http://ansinet.com/itj

ITJ

ISSN 1812-5638

INFORMATION TECHNOLOGY JOURNAL

ANSInet
Asian Network for Scientific Information
308 Lasani Town, Sargodha Road, Faisalabad - Pakistan

Information Technology Journal 9 (3): 412-421, 2010
ISSN 1812-5638

A High-capacity Steganographic Scheme for 3D Point Cloud Models

[1,2]Ke Qi, [1]DaFang Zhang and [2]Dongqing Xie
[1]School of Computer and Communications, Hunan University, Changsha, China
[2]School of Computer Science and Educational Software, Guangzhou University, Guangzhou, China

Abstract: This study presents a new high-capacity spatial steganographic scheme for 3D point cloud models using a Self-Similarity Position Matching (SSPM) procedure. The new scheme partitions the 3D point cloud model to patches, clusters patches into similarity patch chains using self-similarity measures and generates the codebook. The representative patches and similar patches are then taken from the codebook for every similar patch chain as the reference patches and the message patches. Finally, every message point in the similar message patches which has the point-to-point correspondence with a certain reference point in the reference patch can embed at least four bits using the proposed SSPM, which embeds information by shifting the message point from current point to the corresponding embedding position that is computed over virtual sphere with the reference point as the center. Experimental results show that the proposed technique is secure, has high capacity and low distortion and is robust against uniform affine transformations such as transformation, rotation, scaling. In addition, a concise shape description and a similarity measures are devoted to improving performance for forming codebook and constructing point traversal. The technique can be considered as a side-match steganography and has proven to be a feasible alternative to other steganographic schemes for 3D point cloud model.

Key words: Steganography, 3D point cloud models, self-similarity patch chain, codebook, spatial domain

INTRODUCTIONS

Steganography is an art of communicating in a way that hides the existence of the communication (Johnson and Jajodia, 1998). Compared with classical watermarking, which is a process for protecting copyright ownership, steganography is a technique about concealing the existence of the secret messages. Therefore, steganography tends to require higher data capacity, low distortion and security, but it can lead to relatively poor robustness (Cayre and Macq, 2003).

With the development of 3D applications and animation, many steganographic algorithms have been presented for 3D models. While 3D models are usually classified into mesh model and point cloud model, in which 3D mesh model normally consists of vertexes, edges and polygonal surfaces while point sample model consists of 3D geometry i.e., point information only, there have been comparatively many steganographic algorithms for 3D mesh model (Cayre and Macq, 2003; Aspert *et al*., 2002; Maret and Ebrahimi, 2004; Wang and Cheng, 2005; Zafeiriou *et al*., 2005; Cheng *et al*., 2006; Cheng and Wang, 2006, 2007) which always use topology, edge length or angle among meshes to hide information. However, since 3D point cloud model consists only point information without any edges and surfaces which are basic elements of information hiding for 3D mesh models, almost all steganography for 3D mesh models can not be applied to 3D point cloud models and only fewer data hiding approaches (Cheng *et al*., 2006; Cotting *et al*., 2004; Wang and Wang, 2005; Luo *et al*., 2006) for 3D point cloud model have been presented until now. Finding a way to fully exploit the features of 3D point cloud models is an important research issue.

Cheng *et al*. (2006) described an efficient data hiding scheme for point models based on a substitutive procedure. The Virtual Multi-Level Embed Procedure is used to embed three bits per point based on shifting the message point by its virtual sliding, extending and arching geometry property. As a result, they exploited high embedding capacity in the 3D space, which seems to be the source for the maximal capacity on 3D point cloud model.

Cotting *et al*. (2004) proposed a watermarking algorithm of point sampled geometry based on pseudo-spectral analysis. The algorithm partitioned the model into a set of patches by applying a fast hierarchical clustering scheme. Next, each patch was mapped into the space of

Corresponding Author: Ke Qi, School of Computer Science and Education Software, Guangzhou University, No. 230, Waihuan West Road, Guangzhou Higher Education Maga Center, China
Tel: (086) 020 39366925 Fax: (086) 020 39366375

eigenfunctions of a Laplacian operator to obtain discrete frequency bands. Finally, the messages were embedded into the low frequency components. The algorithm is suitable for watermarking since it has high robustness but with low capacity.

Wang and Wang (2005) presented a data hiding scheme for point models. The scheme used a Principal Component Analysis (PCA) (Rencher, 2002) and symmetrical swap procedure to embed messages. The algorithm suffered a capacity drawback that the data capacity in bits generally achieved only about half of the number of points in the model.

Luo *et al*. (2006) presented a reversible data hiding for 3D point cloud model. It started with creating a set of 8 neighbor vertices clustered set with randomly selected seed vertices. Next, an 8-neighbor integer DCT was performed to obtain coefficient. Finally, a highest frequency coefficient modification technique was employed to embed messages. The scheme has characteristic of reversibility but of very low capacity.

All of the above-mentioned steganography for 3D point cloud model can be categorized into transform domain and spatial domain. Approaches in transform domain (Cotting *et al*., 2004; Wang and Wang, 2005) normally have high robustness but very low capacity, while approaches in spatial domain (Cheng *et al*., 2006; Luo *et al*., 2006) have comparatively high capacity but low robustness. Research presented by Cheng *et al*. (2006) seems to be the source for the maximal capacity on 3D point cloud model. However, his scheme normally embeds only about multiples of three bits per embedding point.

