IEICE TRANS. INF. & SYST., VOL.E84–D, NO.12 DECEMBER 2001

PAPER *Special Issue on Machine Vision Applications*

On the Precision of Textures*

Frank NIELSEN† ***and*** **Nicolas De MAUROY**††, ***Nonmembers***

SUMMARY In this paper, we first introduce the notion of texture precision given the 3d geometry of a scene. We then provide an algorithm to acquire a texture/color map of the scene within a given precision. The texture map is obtained using projective devices (like pinhole sensing device) from data acquired either in the real world or computer-synthesized. Finally, we describe a procedure to obtain level of precisions by combining a modified edge-collapse geometry technique with an appropriate remapping texture engine. We report on our experiments and give perspectives for further research.

key words: *texture, color map, geometric model, simplification, level of details*

1. Precision of Textures

There is a considerable amount of literature on level of details (LODs) and texture map simplification (see [2]–[6], [8] for related works). Those methods generally proceed by dividing the object into patches and then, by simplifying the geometry by minimizing the texture distortion/displacement using energetic functions on each patch. As a matter of fact, the borders of the patches are not allowed to be modified. Roughly speaking, we adopt the following scenario: given the known geometry of an object (or more generally a scene composed of objects), how many pictures (and the corresponding locations of the camera) do we need to take in order to build a texture map that has a guaranteed precision. Each picture defines a not necessarily connected super patch. Potential applications of our method include range scanning where we may constrain the positions of the camera to be on a circle centered around the object: we then scan the object in two steps: (1) get the 3d coordinates of a triangle mesh, and (2) select positions of the camera in order to build a texture map within a prescribed quality. Figure 1 points out the pixel distortions obtained from the perspective projection of a checkboard on a 3d mesh. Pixels projected onto much *slanted* triangles exhibit large distortion compared to pixels projected to *perpendicular* triangles.

In the sequel, objects $\mathcal{O}$ are considered to be tesselated manifolds with boundaries of $\mathbb{R}^3$.

Manuscript received February 23, 2001.
Manuscript revised June 21, 2001.

†The author is with Sony Computer Science Laboratories Inc., Tokyo, 141–0022 Japan.
††The author is with École Polytechnique, France.

*This paper was presented at MVA'00, pp. 31-34, ISBN 4-901122-00-2, Nov. 2000. Full version available as a technical report [1].

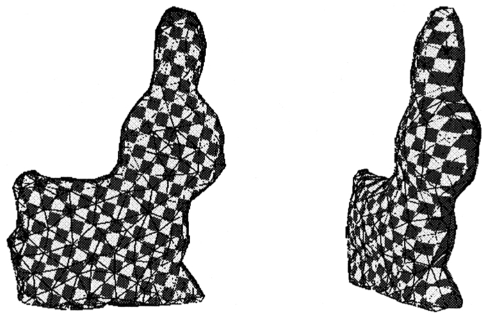


Fig. 1 Projecting a checkboard on a 3d mesh. Observation from two different viewpoints.

Definition 1: We define a texture as an application $\mathcal{T}$ which associates to each point of $\mathcal{O}$ a scalar value (or attribute). Let $\mu(R)$ defines the measure on $\mathbb{R}^3$ defined by the diameter of R. A texture $\mathcal{T}'$ has precision ϵ if we can find a *covering* $\mathcal{C} = \{c_1, \ldots, c_k\}$ of $\mathcal{O}$ (ie., $\bigcup_i c_i = \mathcal{O}$) such that:

- Each element $c_i \in \mathcal{C}$ of the cover has measure less than ϵ (i.e. $\mu(c_i) \leqq \epsilon$), and
- For each element $c_i \in \mathcal{C}$, $\mathcal{T}'(x)$ is constant for all $x \in c_i$. (We consider $\mathcal{T}'(c_i) = \frac{1}{\mu(c_i)} \int_{c_i} \mathcal{T}(s)ds$.)

