Sensitivity Analysis of Euclidean Minimum Spanning Tree

N.Tournier[a, b], W. Puech[a], G. Subsol[a] and J-P. Pedeboy [b]

[a] LIRMM Laboratory, Univ. Montpellier II, CNRS
161 rue Ada, 34392 Montpellier, FRANCE;
[b] Strategies S.a, 41-43 rue de Villeneuve, Parc des Affaires SILIC - BP 80429
94583 Rungis, FRANCE

ABSTRACT

In 3D steganography and watermarking, the synchronization of the hidden data is a major problem. We need to know where the message is embedded in order to extract the correct information. Various algorithms have been proposed for the last couple of years and we focused on a method based on Euclidean minimum spanning tree (EMST) for the mesh vertices. In this paper we analyze the sensitivity of the EMST structure in order to propose a new method more robust. We present a new theoretical analysis and we propose to visualize the robustness of the EMST. Moreover, we can apply this analysis to various applications that can be useful in 3D steganography such fragile area detection and prediction of the 3D object robustness during transmission on a noisy channel.

Keywords: 3D steganography, watermarking, graph theory, Euclidean minimum spanning tree, sensitivity analysis

1. INTRODUCTION

Internet is very useful to broadcast multimedia information. There are more and more 3D object exchange in infography, CAO for example. It's important on the one hand, to protect the file content and on the other hand, to embed metadata. Data hiding may be a solution.

A recent survey of 3D watermarking techniques was proposed by Wang *et al.*[1] Let define a 3D object as a mesh $\mathcal{M} = (\mathcal{V}, \mathcal{F})$, with $\mathcal{V}$ the cloud of n vertices and $\mathcal{F}$ the set of facets. It is important to note, when we hide a message in the mesh we need to locate where the binary data is distributed. This is the synchronization process which is the main difficulty in 3D data hiding. Various methods have been proposed, and we interest in the synchronization by EMST computing proposed by Amat *et al.*[2]

An EMST is a minumum spanning tree (MST), a tree $T = (V, E_T)$ that joins all the vertices of a graph $G = (V, E)$ using the edges $e_i = \{v_s, v_t\} \in E$, with a weight $\omega(e_i) \in \mathbb{R}^+$, that minimize the total weight $\sum_{e_i \in E_T} \omega(e_i)$.

One of the original techniques to synchronize the message is proposed by Amat *et al.*[2] It is a method that does not move any vertex of the mesh, and they propose to use an EMST to be able to cover the mesh in a unique manner for the synchronization of the message. We can see Figure 1.a a representation of a mesh and its corresponding EMST Figure 1.b.

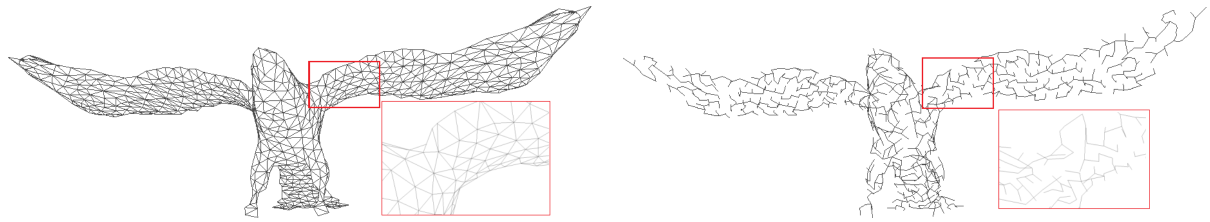
To embed a message into the mesh, they analyze for quadruples in the EMST that verify different conditions of planarity, convexity and overlapping. These quadruples are used for the data hiding and they are detected after the watermarking to extract the message. But this method is very fragile, so we are interested in the analysis of the sensitivity of EMST in order to locate the most robust vertices in the mesh.

As the method is fragile, the sensitivity analysis of EMST may be a solution to find a method more robust. We propose to compute how a vertex can be moved without changing the connexions in the EMST. Since the problem is very difficult, to make it tractable we propose to make some simplifications.

In this paper, we expose in Section 2 previous work on sensitivity analysis of minimum spanning tree (MST). In Section 3 we propose a theoretical sensitivity analysis of EMST for geometrical data. Then, in Section 4 we present experimental results of our approach. To conclude, in Section 5, we discuss about our future prospects.