Since, steganography tends to require high capacity and low robustness, we prefer to develop a blind scheme in spatial domain from the high capacity point of view. This study presents a new blind high-capacity steganography for 3D point cloud model, inspired by the concepts proposed by Hubo *et al*. (2008). The key idea is to construct codebook for self-similarity partitioned patch of the 3D point cloud model using self-similarity measures and exploit Self-Similarity Position Matching (SSPM) procedure to embed information by shifting the message point to a certain spatial position which is computed from the point-to-point corresponding reference point. To the best of our knowledge, this is the first 3D point cloud model steganographic scheme that uses the reference point matching relationship to embed messages which can be considered as a side-match steganography. This procedure also efficiently achieves high capacity of around four bits per point with little visual distortion. Furthermore, patch size, codebook size, the traversal list of codebook and embedding list over each message patch are used as secret keys for more security. Similar to previous 3D steganographic methods, the scheme is robust against affine transformations, which include translation, rotation, uniform scaling, or their combined operations. Experimental results show significant improvements in terms of capacity with respect to the most high capacity technique, that of Cheng *et al*. (2006).

CONSTRUCTION OF SIMILARITY PATCH CHAINS AND CODEBOOK

As an important technique of 3D model compression, self-similarity based compression of point set surfaces (Hubo *et al*., 2008) is a promising compression technology for massive point sets and ray tracing. For many 3D point models consist commonly of massive point sets with repetitive patterns and similar structures, such as creases, ridges, bumps, compression can be completed by replacing similar surface patches with an instance of a representative patch with efficient encoding and decoding.

Self-similarity patch chain is generally defined as the set of segments scattered around 3D point model with similar shapes or structures that can be identified by pre-defined shape descriptors and similarity measures (Fig. 1). As every patch is similar to the other patches in the same self-similarity clustered chain, every pair of the matching points in the pairs of similar patches can embed information using side-match spatial position matching.

In present proposed steganographic scheme, unlike the conventional compression purpose of Hubo *et al*. (2008), similar patch chains are efficiently constructed and the spatial position of the message point is shifted to embed information. Owing to their favorable similarity characteristics, these pairs of matching points are better for a 3D point cloud model to design the stegonagraphic scheme.

We give a short overview of the construction of similarity patch chains and the codebook. First, the

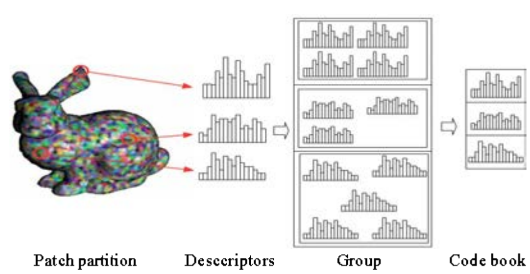

Fig. 1: Example of patch partition, construction of self-similarity patch chains and codebook generation

original 3D point model is partitioned to obtain patches. Then, we design patch descriptors, compute self-similarity, select representative patches and add all similar patches to construct self-similarity patch chains. Finally, each patch chain forms an entry in the codebook.

The construction of self-similarity patch chain can be decomposed in three steps: (1) patch partition with an octree; (2) measuring similarity among patches and (3) codebook generation for self-similarity patch chains.

Patch partition: Clustering-based partition uses iterative clustering as a tool to separate the input model into multiple regions according to local properties of points. k-means (Macqueen, 1967), particle simulation (Pauly *et al*., 2002a, b) and octree (Ai *et al*., 2009) are several partition algorithms.

For the purpose of meaningful self-similarity comparison and improving the likelihood of finding a similar match, the patch partition favors patches that are equally sized and lie on salient features (Hubo *et al*., 2008). To achieve this goal, we use the local surface reconstruction technique (Tamy *et al*., 2005) which chose the octree among possible candidates for its hierarchical multi-resolution structure with equally sized cells of regular shapes and for the efficiently geometrical approximation of the local point cloud in a cell.

As proposed by Tamy (Tamy *et al*., 2005), let P be the original 3D point cloud model. We begin with C, the bounding cube of P . The initial point set P is uniformly splitted into local point set P_i in a sub-cube C_i according to the cell boundaries and we run again the uniform partition along the axes until the following conditions are met.

$$\begin{aligned}&\forall j\in[0,k_i-1],\ n_{ij}\cdot n_i>\delta_a,\ \ \delta_a\in[0,1]\\&\frac{\left|(p_{ij}-o_i)\cdot n_i\right|}{d/2}<\delta_b,\ \ \delta_b\in[0,1]k_i<K\end{aligned}\quad(1)$$

where, P_{ij} be the j point of the partition P_i, n_{ij} be the local normal vector of the point P_{ij}, o_i be the local centroid of the sub-cube C_i, n_i be the average local normal vector of the partition P_i, k_i be the number of points in the sub-cube C_i, d be the edge length of the sub-cube C_i, the threshold value K be the patch size representing the max number of points in the sub-cube C_i, the parameter $\boldsymbol{\delta}_a$ represents the maximal deviation angle that the normal of a point can make according to the average normal of its current sub-cube, the parameter $\boldsymbol{\delta}_b$ represents the distance between a point and its projection onto the local average plane defined by o_i and n_i. We have set $\boldsymbol{\delta}_a$ to 0.2 and $\boldsymbol{\delta}_b$ to 0.3 in present experiments.