For a given 3d location (x_L, y_L, z_L) and 3d orientation $(roll_L, pitch_L, yaw_L)$ of a camera $L = (x_L, y_L, z_L, roll_L, pitch_L, yaw_L)$ and a point $p \in \mathcal{O}$, the precision of p is either infinite if no pixel of the camera projects onto p, or equals to the measure of the projected pixel. We denote by $\mu_L(p)$ the precision of p from camera location and orientation L.

A triangle $\mathbf{t} \in \mathcal{O}$ has precision ϵ if $\forall p \in \mathbf{t}$, $\mu(p) \leqq \epsilon$. Let p_1, p_2 and p_3 be the vertices of $\mathbf{t}$. If $\forall i \in \{1, 2, 3\}$, $\mu(p_i) \leqq \epsilon$ then $\mu(\mathbf{t}) \leqq \epsilon$.

A simple approximation of the quality of the *sphere standard camera* (Informally speaking, each pixel is represented as a circle — see Fig. 2; see [1] for formal definition & further details) is obtained as follows: When projecting the 'pixel' circle onto the plane of the object, we can use a simpler projection than the perspective one. We can first project the pixel onto a plane containing p parallel to the focal plane and then use a parallel projection to project it on the tangent plane of the object (see Fig. 3). This projection is called "weak"

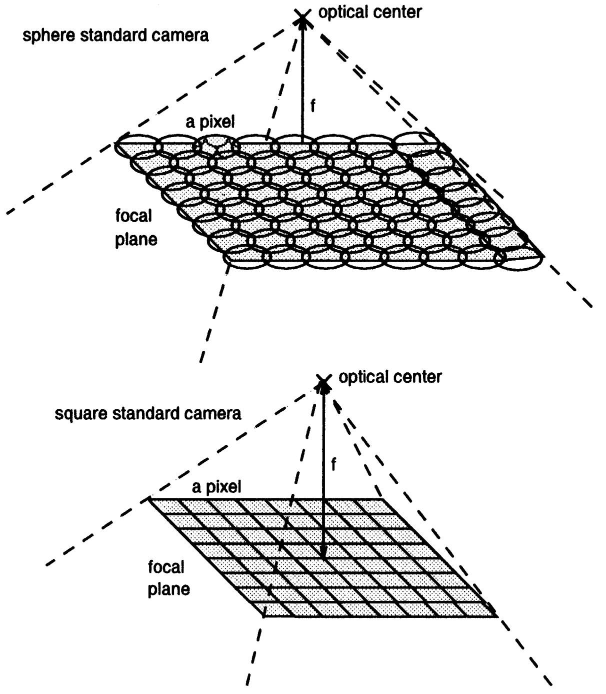


Fig. 2 The two camera models. Top: the so-called "sphere standard camera." Bottom: the so-called "square standard camera."

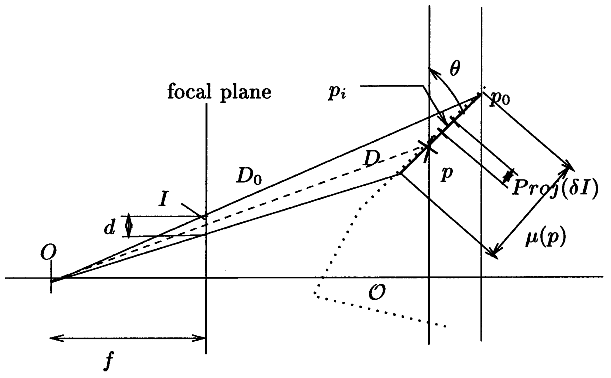


Fig. 3 Perspective projection of a pixel and "weak" approximation.

projection. In this case, the precision of a point is defined by the following simple (approximation to the 1st order) formula:

$$P(p) = \frac{d * D}{f * \cos\theta} \sim \mu_L(p),$$

with d the diameter of the pixels, D the distance between camera center O and p, and θ the angle between the focal plane and the tangent plane (depend on camera location L — parameters $roll_L$ and $pitch_L$).