Further author information:
E-mail: {nicolas.tournier, william.puech, gerard.subsol}@lirmm.fr, Telephone: +33 (0)4 67 41 85 85

Three-Dimensional Image Processing (3DIP) and Applications, edited by Atilla M. Baskurt,
Proc. of SPIE-IS&T Electronic Imaging, SPIE Vol. 7526, 75260G · © 2010 SPIE-IS&T
CCC code: 0277-786X/10/$18 · doi: 10.1117/12.839455



(a) (b)

Figure 1. 3D representations of: a) A 3D Mesh, b) Its corresponding EMST computed by the Prim's algorithm.[3]

2. PREVIOUS WORK

According to Gordeev[4,5] the sensitivity analysis of minimum spanning tree (MST) is an optimization problem based on matroids. Let $G = (V, E)$ a graph with n nodes and m edges. Gordeev considers this following model. He lets $D_m = \{\tau_1, \ldots, \tau_q\}$ with $(q > 1)$ a system of subsets of E that he calls trajectories; $A = (a_1, \ldots, a_m) \subset \mathbb{R}^m$ such as $\forall i \; a_i = \omega(e_i)$, the weight of the edges of the graph G and $\tau(A)$ a functional called the trajectory length for A such as $\tau(A) = \sum_{e_i \in \tau} a_i$.

Therefore, the combinatory problem is defined with the pair (E, D_m), A is the variable to optimize in order to minimize the functional $\tau(A)$. With this model, Gordeev modelizes the MST problem with D_m the set of all spanning trees of G in which the MST is a trajectory that minimizes the functional $\tau(A)$.

Let $\psi(A)$ the indice set i of the optimal trajectories τ_i of the problem for a given A, and $B \in \mathbb{R}^m$ such as $\|B\| < \epsilon$ a perturbation vector. Gordeev talks about ϵ-stability when $\psi(A + B) \subset \psi(A)$.

In the MST problem, for a given noise intensity ϵ it exists some MST that are always solution of the MST problem after the perturbation. He deduces a stability radius $\rho(A) = \sup \epsilon$ such as A is ϵ-stable for the problem. Its algorithm is polynomial and its complexity is $O(n^3 m \log(\frac{n^2}{m}))$ and so for a complete graph $O(n^5)$).

In Dixon *et al.*,[6] the sensitivity analysis problem is to calculate, $\forall e_i \in E$ by how much $\omega(e_i)$ can change without affecting the minimality of the MST $T = (V, E_T)$. They divide the problem in two parts.

First, $\forall e_i \notin E_T$, they compute how much they can decrease $\omega(e_i)$ without changing the MST and, for all the edges $e_i \in E_T$, they compute how much they can raise the weight of $\omega(e_i)$ without changing the MST with a linear time complexity as a function of the number of edges.

In this Section, we have exposed two approaches to analyze the sensitivity of MST. We remind us of our data are geometrical, so:

- $G = (V, E)$ is a complete graph in the sense that each vertex is linked to all the others. This implies $m = n(n - 1)$ and then maximizes the complexity of classical graph processing algorithms;
- the $\omega(e_i)$ are not independent. If $\omega(e_i)$ is pertubated, it implies that at least one of the two vertices of $e_i = \{v_s, v_t\}$ is perturbated and then that the $(n - 2)$ edges coming from this vertex are perturbated.

As for Dixon *et al.* and Gordeev, the weights $\omega(e_i)$ are independant, that is why their propositions are not applicable in our case. So we propose a new approach in order to analyse the sensitivity of EMST based on geometrical data.

3. THEORETICAL SENSITIVITY ANALYSIS

Let a weighted graph $G = (V, E)$, with V the set of n nodes and E the set of edge such as:

- Each vertex is a node $v_i \in V$;
- $\forall v_s, v_t \in V$; $v_s \neq v_t$; $\exists e_i = \{v_s, v_t\} \in E$ and $\omega(e_i) = d(v_s, v_t)$, the Euclidean distance between the two vertices in the mesh.

3.1 Minimum spanning tree problem: reminders and notations

For a given connected weighted graph $G = (V, E)$, a representation of a 3D mesh $\mathcal{M}$, we propose to compute the EMST $T = (V, E_T)$. The MST computing is a well known polynomial problem. We choose to use the Prim's algorithm[3] because it is an incremental algorithm, and at each step we have a tree and a stable.