The unique hierarchical octree structure for the original 3D point model is then constructed using above partition algorithm in which every nonempty node of bottom layer represents a partitioned patch with no more than K points. The number of octree subdivision is saved for extraction. Figure 1 is an example of our patch partition.

Measuring similarity among patches: Before we can group similar patches, we need a way to measure how similar they are. Since, the patches are partitioned into sub-cells with same size and lie on salient features, the measuring is rather straightforward in two steps:

Alignment: Since patches are usually close to flat, they all can be oriented such that their supporting planes coincide. The key idea is to find supporting planes and align these planes. For efficiency reasons and inspired by Cheng *et al*. (2010) which uses principle plane analysis for segmentation, we simply compute the supporting plane defined by the local centroid of the sub-cube o_i and the local average normal vector n_i and rotate the patch such that the plane's normal is aligned with the Z-axis.

Computing and comparing patch descriptors: After patches alignment, we design a simple statistical descriptor to measure the self-similarity. Inspired by Johnson (Johnson and Hebert, 1999) which uses the spin map to compare the similarity; we compute height and distance histograms to compare the similarity of two patches.

For a patch P_i, let $d_H(\vec{p_{ij}})$ be the height value from point P_{ij} projecting to the supporting plane H (Fig. 2a) which can be calculated by $d_H(\vec{p_{ij}})$ = $(p_{ij}-o_i)\ n_i$. Let P_{ij}' (j = 1,..., m) be a set of points collected from projecting the point P_{ij} onto its supporting plane H, shown in Fig. 2b. Let

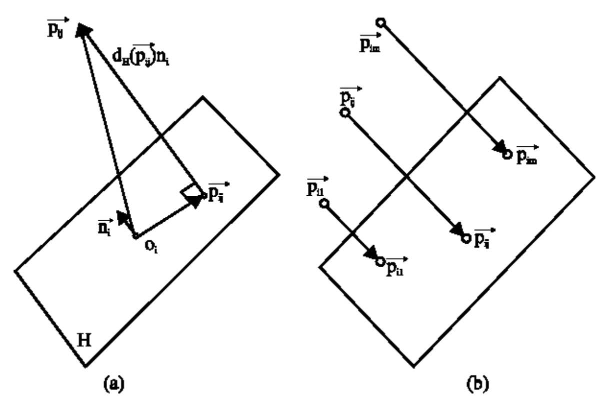

Fig. 2: Projecting point of a patch onto its supporting plane H. (a) the relationship between the original point and its corresponding point on H; (b) H is the supporting plane for the projected points

$r_H(\vec{p_{ij}})$ be the distance between P_{ij}' and local centroid of the sub-cube o_i which can be calculated by $r_H(\vec{p_{ij}}) = \|p_{ij}'-o_i\|$.

We compute two histograms of $d_H(\vec{p_{ij}})$ and $r_H(\vec{p_{ij}})$. The similarity between two patches can then be evaluated using any distance measure suitable for histograms. We use the cross-bin match distance technique proposed by Hubo *et al*. (2008) to measure the similarity:

The two histograms of $d_H(\vec{p_{ij}})$ and $r_H(\vec{p_{ij}})$ are converted to their cumulative version and concatenated into a single descriptor vector. Finally, the similarity of two patches can be simple performed by comparison on this vector using cross-bin match distance:

$$d(E,G) = \sum_i |E_i - G_i| \tag{2}$$

where, $E_i = \Sigma_{j\le i} E_j$ is the cumulative histogram of E and similarly for histogram G.

Codebook generation for self-similarity patch chains: Codebook generation for self-similarity patch chains is achieved by grouping similar items (patch descriptors) together. To achieve the goal, we use codebook generation technique (Hubo *et al*., 2008) with following steps:

- **Searching and clustering:** We select a representative patch and look for a set of all similar patches using a nearest neighbor query in the space of descriptor vectors. The cross-bin match distance and the radius of the query determine codebook size which represents how many code words will be created. This process continues until all patches have been clustered into one of the self-similarity patch chains. We always select the patch with the most number of points in the pool of the remaining patches as the representative patch of a certain patch chain. This ensures that every point in the similar patches can find its matching point in the reference patch
- **Refined matching:** In each cluster, we compute the affine alignment of each patch, i.e., translation and rotation, with respect to its representative patch via the Iterated Closest Point (ICP) algorithm (Rusinkiewicz and Levoy, 2001). After ICP processing, the similar patch orientation and translation matrix can be obtained and we can optimally match every point in the similar patches with a certain reference point in the reference patch. The refined matching is needed to construct point-to-point correspondence among points in the similar patch and the reference patch

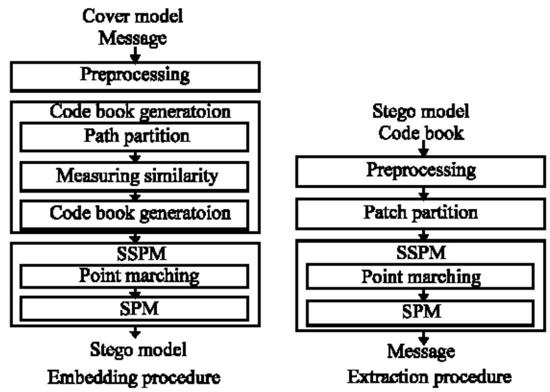

Fig. 3: Overview of the proposed scheme

- **Codebook generation:** After similar patches clustering and refined matching, the codebook is then generated for all self-similarity patch chains in which every entry simply consists of the representative patch, the similar patch index, the similar patch orientation and translation

THE PROPOSED STEGANOGRAPHIC SCHEME

Here, describes the proposed approach of steganography for 3D point cloud model. The overview of the embedding procedure and extraction procedure is shown in Fig. 3.

