3D Watermarking Robust to Accessible Attacks

Mahsa Eshraghi
Department of Computer Science
University of Calgary
mborooje@cpsc.ucalgary.ca

Faramarz F. Samavati
Department of Computer Science
University of Calgary
samavati@cpsc.ucalgary.ca

ABSTRACT

In this paper we propose a new watermarking approach for 3D polygonal meshes. In the field of multimedia watermarking, mesh watermarking is less advanced than image or video. The main difficulty is due to the irregular structure of 3D meshes that makes them more vulnerable to attacks. To address this issue, we introduce a new criterion -accessibility in the common 3D graphics software-for selecting attacks and propose a list of common attacks based on this criterion. Consequently, in this paper, we consider attacks like cut, change of the format, affine transformation, subdivision and small scale deformation that are usually found in the graphical software. We present several schemes to achieve robustness against these attacks. Watermark embedding is done by perturbing the vertices in their tangent space that provides us invisibility of the watermark. Hiding capacity in this method is as big as the number of the vertices of the mesh that is usually big enough for expensive 3D meshes.

1. INTRODUCTION

Digital watermarking is an information-hiding technique extensively used for copyright protection of multimedia images, video, and audio. With the increasing use of 3D graphics in video games, films, CAD, and virtual reality applications, interest has recently moved toward 3D watermarking. Creating a 3D model takes a vast amount of effort, experience and technology, and so copyright protection of this type of data is increasingly important. Whereas watermarking of other digital media has become a fairly mature subject, 3D watermarking has much room to grow.

Polygon meshes are a widespread representation for 3D objects; a mesh M is a collection of vertices V and faces F that define an object 3D. This structure – arbitrarily connected, irregularly sampled points – makes mesh watermarking a challenging problem compared to, say, regularly sampled 2D images.

Watermarking methods should have three basic qualities: high capacity, invisibility , and robustness. Higher capacity allows more identifying information to be stored. Invisibility is usually desired so that the watermark is more difficult to intentionally destroy and also the original shape of model is preserved while adding more information. A very important property is robustness, as an algorithm must be able to recover the watermark when the data is manipulated. In general, it is very hard to anticipate all possible attacks to 3D objects and it is also very difficult to design a 3D watermarking to be robust against all possible attacks.

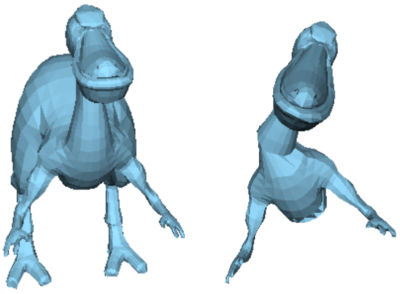


Figure 1: With our robust watermarking method, we could prove that the mesh on the right was originated from the left mesh.

Our strategy, then, is to make a list of important attacks and then design a watermarking technique against this list. The question becomes, what are important attacks? The answer to this question is very important for designing a robust watermarking system. From our perspective, attacks that can be performed with common graphical software (i.e. Maya, 3dsMax, Blender) can be considered as important attacks, because they are easily accessible.

To justify this, consider the following scenario. Bob has an illegal copy of a Reunion Tower building mesh. He does not know any technical details of the model or its data structure, but he knows that the mesh has a watermark. Therefore, Bob loads the mesh in 3dsMax, subdivides the mesh, cuts off part of the mesh, applies a small deformation, and saves it to another format. With many current methods, the watermark would be lost. In this scenario, the available tools in common graphical software define the possible attacks; as the number of tools grows, so does the attack list.

In this paper we make three contributions. First, we propose the novel strategy of ranking and listing attacks based on their practical accessibility, as discussed above. For this work, we consider a variety of attacks such as affine transformations, cutting, change of resolution, change of connec-

IMMERSCOM 2007, October 10-12, Verona, Italy
Copyright © 2007 978-963-9799-06-6
DOI 10.4108/ICST.IMMERSCOM2007.2117

tivity, manipulating the data structure and swapping the 3D format. Second, to obtain robustness against the above attacks , we propose several attack recovery schemes like geometric segmentation, geometric indexing, and resolution normalization. Finally, we introduce a new watermarking methodology based on perturbing vertices in their tangent space, which helps to preserve the mesh shape and also provides a high capacity for the watermark.