The algorithm starts from a given node $v_0 \in V$. At the i^{th} step, it adds to the current tree the closest node v_i between the tree and the remaining vertices. Thanks to this remark, we build a tree sequence $(T_i)_{0<i<n} = (V_i, E_i)_{0<i<n}$ such as $(V_i)_{0<i<n}$ is a sequence of node set and $(E_i)_{0<i<n}$ a sequence of edge set:

- $V_0 = \{v_0\}$;
- $E_0 = \emptyset$;
- $V_i = V_{i-1} \cup \{v_i\}$ $(\forall i < n)$, with $v_i \in \overline{V_{i-1}} = V \backslash V_{i-1}$ the closest vertex of V_{i-1}, defined in equation (1) ;
- $E_i = E_{i-1} \cup \{v_i, f(v_i)\}$ $(\forall i < n)$, with $f : V_i \to V_{i-1}$ a function that compute the closest node of v_i in V_{i-1} defined in equation (2).

$$v_i = \operatorname*{Argmin}_{v_j \in \overline{V_{i-1}}} \min_{v_k \in V_{i-1}} \omega(v_j, v_k); \tag{1}$$

$$f(v_i) = \begin{cases} \operatorname*{Argmin}_{v_j \in V_{i-1}} \min_{v_k \in \overline{V_{i-1}}} \omega(v_j, v_k), & i > 0 \\ v_0, & i = 0. \end{cases} \tag{2}$$

With these notations, the ESMT of G is $T = T_{n-1}$ and we propose, in this section, to study its sensitivity. In other words, we want to know how we can move a vertex in space such as the EMST of the graph representation does not change. It is a difficult problem because the graph G stands on a geometrical data. To make this problem tractable we propose to make some simplifications.

3.2 Sensitivity analysis problem and simplifications

Given a graph $G = (V, E)$, we compute its corresponding EMST at the step i of the Prim's algorithm $T_i = (V_i, E_i)$. We move the last vertex v_i in the 3D space, and denote the new vertex v^*. The graph G is then modified, and noted G^* the new one. Moreover, $\forall i$ $T_i^* = (V_i^*, E_i^*)$ the EMST computed by the Prim's algorithm at the step i.

As it is too complex to analyze the sensitivity of the EMST with respect to disruptions applied to all the vertices at the same time, we make the following simplification assumptions on the disruption of the vertices to make the problem tractable:

- At the step $i > 0$, we will disturb only the position of the vertex v_i, resulting in the disturbed vertex v^*. Moreover, we denote by G^* the new resulting graph and its EMST $T^* = (V^*, E^*)$;
- This geometric disruption will be restricted only to be along the half-line $]f(v_i), v_i)$;

Along the half-line $]f(v_i), v_i)$, we want to know how the vertex v_i can come up to $f(v_i)$ and how can it move away from $f(v_i)$ without changing the EMST at the step i.

With the previous notations, we suppose that T_i^* verify: $\forall k$, $k < i$; $T_k = T_k^*$. In other words, the connexions in the EMST are not modified until the step $(i-1)$. At the step i, we want:

1. $v^* = v_i^*$, the vertex selected at the step i is always selected at this step after the disruption;
2. $f(v_i) = f(v_i^*)$, the 'father' of the vertex selected at the step i is the same.

3.2.1 Minimum distance limit

Let $f(v_i)$, the closest vertex of v_i in V_{i-1}. Let V_k $(k < i)$ the smallest vertex set such as $f(v_i) \in V_k$ and $f(v_i) \notin V_{k-1}$ (see Figure 2). By composition, thanks to the Prim's algorithm, we affirm that all the edges of the EMST $e_j \in E_{i-1} \backslash E_k$ have a weight $\omega(e_j)$ less than $\omega(f(v_i), v_i)$ (equation (3)). The demonstration is presented in Section A.1.

$$\forall e_j = \{v_s, v_t\} \in E_{i-1} \backslash E_k; \omega(e_j) < \omega(f(v_i), v_i). \tag{3}$$

Therefore, we deduce the minimum distance limit d_i^- between v_i and $f(v_i)$:

$$d_i^- = \max_{e_j \in E_{i-1} \backslash E_k} \omega(e_j). \tag{4}$$

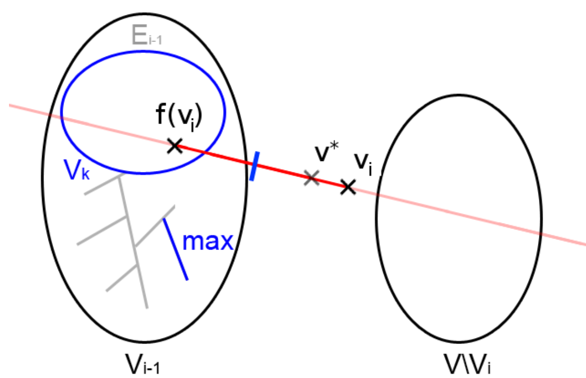


Figure 2. Scheme of the minimum limit computing.