This approximation has a useful property as shown below:

Theorem 1: Let c be a standard camera model and p a point of an object $\mathcal{O}$ locally plane around $\mathcal{O}$. We can find a point p_0 of the projected pixel containing p such as the formula $P(p_0) = \frac{d * D_0}{f * \cos\theta}$ is larger than the measure of the pixel $\mu_L(p)$.

Proof 1: We note that if the camera is a square standard model, we can bound any square pixel by the disc containing it. The measure of the projection of the disc is obviously an upper bound of the measure of the real projected pixel. The approximation formula is obtained by first projecting the pixel with a perspective projection on a plane parallel to the focal plane containing p, and then use a parallel projection to project it on the plane of the object. The plane projection is parallel to a normal vector of the focal plane. We use as point p_0 the point of the projected pixel that is the furthest possible from the optical center of the camera and belongs to the great axis of the projected pixel, which is an ellipsis. Let us select the diameter A as the perspective projection of a pixel I ($\mu_L(p) = |A|$ by definition). Let us call $Proj(I)$ the image by the weak projection at p of a camera pixel I (using distance D) and $Proj_0(I)$ the image by the weak projection associated with p_0 (using distance D_0).

We consider now an infinitesimal portion δI of the pixel I. It has length δd. The length $Proj(\delta I)$ can be obtained by the approximation formula which is a first order approximation. So we have $|Proj(\delta I)| = \frac{d * |Op_i|}{f * \cos\theta}$ with p_i a point of $Proj(\delta I)$. The length of $Proj_0(\delta I)$ is given immediately by $|Proj_0(\delta I)| = \frac{d * |Op_0|}{f * \cos\theta} = \frac{d * D_0}{f * \cos\theta}$, with D_0 the distance from the optical center O to the point p_0. As we have choosen p_0 as being the point of the projected pixel belonging to the great axis and the furthest possible from the optical center, we have $|Proj(\delta I)| \leqq |Proj_0(\delta I)|$. If we integrate this inequality, we have $|Proj(I)| \leqq \mu_L(p) \leqq |Proj_0(I)|$. $|Proj_0(I)|$ is an upper bound of $\mu_L(p)$.

2. Acquiring a Triangle Mesh within a Given Precision

Definition 2: Let p_i be a set of points belonging to an object $\mathcal{O}$. The set p_i is visible with a precision of ϵ if all the points of the set are seen with a precision smaller or equal to ϵ.

From this definition, it follows the definition of the precision of a triangle.

Definition 3: Let $\mathcal{O}$ be an object, and T a triangle belonging to the surface of this object. We define the precision of the triangle T viewed by the camera c as the maximum of the precision of all the points of the triangle viewed by the same camera c.

That is to say if some points have an infinite precision, the precision of the triangle is also infinite. It means a triangle partially hidden is considered to have an infinite precision; That is we consider that we do not have valuable datas about it.

2.1 Taking a Picture of a Triangle with a Precision

We need to separate the triangle into two parts, according to the following definition.

Definition 4: Let T be a triangle and c a camera. If the precision of the triangle is finite, for each point p, we can find a pixel p_i whose projection contains p. p belongs to the interior of T viewed by c if all the points of the projection of p_i belong to T. Otherwise, p belongs to the border of T viewed by c.

We want to give a way to bound the precision of the points of the interior of the triangle if we know the measure of the projection of the pixels of the vertices of the triangle on the plane of the triangle.

Definition 5: Let T be a triangle of an object $\mathcal{O}$. Let c be a standard camera (either sphere or square). We project the pixels of the camera on the plane of the triangle. We call "potential precision" of the triangle T the maximum of the approximation of the precision given by the following formula at the three vertices of the triangle. $P(p) = \frac{d*D}{f*\cos\theta}$, with d the diameter of the pixels, D the distance between camera center O and p and θ the angle between the focal plane and the plane supporting the triangle.

We now give a relationship between the precision of the points of the interior of the triangle t and its potential precision.

Theorem 2: Let T be a triangle and c a standard camera. The precision of any point of the interior of the triangle is smaller or equal than the potential precision of this triangle.

