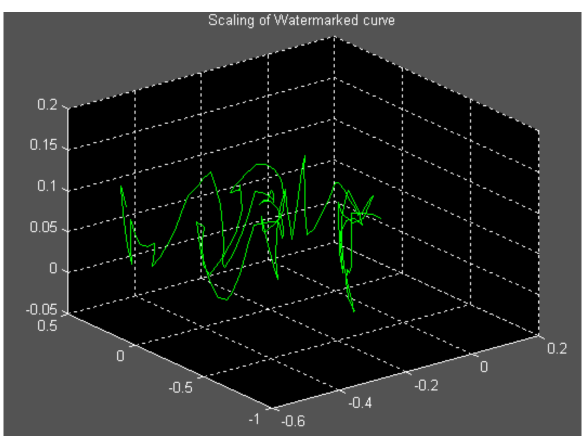


Figure 9: Scaling of Watermarked Curve

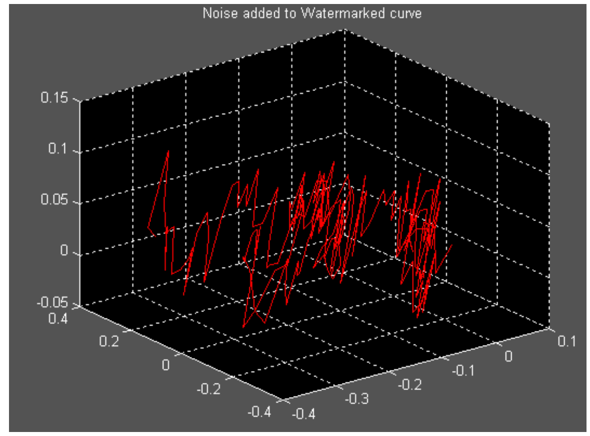


Figure 10: Noise Attack

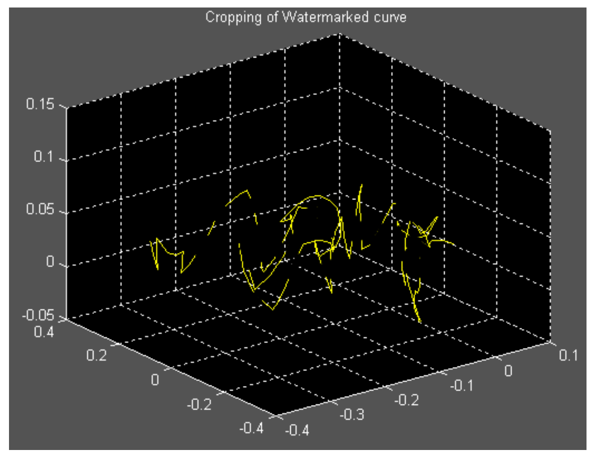


Figure 11: Cropping Attack

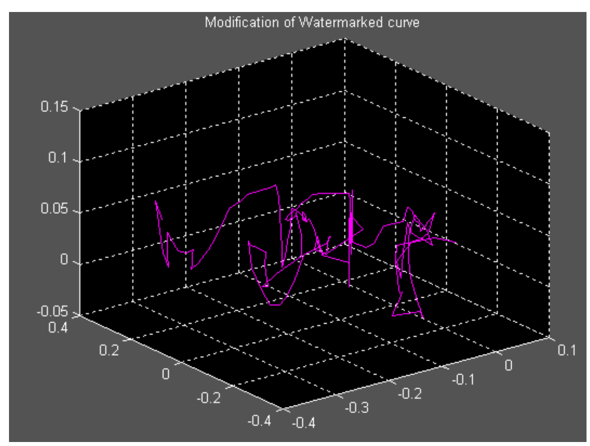


Figure 12: Modify Attack

## 5 Conclusion and Future Work

This paper has presented a watermarking algorithm that embeds information in the wavelet coefficients of the 3D motion data. Experimental results show that the watermark in imperceptible and resistant to affine transformations. The work presented in this paper is preliminary and future work entails incorporating the complexities of motion planning al-