So, to keep the same EMST, v^* must verify:

$$\omega(f(v^*), v^*) > d_i^-. \tag{5}$$

3.2.2 Maximum distance limit

The computing of the maximum distance limit is divided in two parts. First, in order to select the vertex v^* at the step i in the Prim's algorithm, we are looking for the second closest vertex of V_{i-1} denoted by $s(v_i)$ (equation (6)). Obviously, if v_i moves away from $f(v_i)$, it can be shift too far, otherwise it will be $s(v_i)$ the closest vertex of V_{i-1}. This situation is illustrated Figure 3.a.

$$s : V_i \rightarrow \overline{V_i}.$$

$$s(v_i) = \begin{cases} \underset{v_j \in V \backslash V_i}{\text{Argmin}} \min_{v_k \in V_{i-1}} \omega(v_j, v_k), & i < n-1 \\ v_{n-1}, & i = n-1. \end{cases} \tag{6}$$

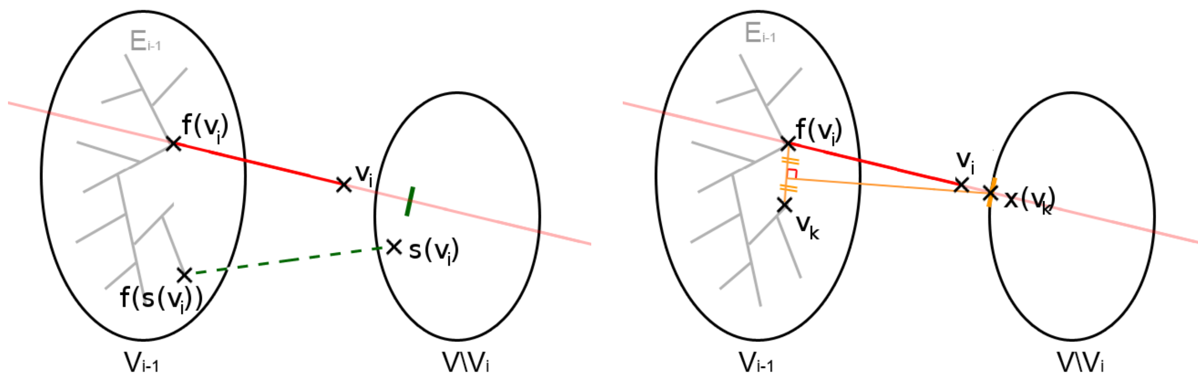
We need to know the distance between $s(v_i)$ and V_{i-1}, that is given by the edge $\{f(s(v_i)), s(v_i)\}$. Thus, we have a first candidate to the maximum distance computing, denotes by d_i^1:

$$d_i^1 = \omega(s(v_i), (f \circ s)(v_i)). \tag{7}$$

Secondly, we need to have $f(v_i) = f(v^*)$, so the vertex $f(v_i)$ must be the closest vertex of v^*. That is why for each vertex $v_k \in V_{i-1}$ we compute the intersection $x(v_k) \in \mathbb{R}^3$ between the half-line $]f(v_i), v_i)$ and the perpendicular bisector of the segment $[f(v_i), v_k]$, illustrated Figure 3.b.

Obviously, we have: $d(f(v_i), x(v_k)) = d(v_k, x(v_k))$ and if $\omega(f(v_i), v^*) > d(f(v_i), x(v_k))$ so v_i^* is closer from v_k than $f(v_i)$. So, v^* must verify for each $v_k \in V_{i-1}$ $\omega(f(v_i), v^*) < d(f(v_i), x(v_k))$. The other candidate to the maximum distance computing, denoted by d_i^2:

$$d_i^2 = \min_{v_k \in V_{i-1}} \omega(p(v_i), x(v_k)). \tag{8}$$



(a) (b)

Figure 3. Scheme of the maximum limit computing: a) Research of the second closest vertex, b) Intersection between perpendicular bisectors and the half-line $]f(v_i), v_i)$.

We compute two limit distances, one to select v^* at the step i of the Prim's algorithm and the other to $f(v_i) = f(v^*)$. We need to verify these two conditions, so the maximum limit distance is the minimum between d_i^1 and d_i^2, and v^* must verify the equation (10). The demonstration is described in Section A.2.

$$d_i^+ = \min\{d_i^1, d_i^2\}. \tag{9}$$

$$\omega(f(v^*), v^*) < d_i^+. \tag{10}$$

4. EXPERIMENTAL RESULTS

In this section, we propose to analyze statically the minimum and maximum distance limits presented in Section 3.2. In Section 4.1 we describe the experimental conditions, then we present a full an example with the 3D object *Angel*, illustrated in Figure 6.a and 6.c to comment the results and we introduce in Section 4.3 a first analysis by viewing the robust areas on the 3D object *Angel*. In Section 4.4, we apply two normalizations to compare the impact on our measures.