This paper is organized as follows. Section 2 gives a brief review of related work. In Section 3, we suggest a list of easily accessible attacks, and propose methods that help the watermark to be recovered after such attacks. Our tangent-space watermarking method is described in Section 4, while Section 5 evaluates the proposed watermarking technique through different attack scenarios. Finally, in Section 6 we conclude the paper and suggest possible directions for future work.

2. BACKGROUND

Image watermarking is a well-developed area in the field of multimedia watermarking. Current image watermarking techniques embed the watermark in three ways (as summarized by Lin and Delp [13]). The simplest way is to embed the information into the spatial domain such as the least-significant bit (LSB) plane of the image in a certain sequence. Another way is to change coefficients in a transform domain, such as Discrete Cosine Transform (DCT). Finally, recent methods exploit properties of the human visual system to find suitable regions for placing message data invisibly.

Video watermarking is another well-developed application. As with images, the watermark can be created either in the spatial or the DCT domain [18]and [14].

While there are many mature techniques for image and video watermarking, fewer methods exist for mesh watermarking. Mesh watermarking techniques can be classified into two different groups: those that embed data in the spatial domain, and those that use a transform domain such as wavelets.

Ohbachi et al. [15] present a spatial-domain method that perturbs geometric and topological features of models, such as the shapes of triangular mesh faces, or the volume of induced tetrahedrons. Their approach has low capacity and is vulnerable to attacks that alter vertex connectivity. Bendens [2] describes another geometry-based method that embeds a watermark in the surface normals, and their method is robust to polygon simplification methods.

Praun et al. [19] present a robust mesh watermarking algorithm that converts the mesh to a coarse base mesh and a sequence of refinement operations. They embed the watermark by perturbing refined vertices that cause the greatest geometric change to the model. The same scalar functions, computed on the original mesh, are later used to extract the watermark. Their watermarking algorithm is robust against many attacks such as transformation, noise, vertex reordering and cropping. Their method is not robust to subdivision attacks that can be easily done with 3D managing software programs such as MAYA.

Some transform-domain methods use multiresolution operators to embed a watermark [7, 22]. Such approaches embed the watermark in the wavelet detail coefficients at a specific level of decomposition, then reconstruct the model from the detail coefficients and the coarse level model in addition to the watermarked detail coefficients. In this way they can guarantee invisibility, because the watermark is embedded into wavelet coefficients. Such methods are also robust to a number of attacks. For example the algorithm presented by Uccheddu et al. [22] is robust against rotation, translation, uniform scaling, filtering , noise, and cropping. The method presented by Kanai et al. [7] is robust against affine transformations and noise. However, such techniques are limited to models with subdivision connectivity, and further subdividing the model will destroy the watermark.

3. ATTACK RECOVERY SCHEMES

In our method, we consider affine transformations (rotation, translation, and scaling), cutting, subdivision, local deformations, and vertex re-ordering (as a result of changing the 3D format) being common manipulations. These operations can be found in most of the graphical software and therefore they are easily accessible for any attacker. Robustness against this set of attacks is achieved by applying “anti-attack” operations before embedding or extracting the watermark. The basic idea is that any possible attack A has a “normalization” solution N_A. For example, if A is subdivision, then the normalization N_A is to “reverse” subdivide the suspected mesh. In the following subsections, we describe these initial stage “anti-attack” operations, and the attacks that they address.

3.1 Coordinate Normalization

Manipulations that affinely transform the model (rotation, translation or scaling) will displace the vertices from their original coordinates. We embed the watermark by slightly altering vertex position, and later extract it by comparing the vertices of the suspect and original meshes and finding the displaced vertices. Therefore, coordinate changes could potentially ruin the watermark unless we bring both meshes into the same coordinate system before watermark extraction. To do this we use a three-step process to transform our models into a unit bounding box with a certain orientation (Figure 2). First, we transfer the center of the model to origin. Next we align the principal components of the mesh with the standard axes. Principal components are calculated by the eigen-decomposition of the covariance matrix of mesh vertices, using principal component analysis [6]. This is mainly because PCA represents an object-oriented coordinate system. Finally, the model is bound into a unit box. We call this unit box as normalized coordinate system. Because PCA is invariant under rotation, translation, uniform-scaling, this process leads to robustness against any combination of these transformations. However, PCA might be changed after a drastic non-uniform scaling. In this case, the order of the main axes can be changed. For example, a tall ellipsoid can be mapped to a wide ellipsoid that leads to the lost of watermark. For this kind of cases, we re-order the main axes for all possibilities(xyz, xzy, yxz,yzx, zxy,zyx) and search for the watermark.