Information embedding: The steps of embedding procedure are as follows.

Preprocessing: Principal Component Analysis (PCA) determines three principle axes centered on the centroid of the model. The original coordinate system is translated to a new one. The new coordinate system has a new origin, which is the center of the model. It also has three basis vectors, which are the three principal axes. The PCA translated model is robust against uniform affine transformation such as translation, scaling and rotation.

Patch partition and codebook generation: This step partitions the model to patches using octree structure, constructs similarity patch chains with self-similarity measures and generates codebook. The central idea of the step is to cluster similar patches and produce codebook for every similar patch chain. The representative patch of every chain in the codebook is then selected as reference patch; we use this reference-patch model to embed information.

Self-similarity position matching procedure (SSPM): We consider every patch except representative patch in the codebook as message patches; we also consider every point in the message patch as message point. To embed information in every message point, we use a SSPM. In SSPM, we embed the information by shifting the message point from current point to a certain spatial position; it guides the change of the position of the message point on the spherical coordinates of the virtual sphere. The SSPM procedure is just like the registration and resampling process (Pauly *et al.*, 2002a, b) as following steps:

- We use a pseudo sequence to traverse the codebook. The traversal list is maintained as a secret key for both the embedding and extraction procedures
- For each entry in the codebook, we take the representative patch P as the reference patch and select the similar patch Q as the message patch according to the similar patch index. The similar patch orientation and translation matrix is then used to accurate aligning the message patch with the reference patch P
- The first step before both the embedding and extraction is a patch object transformation. The reference patch P is translated so that its center is the center of the message patch Q. Let P' be the translated reference patch and p'∈P' be the translated reference point. In the following we always operate on the reference point p' and the reference patch P'
- In order to embed information, we need to establish a point-to-point correspondence among points in the message patch Q and the reference patch P' . For this purpose we use point matching process. The method finds a point q∈Q such that the 3D Euclidian distance from a point p'∈P' is the smallest among all the points (Fig. 4) and repeats the process for all the point p' in P'. Since, we select the reference patch with the most number of points, the points in the message patch can always find its corresponding point in the reference patch
- As the point-to-point correspondence has been established, we embed information using spatial position matching (SPM) process. The key idea is to construct point-to-point correspondence between the message point and the reference point and shifting the message point to a certain spatial position which is computed from the point-to-point corresponding reference point

Let o_q be the local centroid of the patch Q. Assume the reference point p'∈P' is the center of a virtual sphere and the radius of it is a threshold value R. Define embedding points as some specific points inside the virtual sphere or on the surface of the virtual sphere.

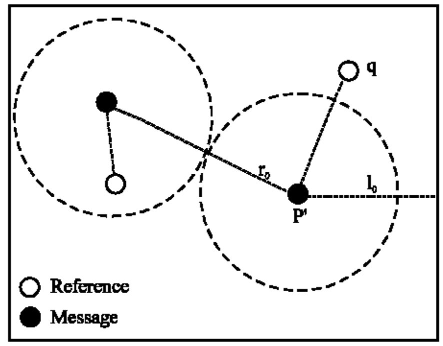

Fig. 4: Find corresponding points between the message patch and the reference patch

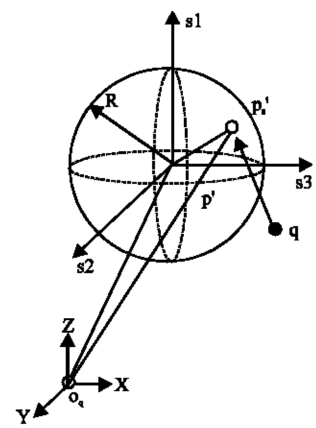

Fig. 5: The virtual sphere and the embedding point p_s'. The message point q shifts to the corresponding embedding point p_s' when embedding information

We construct the local coordinate system with the point p' as origin for the virtual sphere (Fig. 5). The three principal axes of s1, s2 and s3 coincides with the z, y and x axis, respectively. Based on the local coordinate system and virtual sphere, the point p_s' can be represented by its spherical coordinate (r, θ, ϕ) (Fig. 6), where, r is the distance between p' and P', θ is the angle between $\overrightarrow{p'p_s'}$ and s1, ϕ is the angle between $\overrightarrow{p'p_s^{p}{}'}$ and s2 where, $p^p_s{}'$ is the projection point of p' . In terms of (r, θ, ϕ), the coordinate of the point $p_s'(r, \theta, \phi)$ can be calculated as:

$$p_s'(r,\theta,\phi) = o_q + \overrightarrow{o_q p'} + r\cos(\theta)s1 + \sin(\theta)\sin(\phi)s2 + r\sin(\theta)\cos(\phi)s3 \quad (3)$$

where, r∈(0, R], θ∈ [0, π], ϕ∈ [0, 2∈).