Proof 2: We denote by t_1, t_2, t_3 the three vertices of the triangle T, and we assume that t_1 is the furthest from the optical center of the camera. For any point p of the triangle T, we have the following inequalities.

$$P(p) = \frac{d*|Op|}{f*\cos\theta}$$

$$P(p) = \frac{d*|\alpha\vec{Ot_1}+\beta\vec{Ot_2}+\gamma\vec{Ot_3}|}{f*\cos\theta},$$

with $\alpha+\beta+\gamma = 1$ and α, β, γ non negative.

$$P(p) \leqq \frac{d*(\alpha|Ot_1|+\beta|Ot_2|+\gamma|Ot_3|)}{f*\cos\theta}$$

$$P(p) \leqq \frac{d*|Ot_1|}{f*\cos\theta}$$

So, the approximation of the precision of a pixel in any point of the triangle is smaller than the maximum of this approximation of the precision on the three vertices.

We have seen in the previous theorem that for every projected pixel, there is a point such that the approximation of the precision is an upper bound of the precision of this pixel.

Therefore, a pixel of the interior if the triangle has its projected pixel entirely inside the triangle, so there is a point of the triangle where the approximation of the precision is bigger than the measure of the projected pixel, and this approximation is smaller than the maximum of the approximation for the vertices. So, the measure of this pixel is smaller than the maximum of the approximation for the vertices.

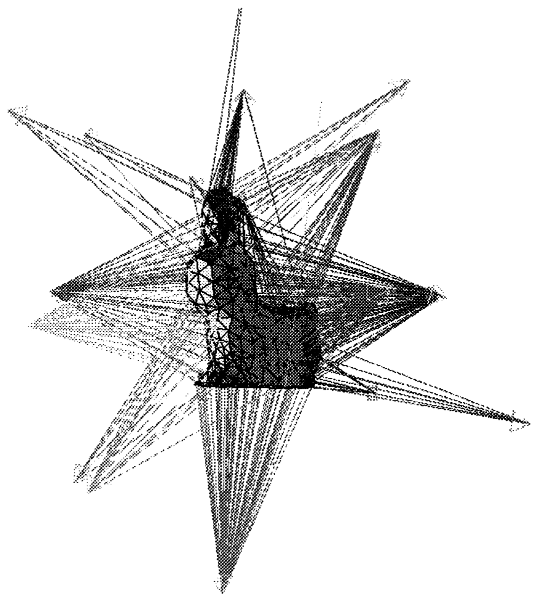


Fig. 4 A set of 18 cameras covering 98% of a 800-triangle bunny model. Each position of the camera defines a set of triangles having precision at least ϵ.

We present below a heuristic for capturing a triangle mesh within precision ϵ:

1. Find a set of camera positions (i.e. camera locations and attitudes) $\mathcal{L}$ so that each triangle can be "seen" at least once within precision ϵ. (see Fig. 4.)
2. Select a subset $\mathcal{L}' \subseteqq \mathcal{L}$ so that all triangles can be seen within precision ϵ. (This amounts to a set cover problem [9].)
3. Take the "pictures" from $\mathcal{L}'$ and create the corresponding texture map (see Fig. 5).

Step 1 is a *preselection* step. A naive approach consists of regularly sampling (grid-like) a bounding box centered around the object. Then at each point of the grid we sample the orientations of the camera. Sampling each parameter p times yield a set $|\mathcal{L}| = O(p^6)$ of camera positions that are not necessarily covering all the object. We investigated several heuristics (see [1]) that cover usually 99% of the object with linear order of camera positions. (We handle the few not-yet-covered triangles one by one.) Note that the combinatorial complexity of the visibility graph of a nonconvex polyhedron is $\Theta(n^9)$ [10].