4.1 Experimental conditions

For the experiments we have used a database composed of 14 3D meshes selected from various sources (Stanford University Graphics Laboratory[*], MADRAS project[†], Strategies S.a[‡] and Aimasharpe[§]). Their shapes are very

[*]http://www-graphics.stanford.edu
[†]http://www-rech.telecom-lille1.eu/madras
[‡]http://www.cadwin.com
[§]http://dsw.aimatshape.net

different as we can see on Figure 4 and they are used in different application field such CAD, manufactory, medicine or entertainment.

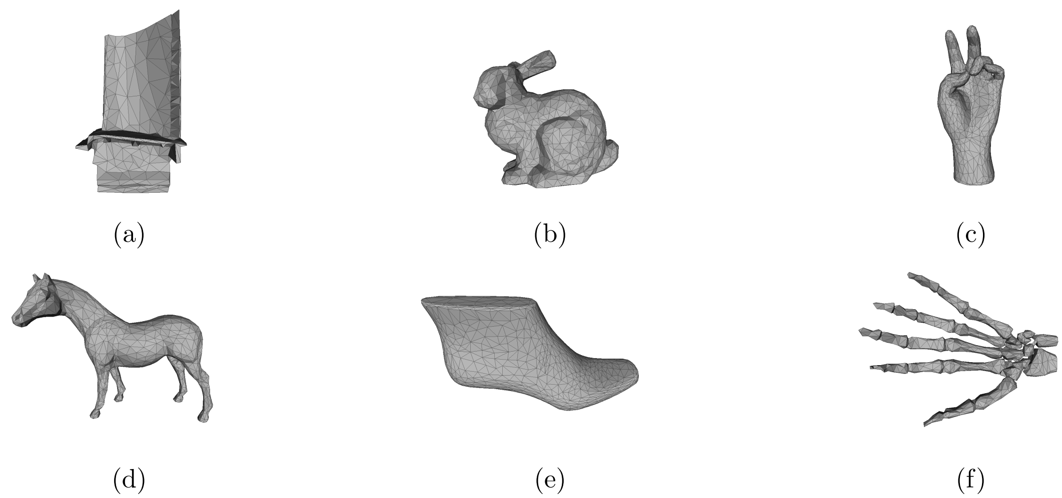


Figure 4. Selection of 3D objects: a) Blade, b) Bunny, c) Hand, d) Horse, e) Shoe, f) Skeleton.

In order to compare the meshes, we have subsampled them to have approximately the same number of vertices. For practical reasons, we choose around 1,000 vertices.

To be able to compare their disruptions, we have normalized all these objects, using two normalizations based on:

1. The size of the bounding box (normalization (1));
2. The average distance between two vertices in the mesh (normalization (2)).

Then we compute for each normalized mesh and for each vertex, the upper radius r_i^+ and the lower one r_i^-:

$$r_i^+ = d_i^+ - \omega(f(v_i), v_i); \tag{11}$$

$$r_i^- = \omega(f(v_i), v_i) - d_i^-. \tag{12}$$

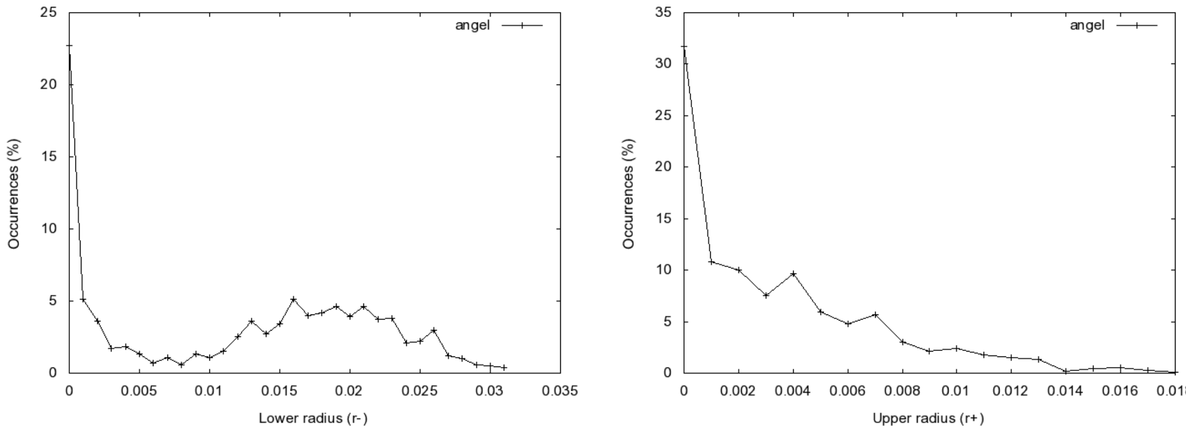
4.2 A full example based on *Angel*

The results of our analysis are presented for the mesh *Angel* normalized by the normalization (1) illustrated Figures 6.a. and 6.c. The Figures 5.a and 5.b represents respectively the distributions of r^+ and r^- subsampled each 0.001.