Extending the QIM concept to the r, θ, ϕ directions of virtual sphere, we define embedding points $p_s'(r_\omega, \theta_\alpha, \phi_\beta)$ in terms of their local spherical coordinate as:

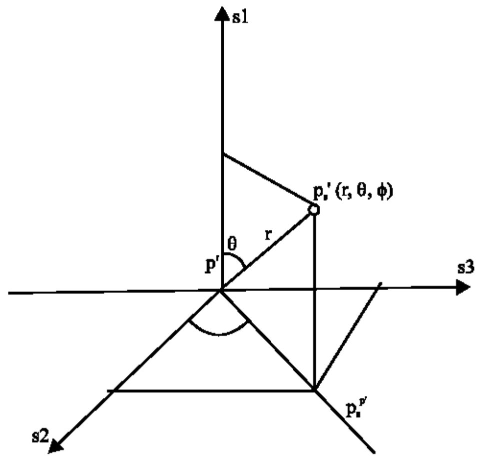

Fig. 6: Local spherical coordinate for the embedding point

$$\begin{cases} r_\omega = \dfrac{R}{\rho}(\omega+1) \ (\omega \in \{0,1,\ldots,\rho-1\}) \\ \theta_\alpha = \dfrac{\pi}{m}(\alpha+0.5) \ (\alpha \in \{0,1,\ldots,m-1\}) \\ \phi_\beta = \dfrac{2\pi}{n}(\beta+0.5) \ (\beta \in \{0,1,\ldots,n-1\}) \end{cases} \quad (4)$$

where, $s = m\times n\times\omega+n\times\alpha+\beta$; ρ, m, n be the interval ratio in the r, θ, ϕ directions of the virtual sphere, respectively; r_ω, θ_α, ϕ_β be the local spherical coordinate of the embedding point resolved by the parameter ω, α, β, respectively.

Since, the virtual sphere is equally divided by the local principle coordinate plane, we set ρ to 2, m to 2 and n to 4. Because there have sixteen embedding pints ($\rho\times m\times n = 16$), The embedding points p_s' (r_ω, θ_α, ϕ_β) can then be encoded with four bits, since $\log_2 = (\rho\times m\times n) = \log_2 16 = 4$. For example:

$$p_0'(\frac{R}{2},\frac{\pi}{4},\frac{\pi}{4}),\ p_8'(R,\frac{\pi}{4},\frac{\pi}{4})$$

are two embedding points when we set $\omega = 0$, $\alpha = 0$, $\beta = 0$ and $\omega = 1$, $\alpha = 1$, $\beta = 1$, respectively (Fig .7). We define the message state M_i of each embedding point p_s' (r_ω, θ_α, ϕ_β) ($s\in \{0, 1.., (\rho\times m\times n\text{-}1)\}$) as i = s.

Now, we consider the message patch Q. Assume that the message point q in the message patch Q be the point-to-point correspondence point of reference point p'. Let i represent the secret message that we intend to embed on the message point q . As described, embedding this message into point q leads to shift the message point to an appropriated embedding point p' (r_ω, θ_j, ϕ_k), which has the same message state as the secret message (Fig. 5). We can determine this embedding point using Eq. 5.

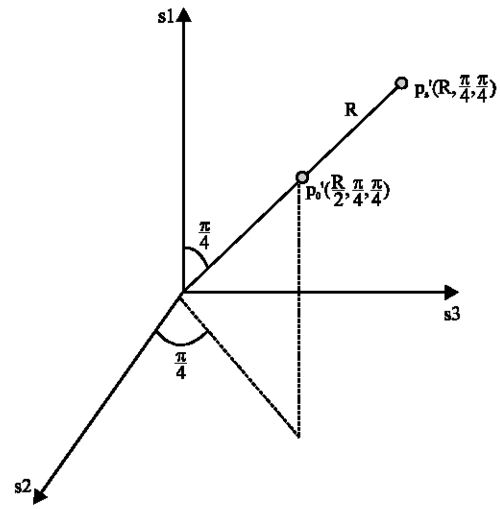

Fig. 7: Two embedding points: $p_0'(\frac{R}{2},\frac{\pi}{4},\frac{\pi}{4})$ and $p_8'(R,\frac{\pi}{4},\frac{\pi}{4})$

$$\begin{cases} \text{if } \left|qp_i'(r_\omega,\theta_j,\phi_k)\right| \le \varepsilon : \text{No point shift is needed.} \\ \text{if } \left|qp_i'(r_\omega,\theta_j,\phi_k)\right| > \varepsilon : \text{We need to shift q to the embedding point} \\ \qquad p_i'(r_\omega,\theta_j,\phi_k)\text{, replace the coordinate values} \\ \qquad \text{of the q with that of the } p_i'(r_\omega,\theta_j,\phi_k). \end{cases} \quad (5)$$

with the appropriate value for ε (for example, 10^{-6}).

Finally, after all points in the message patch have been processed by the SPM, we use inverse transformation of patch orientation and translation matrix to recover the embedded message patch to its original position and orientation.

In fact, the method is not limited to embedding four bits per message point. The real limitation is data representation precision. For instance, when we equally divide the spherical coordinates (r, θ, ϕ) of virtual sphere into M, N, P ($M\ge 2$, $N\ge 2$, $P\ge 2$) parts, respectively, we can embed log ($M\times N\times P$) bits per message point.

Let r_0 be the 1/2 of the minimum distance from the point p' to all the points in the reference patch P' . Let l_0 be the minimum distance from the point p' to the cell boundaries (Fig. 4). The threshold value R is set to $R = \alpha = \min(r_0, l_0)$ where, α is the modulation amplitude ratio, which ensures that the shifting of the message point will not cross the cell boundaries.