3.2 Geometric Segmentation

To avoid losing the watermark after local deformations and cutting that affect some parts of a mesh while leaving others unchanged, we divide the mesh into meaningful segments and embed the watermark in each segment separately. In this way, we can extract the watermark from any unchanged segment(s). Segmentation not only achieves ro-

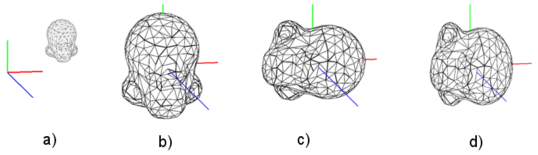


Figure 2: The coordinate normalization process: a) original model; b) centering c) aligning d) bounding.

bustness against local deformations, but also prevents illegal copying of individual parts of the mesh.

We use the segmentation algorithm proposed by Katz et al. [8], which segments a mesh into visually meaningful parts at regions of concavity (Fig. 3). The main advantage of this method over others ([12] and [10]) is its invariance to pose changes – such as moving an arm of monkey model – and local deformations – such as locally scaling the head of the monkey. These pose and deformation invariances lead to more robust watermarking.

For our purposes, segmentation is done as an initial stage before embedding and extracting the watermark. Afterward, each segment of the mesh is processed separately to embed the watermark. In the extraction part, comparison is done segment by segment between the original and suspect models. By embedding the watermark several times, we achieve robustness against local manipulation and cutting attacks.

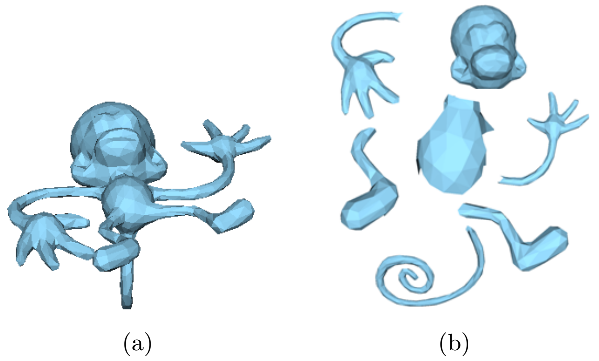


Figure 3: Segmentation of a monkey model. Robustness against local deformations is achieved by embedding the watermark in each segment.

3.3 Geometric Indexing

Watermarking methods based on vertex positions require a consistent ordering of vertices. Often the vertex ordering given by the mesh file format is used, but such a watermark can be easily destroyed by simply changing the format or vertex connectivity. Therefore, we would like to find a unique order for the vertices that does not depend on format or connectivity. For this purpose we re-arrange the vertices according to geometrically invariant features of the mesh. Our approach is to sort the vertices based on their projections onto the principal components (Fig. 4).

The first principal component is the line that goes through the centroid and also minimizes the square of the distance of each point to that line. Therefore it goes through the maximum variation in the data, and as the result the probability that two or more points project to the same place is diminished. However, if this happens we can sort next by their projection onto the second principal component, and finally we can sort according to the third principal component. The resulting arrangement is used to re-index the vertices. In this way, the watermark is robust to change to vertex ordering or connectivity.

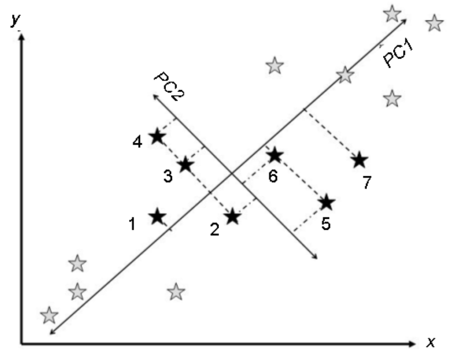


Figure 4: A geometrically-invariant vertex ordering is obtained by projecting onto the principal component axes (ascending order).