Step 2 is a *set cover problem* in disguise [11]. In-

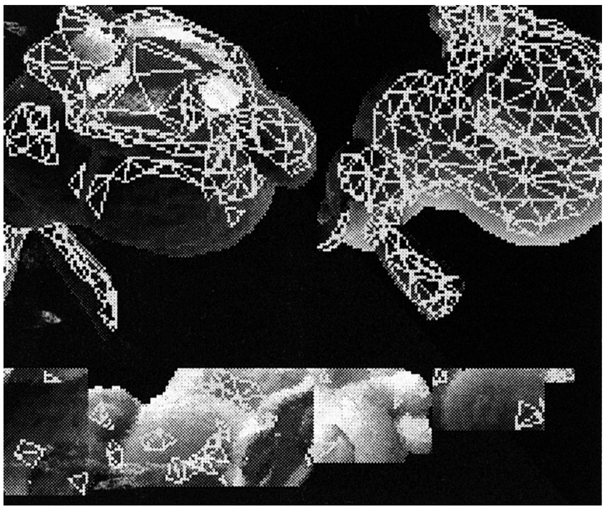


Fig. 5 Acquired pictures that combine into a texture map. Bounded triangles indicate the ones that have the desired precision.

deed, let $L(\mathcal{O}) = \mathbf{t}(L)$ be the set of triangles of $\mathcal{O}$ having precision at most ϵ. We want to minimize $|\mathcal{L}'|$ such that $\bigcup_{L\in\mathcal{L}'} \mathbf{t}(L) = \mathcal{O}$. Note that we can associate a cost (penalty) $w(L)$ to each camera position and ask for minimizing $\sum_{L\in\mathcal{L}'} w(L)$ as well. Step 1 is the most delicate part since we need to find for each triangle t a position L such that t is fully visible from L and $\mu_L(t) \leqq \epsilon$. In most cases, where we sampled the combinatorial space of camera positions, we prefer to solve a *partial set cover*, where we first ask to cover at least a fraction of the triangles (using configuration space, say $\mathcal{L}_1$). We then compute for the not-yet-covered triangles corresponding camera positions and attitudes (configuration space, say $\mathcal{L}_2$), and finally solve the *set cover problem* on $\mathcal{L}_1 \cup \mathcal{L}_2$.

Step 3 acquires the pictures either in the real world or by computer simulations, eg. raytracing. (In the latter case, we can choose orthographic projection, etc.) The portions of each picture combine in the overall texture map using 2d bin packing [12] as depicted in Fig. 5. Note that we omitted taking care of the pixels whose projections fall into several triangles (border or vertex) as depicted in Fig. 6). Those are given a special treatment as explained in [1].

The preselection step is important in practice since it samples the configuration space which otherwise will be too costly to compute. Let f denotes the focal length of a pinhole camera, and d the size of the pixels. Given a precision ϵ, we define the α-distance as $\frac{\alpha\epsilon f}{d}$.

Below, we report on several heuristics (see Fig. 7):

• *Spherical discretization:*
This heuristic selects cameras on a sphere centered at the object with radius the α-distance and camera orientations pointing to the center. The surface of the sphere is then discretized both in lattitude and longitude. Although this heuristic is very simple, we obtain good results in practice for values of α around 0.6 and a hundred points sampled on the sphere.

• *Triangle normal method:*
For each triangle of the object, we put a camera in the point c such as, if g is the centroid of the triangle, cg is a normal to the plane of the triangle, c is in the outside side of the triangle, and $|cg|$ is the α-distance. This algorithm works slightly better than the "spherical discretization", but for large objects, such as 100000 polygon objects, the size of the preselection set is prohibitive.

• *Simplified triangle normal method:*
This method is roughly speaking the same as the previous one, except that we use a simplified version of the triangle mesh. So we only have one camera for every triangle of the simplified object which can have 10 or 100 times less triangles than the original model.