Note that we use a pseudo sequence to traverse the points in the reference patch and construct the embedding list which is maintained as a secret key for both the embedding and extraction procedures.

The shifting of the message point keeps the feature of the smallest 3D Euclidian distance between the reference point and the embedding point, which should not change the point-to-point correspondence between them. It is impossible to distort the embedding list in the extraction. Since the codebook is used in both the embedding and extraction, it is a side-match steganography.

Information extracting: During extraction, the following steps are performed. First, the stego model is analyzed using the PCA technique to obtain the orientation. Then we partition the stego model into patches using octree and the saved number of subdivision. Finally, with the help of codebook and the secret keys, we extract the message using SSPM procedure with point matching and spatial position matching just the same as embedding procedure.

EXPERIMENTAL RESULTS

We implement the proposed technique using C++ programming language and performed experiments to validate the feasibility of our approach. Results are collected on a personal computer with an Intel Pentium Dual-Core CPU 2.0 GHz processor and 2 GB memory.

In this study, one of our main goals is capacity. We assume no robustness requirements, except basic operations of 3D point cloud models such as affine transformations, which include translation, rotation, uniform scaling, or their combined operations. For simplicity, we only hide four bits per message point in the spatial domain using the SSPM. Nevertheless, it is possible for our method to hide a message with a capacity of more than four bits per point except the points of the reference patches; this depends on the data representation precision. No errors were found in the recovered messages, even when we applied some arbitrary affine transformations.

As suggested by Cheng *et al.* (2006) we evaluate the distortion using Eq. 6.

$$\frac{\max(RMS(U,C),RMS(C,U))}{d_B} \quad (6)$$

where, U is the original point cloud and C the steganographic point cloud, d_B is the bounding box diagonal of U and RMS is the symmetric root mean square distance which measures Hausdorff distance between discrete point-pairs.

Since, the distortion and capacity are influenced by patch size and codebook, we show the correlation among them. We define codebook ratio as the ratio of the original number of patches vs. the number of patches in the codebook to measure codebook size (Hubo *et al.*, 2008). Codebook ratio can be influenced by changing the maximal search distance of the nearest neighbor query in the patch space. Since a larger codebook which represents shorter patch chains and more reference patches decreases the capacity, patches should be as big as possible to reduce codebook. However, using big patches results in low capacity because big patches cannot find similar patches as easy as small patches.

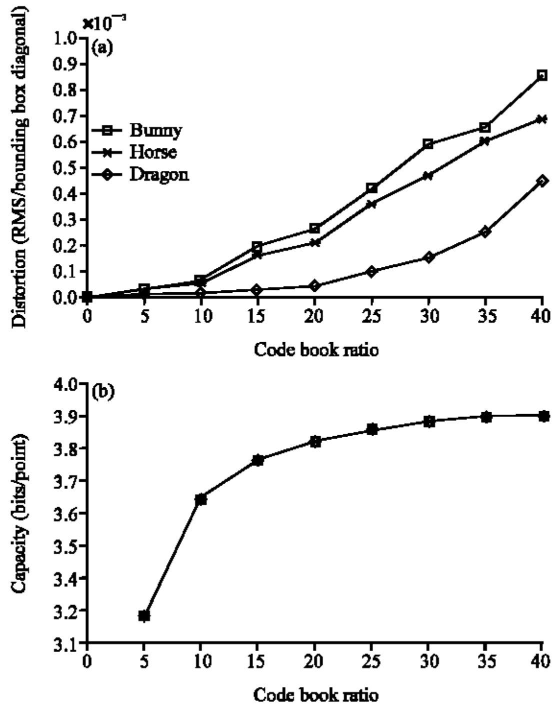

Fig. 8: Relation between code book ratio and distortion and code book ratio and capacity for different models with the same patch size = 60. (a) Distortion with code book ratio (patch size 60) and (b) capacity with code book ratio (patch size 60)

Theoretically, patches should be as big as possible but still remain a unit to express self-similarity. However, the optimal patch size is totally dependent on the geometry of individual model. Let g be the codebook ratio and v be the number of points in the model. The theoretical maximal capacity (M) of our scheme can be simply stated by Eq. 7.

$$M = 4 \times \frac{g-1}{g} \times v \quad (7)$$

We have tested our steganographic scheme on several models with patch size being equal to 60 points: bunny, dragon and horse. Figure 8a shows the relation between the code book ratio and distortion. Figure 8b shows the relation between the code book ratio and the capacity. The curves indicate a larger code book ratio has

Table 1: Results of various models. For testing, we embedded 3 bits per point except points of reference patches and chose patch size = 60, code book ratio = 15

Model	Points	Embedded messages (bits)	Distortion	Segmentation	Time cost (sec) Clustering	Embedding
Bunny	35947	130152	1.93×10^{-4}	0.29	10.98	1.43
Horse	48485	180972	1.58×10^{-4}	0.26	11.54	1.86
Dinosaur	56194	209748	2.17×10^{-4}	0.35	14.86	2.38
Teeth	116604	435164	7.94×10^{-4}	0.41	23.63	4.15
Elephant	148688	555060	8.43×10^{-4}	0.48	32.32	5.32
Dragon	437645	1632956	2.78×10^{-4}	1.36	88.35	10.87