3.4 Resolution Normalization

A model's resolution can be changed easily by subdivision, destroying the watermark. For example, Catmull-Clark and Loop are two popular subdivision methods for 3D meshes(see [24]) and they are supported in most graphical software. Subdivision methods enhance the quality of the mesh via face or vertex split followed by some geometric displacements. Therefore, they change the connectivity and the geometry at the same time . The resulting mesh would have higher quality(resolution). Such operations can be used as a kind of attack to remove many kind of watermarks including our tangent space displacement. However, for almost every subdivision method there exists a reversal method that can recover the original data, such as Wang et al. [23] and Olsen et al [16]. Therefore, prior to extracting the watermark we find the closest resolution of the suspect to the original mesh.

4. EMBEDDING AND EXTRACTION

The main goal of watermarking is to hide some data in an object, in a way that is shape preserving and such that the watermark can be recovered after manipulations that intend to destroy it.

4.1 Watermark Embedding

We embed the watermark by displacing the mesh vertices. Since displacing the vertices can deform the mesh, we perturb each vertex slightly in its tangent plane that has the least impact on the shape of the mesh. A watermark can be represented as a binary sequence $W = \{w_1, w_2, ..., w_m\}$. To embed the watermark each vertex is displaced by a vector

$$d_i = w_j * \delta_i * T_i \ ,$$

where w_j is the j-th digit of the watermark's binary sequence, δ_i is the displacement scalar value for vertex v_i and

T_i is a vector in the tangent space of v_i. For a consistent direction of perturbation T_i is calculated as the cross product of normal vector n_i with a fixed global vector.

The displacement factor is an important parameter, it cannot be set to a constant threshold for any kind of the vertices. For a vertex with very close neighbors, the threshold must be small while for a vertex with far neighbors it can be a bigger threshold. Therefore, we calculate a geometric tolerance by defining an upper bound for δ_i as $\delta_i^{max} = \sum_k^1 |e_j|/k$ (Figure 5), where $|e_j|$ is the Euclidean distance between v_i and its j_{th} neighbor.

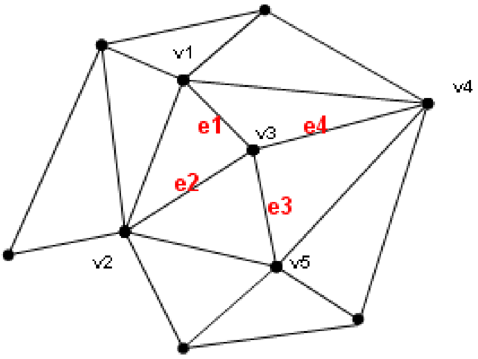


Figure 5: δ_3^{max} for v_3 is calculated as $\delta_3^{max} = (|e_1| + |e_2| + |e_3| + |e_4|)/4$.

The displacement of each vertex should satisfy $0 \leq \delta_i \leq \delta_i^{max}$, where δ_i is calculated as $\delta_i = \mu * \delta_i^{max}$ for $0 \leq \mu \leq 1$. If μ is large, the shape of the mesh may be impacted (i.e. the watermark may be visible). Smaller μ values can better preserve the shape, but also leave the watermark more sensitive to any attack. Therefore, there is a trade-off between invisibility of the watermark and robustness of the technique.

We tested watermark embedding and extraction for different values of μ in the form of $\{\mu = 1/n | n \in N\}$ (N stands for Natural numbers). Table 1 shows the maximum value of μ (μ_{max}) that is shape preserving as well as robust for its 3D model. $|V|$ in the second column is the number of the vertices for each mesh. Using 1/100 for μ can work for all of the meshes. However, it is clear that the curvy (free-form) surfaces are less sensitive to perturbation than engineering surfaces (like the plane and the building). In general free-form surfaces support larger μ.

4.2 Watermark Extraction

Watermark extraction is done segment by segment. For each segment of the suspect mesh we find the closest segment in the original mesh using a resemblance threshold ρ. Watermark is extracted by taking the difference of each vertex v_i in the original segment and those of the watermarked segment $\acute{v_i}$ using the calculated arrangement: $\lambda_i = \acute{v_i} - v_i$. Comparison is done in the same coordinate system. The resemblance threshold ρ is different for each segment and is calculated in the initial stage as:

$$\rho = \sum_{i=1}^{n} d_i \pm \epsilon$$

where d_i is the amount of perturbation for vertex v_i after embedding and ϵ accounts for possible attacks on the watermarked model that displace the vertices.