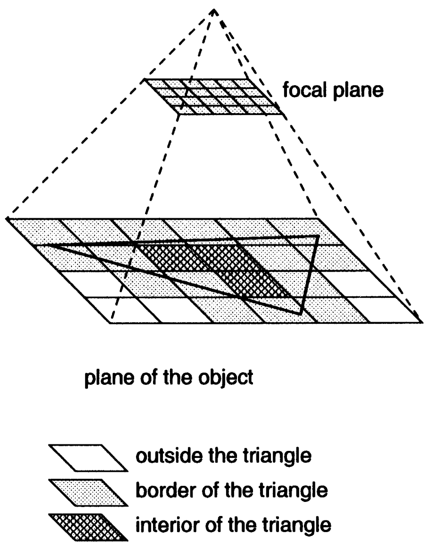


Fig. 6 Interior and border of a triangle for a given camera.

Object	Standard sphere	Triangle normal	Simplified triangle normal
A	1470 (99%)	1476 (99%)	1474 (99%)
B	1493 (97%)	1518 (99%)	1526 (100%)
C	1587 (97%)	1634 (100%)	1634 (100%)

Fig. 7 Efficiency of the preselection heuristics. **A** is a bunny model (1477 triangles), **B** is a Champagne cup (1526 triangles) and **C** is a compund scene of four animals (1634 triangles).

3. Building Levels of Precisions

We extend the notion of precision ϵ of the texture to both the *geometry* and the *texture* (in order to reduce the number of vertices and henceforth triangles). Informally speaking, we want to approximate the object so that one cannot distinguish the simplified from the original one by taking pictures having resolution at least ϵ.

Our algorithm works as follows:

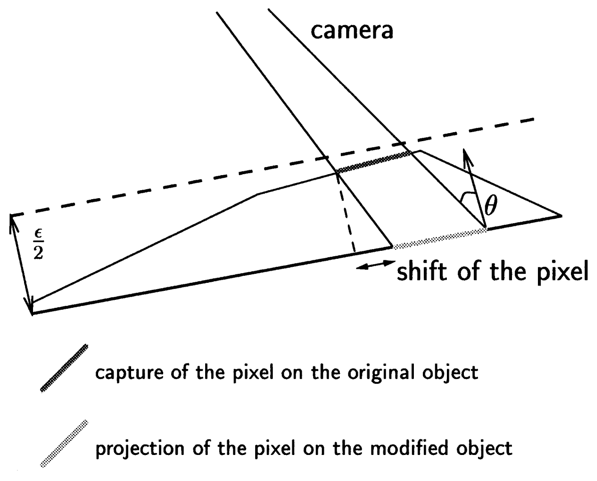

camera
θ
$\frac{\epsilon}{2}$
shift of the pixel
capture of the pixel on the original object
projection of the pixel on the modified object

Fig. 8 Mapping the photos of the original object on the simplified one

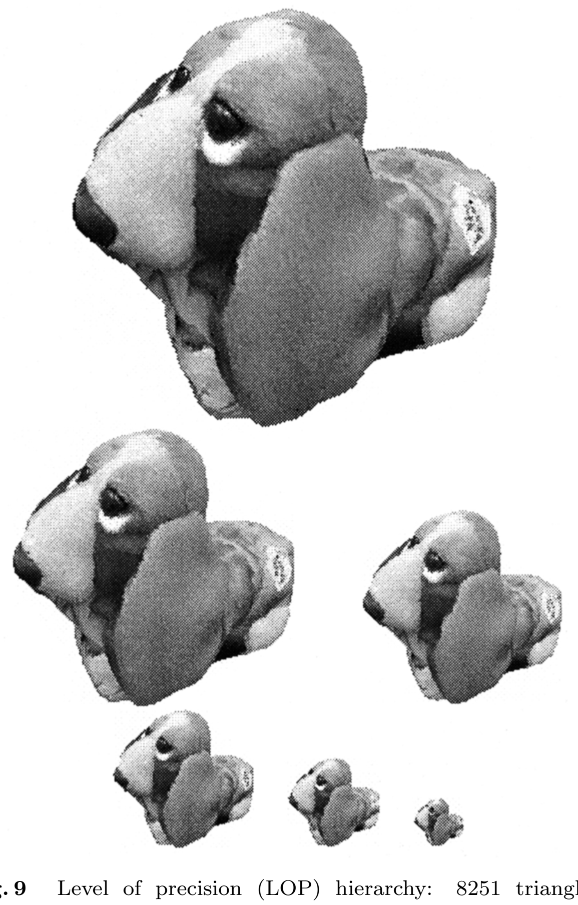


Fig. 9 Level of precision (LOP) hierarchy: 8251 triangles (200 k), 2925 triangles (120 k), 2238 triangles (71 k), 1567 triangles (33 k), 1034 triangles (18 k), 425 triangles (6 k).