First, we remark there are many vertices that cannot be moved on one direction, more than 20% for the lower radius r^- and around 30% for the upper radius r^+. More the radius value is significant, more the vertex is robust. The results are not amazing; it reveals the fragility of the EMST structure.

For the upper radius r^+, the occurrences are decreasing very fast when the radius value increases. Lets note the maximum upper radius is around 0.018 whereas as for the lower radius the maximum is around 0.030. So it is easier to move a vertex v_i closer from its 'father' $f(v_i)$ than moving it away.

Then, for the lower radius, we can note an interesting peek between 0.015 and 0.02. With this observation, we can consider two kinds of vertex: those that cannot be moved closer from theirs 'father' and those can be moved almost as close as we want to.



(a) (b)

Figure 5. Lower and upper radius for the mesh *Angel*: a) Distribution of the lower radius r^- (sampled each 0.001), b) Distribution of the upper radius r^+ (sampled each 0.001).

4.3 Robust area visualization

We propose to visualize the more robust vertices in order to locate the most robust areas to synchronize the hidden data during the watermarking process. To connect our theoretical analysis criteria defined by r^+ and r^-, for each vertex, we have choosen to use the upper radius r_i^+ to visualize these more robust areas. Figure 6.b and 6.d illustrate the result of the visualization. The darkest colors correspond to the most fragile areas, whereas the lightest colors are for the most robust areas.

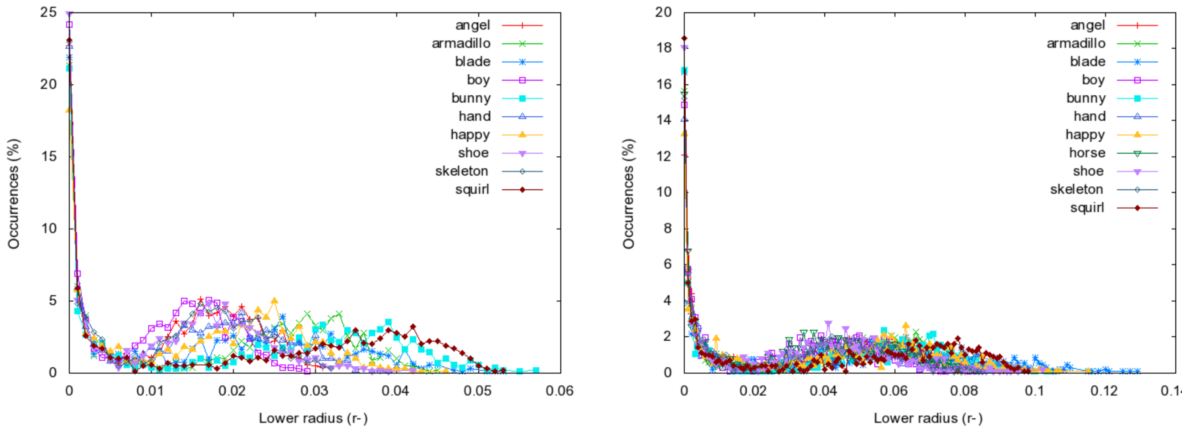
(a) (b) (c) (d)

Figure 6. Visualizations of *Angel*: a) Front view, b) Robust areas in front view, c) Behind view, d) Robust areas in behind view.

As we can see a lot of areas are fragile, illustrated in darkest color. And we can quote, for the most robust vertices, the spacing between itself and its neibourhood is not regular. It might be a lead for the research of geometrical criterion.

4.4 Impact of the normalization

Figure 7 illustrates the distributions of the lower radius r^- for the 3D objects of the database. On Figure 7.a, the objects are normalized by normalization (1) and on the Figure 7.b by the normalization (2).