Depending on the watermark binary digit w_j, $\acute{v_i}$ can be displaced for $w_j = 1$ or not displaced for $w_j = 0$. Knowing this fact, we extract a binary sequence W^* by analyzing the value of λ_is. An accurate analysis is very important to avoid the false positive or false negative problems. To account for numerical instability, we determine W^* according to:

$$w_i^* = \begin{cases} 1, & \lambda_i > \alpha \\ 0, & \lambda_i \leq \alpha \end{cases},$$

where α is a very small threshold value.

By comparing the extracted binary sequence W^* and original watermark W, we can claim the copy-right.

5. RESULTS

In this section we demonstrate the result of investigating mesh robustness against affine transformation, cutting, local deformation, change of format and subdivision while watermark extraction. A dinosaur 3D model with 2039 vertices and a monkey model with 1252 vertices are used to evaluate the performance of the method. Figure 6 shows the effect of different attacks on the watermarked monkey (a) and watermarked dinosaur (g). Invariance to affine transformation attacks that change the coordinate of vertices, is obtained through coordinate normalization. Rotated monkey and dinosaur shown in Fig 6(b) and (h) are two examples that we could extract the watermark with no error. Through segmentation we partition the models and process each segment separately in the watermark embedding and extraction. Fig 6(e) and (k) show a few locally deformed models, we extracted the watermark from individual segments successfully. Fig 6(c) and (i) show the models after cutting attack. In these cases, watermark was extracted from the remaining segments without error. Fig 6(d) and (j) show subdivision attacks on the models. We used a few iterations of reverse subdivision to bring the suspect model to the closest resolution to the original model and then extracted the watermark. Watermark could also be recovered against a combination of considered attacks. For example

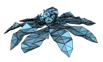
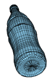
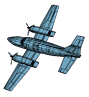
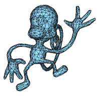
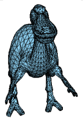
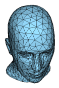
model	$\lvert V\rvert$	μ_{max}	model	$\lvert V\rvert$	μ_{max}
	806	1/14		616	1/5
	4656	1/50		3745	1/60
	624	1/100		1252	1/4
	2039	1/10		689	1/6

Table 1: $|V|$ is the number of vertices for each mesh, μ_{max} is close to the maximum value of μ for which the shape of the mesh is preserved and the watermark is robust.

Fig 6 (f) shows the watermarked monkey after cutting arms and legs, applying one level of Catmull-Clark subdivision, and rotating the result, and Fig 6 (l) is the watermarked dinosaur after a local scale on the beak, cut of tail, followed by change of format from OBJ to OFF. We could recover the watermark from these two models successfully.

6. CONCLUSION AND FUTURE WORKS

In this paper we presented a new watermarking method for 3D polygonal meshes. Invisibility is achieved through embedding the watermark by slight tangential displacement of mesh vertices. To accomplish robustness, we first suggested a list of common attacks on 3D models, and for each attack in this list we found a “normalization” scheme to recover the watermark from that attack as shown in Table 2. We divide different attacks into high level and low level groups. High level group includes the manipulations that can be done easily by ordinary people using the common graphical tools, for instance cutting or subdivision in Maya or 3dsMax. On the other side, low level attacks deal with manipulations that require enough knowledge of the data structure of 3D meshes like moving all the vertices in a way that preserves the shape of the model. Our attack list includes the high level attacks that are more common among computer users such as affine transformation, cutting, local manipulations, change of format, and subdivision.

In the application of 3D meshes in computer generated animation our watermarking scheme will work for still objects. It also works for moving objects, unless all the different segments are manipulated that is less probable. As future works we like to improve our watermarking robustness against more attacks.

Attack	Normalization Scheme
Translation	Coordinate Normalization
Rotation	Coordinate Normalization
Scale	Coordinate Normalization
Cut	Geometric Segmentation
Local manipulation	Geometric Segmentation
Change of Format	Geometric Indexing
Subdivision	Resolution Normalization

Table 2: List of the attacks and their solutions.