1. Simplify the triangle mesh within some precision ϵ'. (We use a modified edge-collapse algorithm that do not necessarily preserve the topology as we wished.)
2. Compute, for the simplified triangle mesh, the positions L of the camera in order to acquire the whole object within some precision ϵ''.
3. Take pictures of the original object at the positions computed in Step 2 and build the overall texture map.

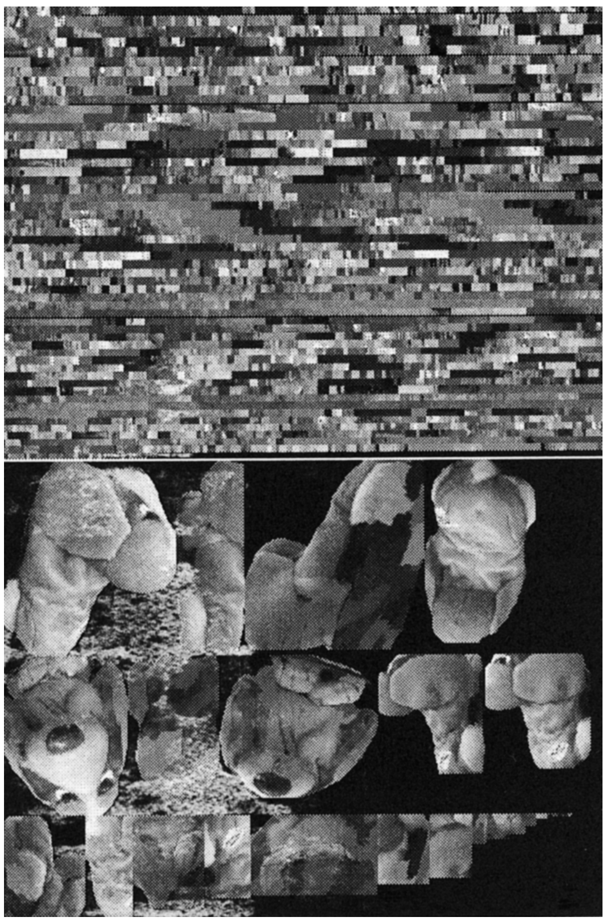


Fig. 10 Top: Texture map of a 3d textured object obtained from a commercial range scanner. Bottom: Resynthesized texture map with guaranteed precision from its 3D geometry.

Theorem 3: A simplified object satisfying the following conditions is an ϵ-approximation:

- The symmetric Haussdorf distance between the simplified and original model is at most $\epsilon' = \frac{\epsilon}{2}$.
- The texture is acquired within precision $\epsilon'' = \frac{\epsilon}{2}$.
- The angle from any ray emanating from a camera to the normal of a selected triangle is at most $\frac{\pi}{4}$ (see Fig. 8).

We add the constraints on incidence angle in Step 2 of the simplification algorithm [1]. That is, for a given camera position and location L, we say that a triangle t is in $\mathbf{t})(\mathbf{L})$ if it is fully visible and that the incidence angle θ between the focal plane induced by L and the plane supported by t is at most $\frac{\pi}{4}$. Figure 9 shows a hierarchy of level of precisions. Note that our method guarantees the precision: Especially at the silhouette, our method does not have the "polygonalization" effect (see [2]–[7] for a comparison). However, since we perform global optimization in the set cover problem, our algorithm is time consuming. Also for each level of precision, we obtain a different independent texture parametrization.