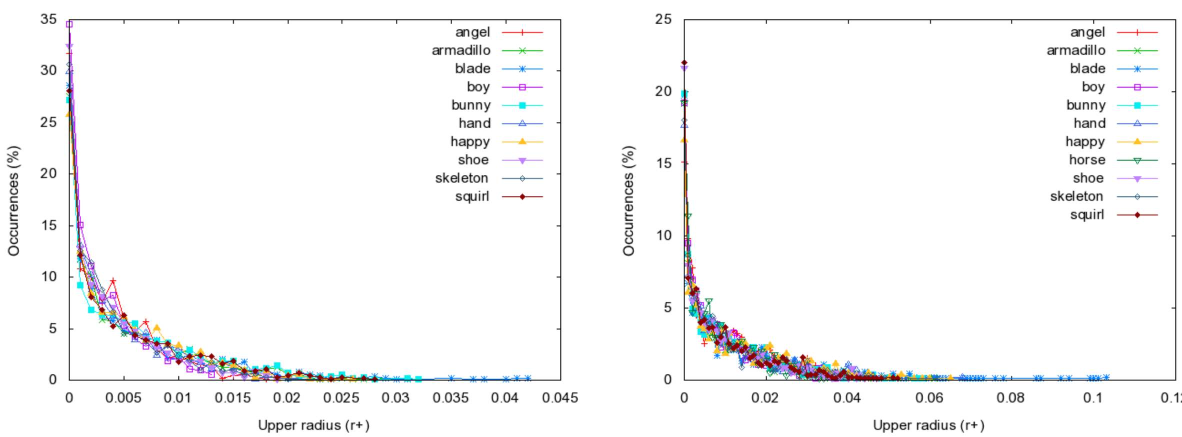
(a) (b)

Figure 7. Distribution of the lower radius (sampled each 0.001) for the normalization: a). As a function of the bounding box size, b) As a function of the average distance between two vertices in the mesh.

We remark the shape of the plots are globaly similar. For the normalization (2) the peak is almost at the same value. Otherwize, for the normalization (1), for each object, the peak is not positioned at the same value.

The normalization as a function of the size of the bounding box is more discriminatory for this criterion and it might be a robustness criterion.

Figure 8 shows the distributions of the upper radius r^+ for some 3D objects. As previously, Figure 8.a illustrates the results for the normalization (1) and Figure 8.b for the normalization (2).



(a) (b)

Figure 8. Distribution of the upper radius (sampled each 0.001) for the normalization: a). As a function of the bounding box size, b) as a function of the average distance between two vertices in the mesh.

Except the maximum upper radius for the normalization (2) that is 2 at 3 times upper the other normalization, the plots are also globaly similar.

5. CONCLUSION AND PERSPECTIVES

To conclude, we propose a new approach to analyse the sensitivity of the EMST based on geometrical data. We need to make some simplification in order to make the problem more treatable. For each vertex v_i, we consider it can move only along a streght half-line $]f(v_i), v_i)$, with $f(v_i)$, the 'father' of v_i in the EMST according to the Prim's algorithm. And we compute how v_i can be closer and away from $f(v_i)$, these distances respectively correspond to the lower radius r^- and the upper radius r^+.

This method will permete us to visualize the robust areas of the object for a synchronisation by EMST computing. We expose our first results for given criteria. We have to perform this kind of appilication in order to predict the robustness of the watermarking technique.

Moreover, we are intereted in, what ahappen when we move all the vertices. For exemple, when a 3D object is transmitted on a noisy channel, some vertices are moved. So we need to make a correlation between our robust area prediction and the impact of a noise on the 3D object.

APPENDIX A. DEMONSTRATION

We suppose v_0 is always the same seed of the Prim's algorithm, and $\forall i \; V_{i-1} = V^*_{i-1}$. The proofs are decomposed in two parts:

1. $\forall i \; v^*$ is selected at the i^{th} step of the Prim's algorithm;
2. $\forall i \; f(v^*) = f(v_i)$.

A.1 Minimum limit computing

Referring to the equation (4), lets demonstrate this relation in order to keep the same EMST, according to the hypothesis of the sensitivity problem:

$$\omega(f(v^*), v^*) > d_i^- = \max_{e_j \in (V_{i-1} \backslash V_k)^2} \omega(e_j)$$

A.1.1

We denote by V_k the smallest set that contains the vertex $f(v_i)$ (i.e $f(v_i) \notin V_{k-1}$ and $f(v_i) \in V_k$); and $e_l = \{f(v_l), v_l\}$ the edge, if it exist, verifying $e_l = \max_{e_j \in (V_{i-1} \backslash V_k)^2} \omega(e_j)$. If this edge does not exists, it means $d_i^- = 0$ so we can move the vertex v_i as closed as we want from $f(v_i)$.

We suppose that it exists at least one edge e_l such as $e_l = \max_{e_j \in (V_{i-1} \backslash V_k)^2} \omega(e_j)$:

$$d_i^- = \omega(f(v_l), v_l).$$

It is important to note that $k \leq l < i$, in other words, in the chronogical vertex selection in the Prim's algorithm v_k is selected before v_l, and v_l before v_i. With this scheme, lets demonstrate *reductio ad absurdum* that the equation (4) must be verified to keep the same order in the sequence $(V_i^*)_{0<i<n}$.