Acknowledgments

We would like to thank Sagi Katz for providing us with the segmented models. We also thank Luke Olsen for editorial help. The support of the Natural Sciences and Engineering Research Council of Canada and ICORE is gratefully acknowledged.

7. REFERENCES

[1] M. Arnold. *Techniques and Applications of Digital Watermarking and Content Protection.* Northwood, MA, USA: Artech House, Inc., 2003.

[2] O. Benedens. Geometry-based watermarking of 3d models. In *IEEE CG and A*, pages 46–55, 1999.

[3] I. Biederman. Recognition-by-components: A theory of human image understanding. In *Psychological Review*, volume 94, pages 115–47, 1987.

[4] L. Brog and P. Groenen. *Modern Multi-dimensional Scaling: Theory and Applications.* Springer, Berlin Heidelberg, 1977.

[5] M. Corsini, M. Barni, F. Bartolini, R. Caldelli, V. Cappellini, and A. Piva. Towards 3d watermarking technology. In *EUROCORN*, pages 393–396, 2003.

[6] I. T. Jolliffe. *Principal Component Analysis.* Springer Series in Statistics, 2002.

[7] S. Kanai, H. Date, and T. Kishinami. Digital watermarking for 3d polygons using multiresolution wavelet decomposition. In *Sapporo Japan*, 1998.

[8] S. Katz, G. Leifman, and A. Tal. Mesh segmentation using feature point and core extraction. In *The Visual Computer*, volume 21, pages 649–658, 2005.

[9] S. Katz and A. Tal. Hierarchical mesh decomposition using fuzzy clustering and cuts. In *ACM Transactions on Graphics*, pages 954–961, 2003.

[10] Y. Lee, S. Lee, A. Shamir, D. Cohen-Or, and H. Seidel. Intelligent mesh scissoring using 3d snakes. In *IEEE Computer Society*, pages 279–287, 2004.

[11] R. Li and H. Zhang. Segmentation of 3d meshes through spectral clustering. In *IEEE Computer Society*, pages 298–305, 2004.

[12] X. Li, T. W. Toon, and Z. Huang. Decomposing polygon meshes for interactive applications. In *SIGGRAPH*, pages 35–42, 2001.

[13] E. T. Lin and E. J. Delp. A review of data hiding in digital images. In *Video and Image Processing Laboratory*, 1999.

[14] L. J. P. T. J. D. G. H. J. Maes. M, Kalker. T. Digital watermarking for dvd video copy protection: What issues play a role in designing an effective system? In *IEEE Signal Processing Magazine*, volume 17, pages 47–57, 2000.

[15] R. Ohbuchi, H. Masuda, and M. Aono. Watermarking three-dimensional polygonal models. In *ACM Multimedia, USA*, pages 261–272, 1997.

[16] R. Olsen. L, Samavati. F.F. Bartels. Multiresolution for curves and surfaces based on constraining wavelets. In *Computers and Graphics*, volume 31, pages 449–462, 2007.

[17] J. E. Overall and M. Spencer. Subset scaling (subscl): Multidimentional scaling based on a subset of objects or variables. In *Journal of Educational and Psychological Measurement*, 1990.

[18] F. Perez-Gonzalez and J. R. Hernandez. A tutorial on digital watermarking. In *Security Technology*, 1999.

[19] E. Praun, H. Hoppe, and A. Finkelstein. Robust mesh watermarking. In *ACM SIGGRAPH*, 1999.