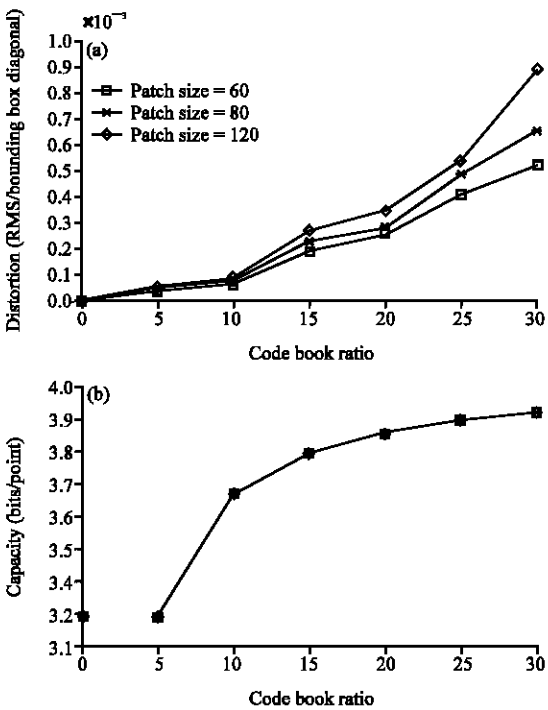

Fig. 9: Relation between code book ratio and distortion and code book ratio and capacity for bunny using different patch sizes. (a) Distortion with code book ratio (patch size 60) and (b) capacity with code book ratio (patch size 60)

a smaller codebook size in return for a higher capacity, but leads to more distortion.

We also tested our scheme for the dragon model using three different patch sizes for patch size influences the codebook ratio. The results are shown in Fig. 9a and b. The curves show that the capacity almost keeps unchanged with the various patch size but the distortion is dependent on the chosen patch size.

Table 1 presents the model details, the embedded message, model distortion and processing time under the specified conditions. The distortion values are rather small, indicating insignificant loss of quality for the stego models.

Visual results of the cover and stego models of the bunny and horse models are shown in Fig. 10 and 11, respectively. The visual appearance of images shows insignificant distortion for the stego models.

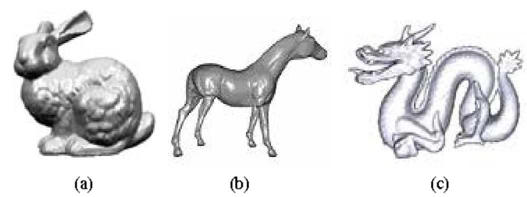

Fig. 10: The cover models for (a) Bunny, (b) Horse and (c) Dragon

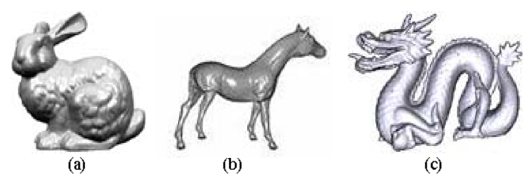

Fig. 11: The stego models for (a) Bunny, (b) Horse and (c) Dragon

From the security point of view, the guarantee of security of our scheme comes from four keys: patch size which influences the partition, codebook size which is controlled by the maximal search distance of the nearest neighbor query in the patch space, the traversal list of codebook and embedding list over each patch which is controlled by pseudo-random sequence. It is difficult for attacks to obtain four keys even they use exhaustive search. Retrieving the message without the keys is virtually impossible. So our scheme has high security in the sense of cryptography.

We also estimate the complexity of our algorithm by giving execution times for various models, as shown in Table 1. The total time cost of our algorithm can be divided in three parts: the segmentation phase, the clustering phase and the embedding phase. The first phase finishes less than two second. The second phase, on the other hand, may take comparatively long time from ten seconds to ninety seconds according to the number of points. The third phase also takes time from one second to ten second. Note that although the segmentation and clustering may take comparatively long time, their result can be reused to achieve different

Table 2: The comparison of the five related methods. The capacity is denoted as average bits per point

	Cheng *et al.* (2006)	Cotting *et al.* (2004)	Wang and Wang (2005)	Luo *et al.* (2006)	Our method
Capacity	≈3	≈1	≈0.5	<1	≈4
Extraction	Blind	Blind	Blind	Blind	Blind
Robustness	Affine	Affine	Affine	Affine	Affine
Domain	Spatial	Transform	Transform	Transform	Spatial

embedding capacity. As the actual embedding time cost of our scheme is low, the codebook can be constructed and stored in database in advance to reduce time cost and improve the performance of steganography.

Finally, Table 2 offers a detailed comparison of the five related steganographic methods for 3D point cloud, which include (Cheng *et al*., 2006; Cotting *et al*., 2004; Wang and Wang, 2005; Luo *et al*., 2006). All of them are robust against affine transformations and can extract messages without the assistance of the cover model. As mentioned previously, our approach goes one step further to achieve side-match steganography for 3D point cloud. Moreover, our approach offers a large improvement in capacity with little distortion, compared to the previous 3D point cloud steganographic methods.

CONCLUSIONS

This study presents a high-capacity spatial blind steganographic approach for 3D point cloud models. A new methodology has been developed to construct self-similarity patch chains and embed messages to every matching point using proposed SSPM procedure. Our technique provides steganography with high capacity, security, low distortion and robustness against affine transformations.