Supposing $\omega(f(v^*), v^*) \leq d_i^- \Rightarrow \omega(f(v^*), v^*) \leq \omega(f(v_l^*), v_l^*)$. At the $(l-1)^{th}$ step of the Prim's algorithm we know $f(v_i^*), f(v_l^*) \in V_{l-1}$. According to the hypothesis $\omega(f(v^*), v^*) \leq \omega(f(v_l^*), v_l^*)$, so v^* will be chosen at the $(l-1)^{th}$. That is in contradiction to the stability of the EMST.

A.1.2

Obviously, v_i is the closest node of $f(v_i)$ in $V \backslash V_i$, if we move v_i along the half-line $)f(v_i), v_i)$ closer from $f(v_i)$, the resulting vertex $v^* \in V \backslash V_i$ is the closest node of $f(v_i)$. In conclusion, the 'father' of v_i is also the 'father' of v^* with this displacement.

A.2 Maximum limit computing

Referring to the equation (9), lets demonstrate this relation in order to keep the same EMST, according to the hypothesis of the sensitivity problem:

$$\omega(f(v^*), v^*) < d_i^+ = \min\{d_i^1, d_i^2\};$$

$$d_i^1 = \omega(s(v_i), (f \circ s)(v_i));$$

$$d_i^2 = \min_{v_k \in V_{i-1}} \omega(p(v_i), x(v_k)).$$

A.2.1

Lets demonstrate *reductio ad absurdum* that $\omega(f(v^*), v^*) < d_i^1$ must be verified to keep the same order in the sequence $(V_i^*)_{0<i<n}$.

We suppose $\omega(f(v^*), v^*) \geq d_i^1 = \omega(s(v_i), (f \circ s)(v_i))$. According to the hypothesis v_i is the closest vertex of V_{i-1}, and $s(v_i)$ the second one. Then, v_i is disturb in v^* but the other vertices does not move. Moreover $(f \circ s)(v_i), f(v^*) \in V_{i-1}$ and $s(v_i), v^* \in V \backslash V_{i-1}$. So according to the Prim's algorithm at the step i it choose the closest vertex of V_{i-1}, it is $s(v_i)$. v^* is to far from $f(v_i)$, so to verify the condition of our EMST stability problem $\omega(f(v^*), v^*) < d_i^1$.

A.2.2

Let $v_k \in V_{i-1}$ a vertex satisfying the relation $\omega(v_k, x(v_k)) = \min_{v_j \in V_{i-1}} \omega(v_j, x(v_j))$. Obviously, on the line $(f(v_i), v_i)$, the vertices $\{f(v_i), v_i, x(v_k)\}$ are aligned in this order.

Moreover, $x(v_k)$ is the equidistant vertex between $f(v_i)$ and v_k. Obviously it separates the half-line $]v_i;\infty)$ in two parts:

- $\forall v^* \in]v_i; x(v_k)[$, $\omega(f(v_i), v^*) < d(f(v_i), x(v_k))$, v^* is closer from $f(v_i)$ than v_k;
- $\forall v^* \in]x(v_k); \infty[$, $\omega(f(v_i), v^*) > d(f(v_i), x(v_k))$, v^* is closer from v_k than $f(v_i)$.

It proves the proposition.

REFERENCES

[1] Wang, K., Lavoué, G., Denis, F., and Baskurt, A., "A Comprehensive Survey on Three-Dimensional Mesh Watermarking," *IEEE Transactions on Multimedia* **10**, 1513–1527 (2008).

[2] Amat, P., Puech, W., Druon, S., and Pedeboy, J., "Lossless Data Hiding Method Based on MST and Topology Changes of 3D Triangular Mesh," in [*EUSIPCO'08: 16th European Signal Processing Conference, Lausanne, Switzerland*], (2008).

[3] Prim, R., "Shortest Connection Networks and Some Generalizations," *Bell System Technical Journal* **36**, 1389–1401 (1957).

[4] Gordeev, E., "Stability Analysis of the Minimum Spanning Tree Problem," *Computational Mathematics and Mathematical Physics* **39**(5), 738–746 (1999).

[5] Gordeev, E., "Stability Analysis in Optimization Problems on Matroids in the Metric l1," *Cybernetics and Systems Analysis* **37**, 251–259 (2001).

[6] Dixon, B., Rauch, M., and Tarjan, R., "Verification and Sensitivity Analysis of Minimum Spanning Trees in Linear Time," *SIAM Journal on Computing* **21**, 1184–1992 (1992).