4. Concluding Remarks

Our methods extend naturally to simplification of already captured color map objects. Given a 3d textured model, we first compute its precision and then build a hierarchy of level of precisions. In that case, we can simulate orthographic projection or other appropriates sensing models. Figure 10 shows such an example. The detailed algorithms also apply for various geometric model representations, where for example the surfaces can be coded as a rought 3d mesh and a normal shift map. Our algorithms extend among others to scene illuminations and positions of cameras for telesurveillance planning.

References

[1] N. de Mauroy and F. Nielsen, "On the precision of geometric objects and some of its algorithmics," Technical Report CSL-99, Sony Computer Science Laboratories, FRL, 1999.

[2] M. Maruya, "Generating a texture map from object-surface texture data," Computer Graphics Forum, vol.14, no.3, pp.397–406, Aug. 1995. Proc. Eurographics '95. ISSN 1067-7055.

[3] M. Rioux, M. Soucy, and G. Godin, "A texture-mapping approach for the compression of colored 3D triangulations," The Visual Computer J., vol.12, no.10, pp.503–514, 1996. ISSN 0178-2789.

[4] A. Certain, J. Popovic, T. DeRose, and T. Duchamp, "Interactive multiresolution surface viewing," Computer Graphics, vol.30, pp.91–98, 1996.

[5] J. Cohen, M. Olano, and D. Manocha, "Appearance-preserving simplification," Proc. SIGGRAPH '98, Computer Graphics Proc., Annual Conference Series, pp.115–122, July 1998.

[6] P. Cignoni, C. Montani, C. Rocchini, and R. Scopigno, "A general method for recovering attribute values on simplified meshes," Proc. 9th IEEE Conf. on Visualization, pp.59–66, 1998.

[7] P.V. Sander, J. Snyder, S.J. Gortler, and H. Hoppe, "Texture mapping progressive meshes," Proc. SIGGRAPH 2001, Computer Graphics, 2001.

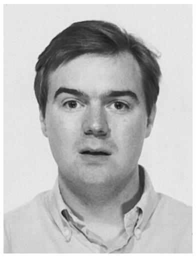
[8] H. Hoppe, "New quadric metric for simplifying meshes with appearance attributes," Proc. IEEE Conf. Visualization, ed. D. Ebert, M. Gross, and B. Hamann, pp.59–66, 1999.

[9] D.S. Hochbaum, "Efficient bounds for the stable set, vertex cover and set packing problems," Discrete Appl. Math., vol.6, pp.243–254, 1983.

[10] Z. Gigus, J. Canny, and R. Seidel, "Efficiently computing and representing aspect graphs of polyhedral objects," IEEE Trans. Pattern Anal. & Mach. Intell., vol.13, no.6, pp.542–551, June 1991.

[11] H. Brönnimann and M.T. Goodrich, "Almost optimal set covers in finite VC-dimension," Discrete Comput. Geom., vol.14, pp.263–279, 1995.

[12] J. Csirik, J.B.G. Frenk, M. Labbe, and S. Zhang, "On the multi-dimensional bin packing," Acta Cybern., 1990.



Frank Nielsen was born in France in 1971. He received the B.S. and M.S. degrees from École Normale Supérieure (ENS) of Lyon in 1992 and 1994, respectively. He defended his Ph. D. thesis on "Adaptive Computational Geometry" prepared at INRIA Sophia-Antipolis under the supervision of Pr. Boissonnat in 1996. As a civil servant of the University of Nice (France), he gave lectures at the engineering schools ESSI and ISIA (École des Mines). In 1997, he served army as a scientific member in the computer science laboratory of École Polytechnique (LIX). In 1998, he joined Sony Computer Science Laboratories, Tokyo (Japan) as an associate researcher. His current research interests include computational geometry, algorithmic vision, combinatorial optimization for geometric scenes and compression. His e-mail address is `nielsen@csl.sony.co.jp`

Nicolas de Mauroy was born in 1977. He received a B.S. from École Polytechnique (Palaiseau, France) in 1999, and is currently a M.S. student in École Nationale Supérieure des Télecommunications (Paris, France). He has been an intern in Sony CSL in 1999, and in IBM Global Services in 2000.