The main remarkable features of the proposed scheme include: (1) PCA-based preprocess and octree-based partition uniquely segment 3D point cloud models to patches, which provides the robustness of affine attacks. (2) Reference patches which are selected from constructed similarity patch chains and codebook are side-match information for steganography. (3) The scheme efficiently gains high capacity that every point except points of reference patches can embed at least four bits using the proposed SSPM. (4) Optimal matching among the message points and reference points by refined matching procedure decreases the distortion. In addition, the secret keys of patch size, codebook size, the traversal list of codebook and embedding list over each patch provide more security, recovering messages without assistance of the secret keys is really impossible.

To the best of our knowledge, this is the first side-match steganographic scheme for 3D point cloud that uses the spatial point marching to embed messages. Our approach is intuitive and can achieve high capacity with little distortion.

ACKNOWLEDGMENTS

We thank C.A. Zhao and J. Liu for their helpful discussions. Present work was supported by National High Technology Research and Development Program of China (Grant no. 2009AA012420) and Natural Science Fund of Guangdong Province (Grant No. 9151009001000059), China.

REFERENCES

Ai, Q.S., Q. Liu and Z.D. Zhou, 2009. A new digital watermarking scheme for 3D triangular mesh models. Signal Process., 89: 2159-2170.

Aspert, N., E. Drelie, Y. Maret and T. Ebrahimi, 2002. Steganography for three-dimensional models. Proc. SPIE, 4790: 705-708.

Cayre, F. and B. Macq, 2003. Data hiding on 3-D triangle meshes. IEEE Trans. Signal Process., 51: 939-949.

Cheng, S.C., C.T. Kuo and D.C. Wu, 2010. A novel 3D mesh compression using mesh segmentation with multiple principal plane analysis. Pattern Recognit., 43: 267-279.

Cheng, Y.M. and C.M. Wang, 2006. A high-capacity steganographic approach for 3D polygonal meshes. IEEE Trans. Visual Comp., 8: 845-855.

Cheng, Y.M., C.M. Wang, Y.Y. Tsai, C. Chung-Hsien and W. Peng-Cheng, 2006. Steganography for three-dimensional models. Lecture Notes Comput. Sci., 4035: 510-517.

Cheng, Y.M. and C.M. Wang, 2007. An adaptive steganographic algorithm for 3D polygonal meshes. IEEE Trans. Visual Comput., 9: 721-732.

Cotting, D., T. Weyrich, M. Pauly and M. Gross, 2004. Robust watermarking of point-sampled geometry. Proceedings of International Conference on Shape Modeling and Applications, June 7-9, Palazzo Ducale, Genova, Italy, pp: 233-242.

Hubo, E., T. Mertens, T. Haber and P. Bekaert, 2008. Self-similarity based compression of point set surfaces with application to ray tracing. Comput. Graphics, 32: 221-234.

Johnson, A.E. and M. Hebert, 1999. Using spin-images for efficient multiple model recognition in cluttered scenes. IEEE Trans. Pattern Anal. Machine Intelligence, 21: 433-449.

Johnson, N.F. and S. Jajodia, 1998. Exploring steganography: Seeing the unseen. Computer, 31: 26-34.

Luo, H., J.S. Pan, Z.M. Lu and H.C. Huang, 2006. Reversible data hiding for 3D point cloud model. Proceedings of the International Conference on Intelligent Information Hiding and Multimedia Signal Processing, Dec. 18-20, Pasadena, California, USA., pp: 863-867.

Macqueen, J.B., 1967. Some methods for classification and analysis of multivariate observations. Proc. 5th Berkeley Symp. Math. Statist. Prob., 1: 281-297.

Maret, Y. and T. Ebrahimi, 2004. Data hiding on 3D polygonal meshes. Proceedings of the Workshop on Multimedia and Security and International Multimedia Conference, Sept. 20-21, Magdeburg, Germany, pp: 68-74.

Pauly, M., L.P. Kobbelt and M. Gross, 2002a. Multi-resolution modeling of point-sampled geometry. CS Technical Report No. 378, September 16, ETH Zurich.

Pauly, M., M. Gross and L.P. Kobbelt, 2002b. Efficient simplification of point sampled surfaces. Proceedings of the Conference on Visualization, Nov. 1, IEEE Computer Society, Washington, DC, USA., pp: 163-170.

Rencher, A.C., 2002. Methods of Multivariate Analysis. 2nd Edn., John Wiley and Sons, New York, pp: 451-503.

Rusinkiewicz, S. and M. Levoy, 2001. Efficient variants of the ICP algorithm. Proceedings of the 3rd International Conference on 3D Digital Imaging and Modeling, May 28-Jun. 1, Quebec City, Canada, pp: 145-152.

Tamy, B., R. Patrick and S. Christophe, 2005. Visualization of point-based surfaces with locally reconstructed subdivision surfaces. Proceedings of the International Conference on Shape Modeling and Applications IEEE, June 15-17, Cambridge, MA, USA., pp: 23-32.

Wang, C.M. and P.C. Wang, 2005. Data hiding approach of point-sampled geometry. IEICE Trans. Comm., E88: 190-194.

Wang, C.M. and Y.M. Cheng, 2005. An efficient information hiding algorithm for polygon models. Comput. Graph. Forum, 24: 591-600.

Zafeiriou, S., A. Tefas and I. Pitas, 2005. Blind robust watermarking schemes for copyright protection of 3D mesh objects. Trans. Visual. Comput. Graph., 11: 596-607.