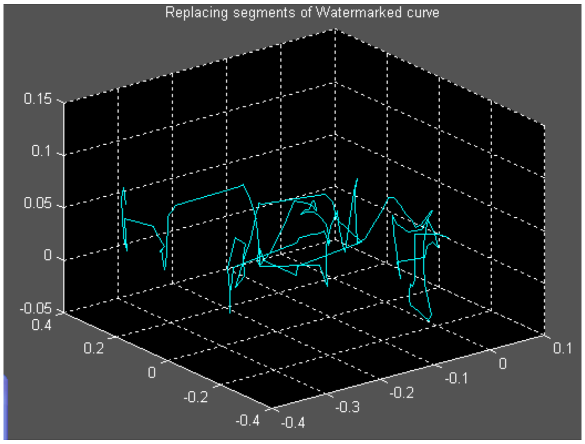


Figure 13: Replacement Attack

gorithms such as distance of motion planning solution path from obstacles. Additional future work will be towards adapting the watermarking algorithm to solution paths for different terrains and non-rigid bodies such as humanoid robots.

## References


[1] J. Kuffner, K. Nishiwaki, S. Kagami, M. Inaba, and H. Inoue, "Motion planning for humanoid robots," in *Proceedings of 11th International Symposium of Robotics Research (ISRR)*, 2003, pp. 1–10.

[2] F. Yu and T. Li, "Coordinated motion planning for 3d animation: Using chinese lion dance as an example," in *Proceedings of International Workshop on Advanced Image Technology*, 2007.

[3] S. LaValle, *Planning Algorithms*. Cambridge University Press,, 2006.

[4] P. Agarwal, K. Adi, and B. Prabhakaran, "Robust blind watermarking mechanism for motion data streams," in *Proceedings of the 8th workshop on Multimedia and security*. ACM, 2006, pp. 230–235.

[5] S. Li and M. Okuda, "Watermarking for progressive human motion animation," in *IEEE International Conference on Multimedia and Expo*, 2007, pp. 1259 – 1262.

[6] S. Yamazaki, "Watermarking motion data," in *Proceedings of Pacific Rim Workshop on Digital Steganography*, 2004, pp. 177–185.

[7] L. Kezheng, Y. Wei, and L. Pie, "Video watermarking temporal synchronization on motion vector," in *International Conference on Intelligent System and Knowledge Engineering*, 2008, pp. 1105–1110.

[8] N. Mohaghegh and O. Fatemi, "H.264 copyright protection with motion vector watermarking," in *International Conference on Audio, Language and Image Processing*, 2008, pp. 1384–1389.

[9] Z. Liu, H. Liang, X. Niu, and Y. Yang, "A robust video watermarking in motion vectors," in *Proceedings of the International Conference on Signal Processing*, 2004, pp. 2358– 2361.

[10] J. Zhang, J. Li, and L. Zhang, "Video watermark technique in motion vector," in *Proceedings of the XIV Brazilian Symposium on Computer Graphics and Image Processing*, 2001, p. 179.

[11] O. Benedens, "Geometry-based watermarking of 3d models," in *IEEE Computer Graphics and Applications*, 1999, pp. 46–55.

[12] S. Li and M. Okuda, "Iterative frame decimation and watermarking for human motion animation," *ICGST International Journal on Graphics, Vision and Image Processing,Special Issue on Watermarking*, vol. 07, pp. 41–48, 2010.

[13] P. Agarwal and B. Prabhakaran, "Blind robust watermarking of 3d motion data," *ACM Transactions on Multimedia Computing and Communication Applications*, vol. 6, no. 1, pp. 1–32, 2010.

[14] M. Ali, "An introduction to wavelets and the haar transform," http://www.cs.ucf.edu/m̃ali/haar/, last Accessed August 5, 2010.

[15] R. Polikar, "Wavelets in signal processing," http://users.rowan.edu/p̃olikar/WAVELETS /WTtutorial.html, last Accessed August 5, 2010.

[16] A. Graps, "An overview of wavelets," http://www.amara.com/current/wavelet.html, last Accessed August 5, 2010.

[17] E. Plaku, K. Bekris, and L. Kavraki, "Oops for motion planning: An online open-source programming system," in *International Conference on Robotics and Automation*, 2007, p. 37113716.

[18] OOPSMP, "Oopsmp package - Rice University Physical and Biological Computing Lab," http://www.kavrakilab.rice.edu, last Accessed August 5, 2010.

[19] M. Norusis, *SPSS Introductory Guide: Basic Statistics and Operations*. McGraw-Hill, 1985.