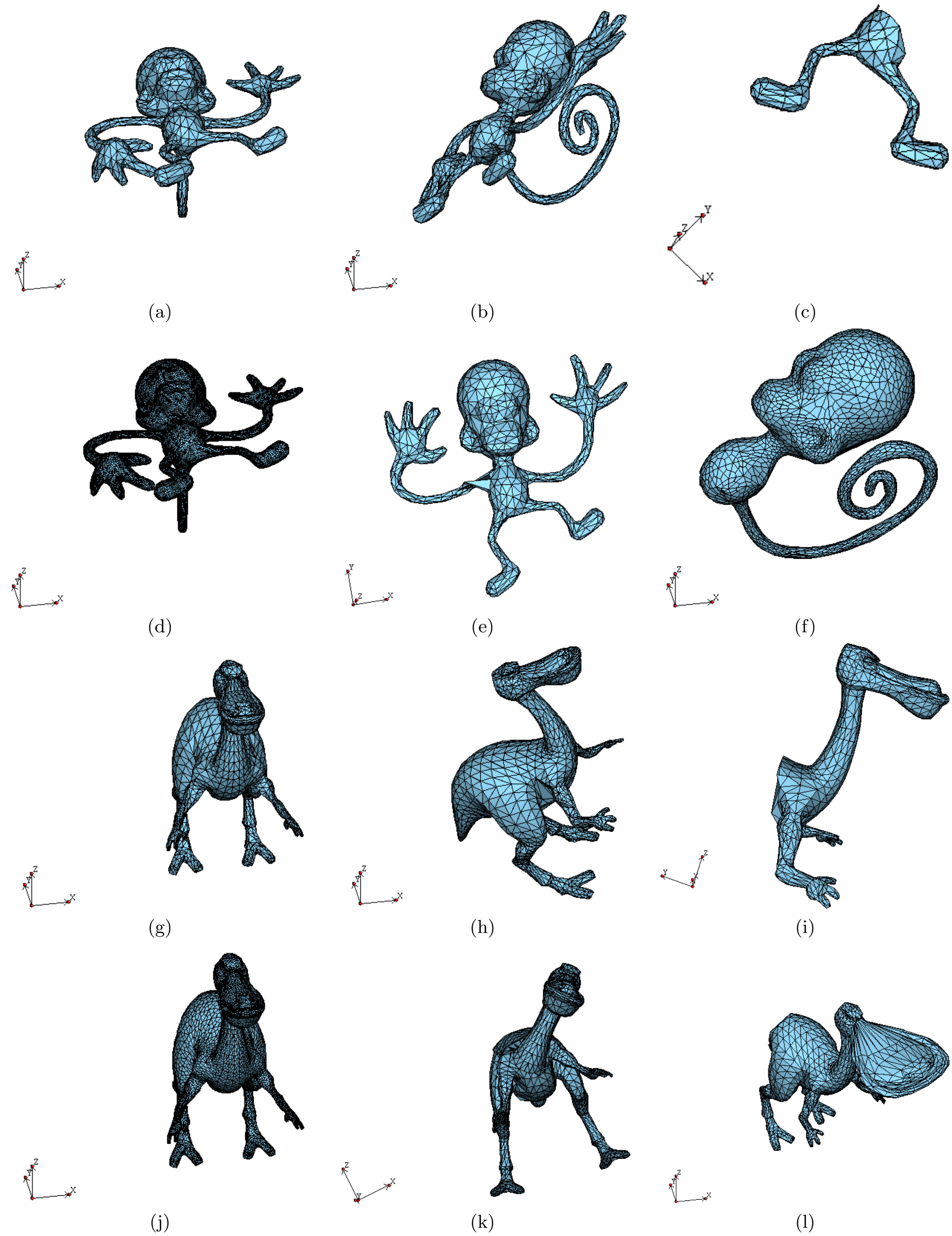
[20] L. I. Smith. A tutorial on principal components analysis. 2002.

[21] K. Tanaka, Y. Nakamura, and K. Matsui. Embedding secret information into a dithered multi-level image. In *IEEE Military Communications Conference*, 1990.

[22] F. Uccheddu, M. Corsini, and M. Barni. Wavelet-based blind watermarking of 3d models. In *MM and Sec, Germany*, pages 143–154, 2004.

[23] H. Wang, K. Qin, and K. Tang. Efficient wavelet construction with catmull-clark subdivision. In *Visual Computer*, 2006.

[24] D. Zorin and P. Schroder. Subdivision for modeling and animation. In *SIGGRAPH*, 2000.



(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

(j) (k) (l)

Figure 6: Watermarked monkey and dinosaur and effect of different attacks on them: (a) watermarked monkey (b) rotation (c) cut (d) subdivision (e) local rotation and translation (f) cut, subdivision, and rotation (g) watermarked dinosaur (h) rotation (i) cut (j) subdivision (k) local rotation (l) local scale, cut, change of format.