

Estimating affine normal vectors in discrete surfaces

Nayane Freitas Maria Andrade Dimas Martínez
Mathematics Institute
Federal University of Alagoas
Maceió, Brazil

Abstract—The invariance of geometric properties is a crucial factor in many areas of Mathematics, particularly in Computer Graphics. The affine geometry has occupied a significant place in this field of application, having an intermediate position between euclidean and projective geometries. The affine geometry is a generalization of the euclidean geometry, but it is simpler than projective geometry, both from the analytical and computational point of view. It can be used to describe many common operations in Computer Graphics. However, we did not find in literature estimators for affine geometric properties in discrete surfaces. The proposal of this work is to search for affine invariants in these surfaces, beginning with an estimate of the affine normal vector. This estimate was obtained from a discrete representation of the surface using as elements pieces of paraboloids instead of planes.

Keywords—Discrete Surface; Triangular Bézier Patch; Affine Normal Vector.

I. INTRODUCTION

The objective of this work¹ is to compute affine normal vectors in discrete surfaces, so that such estimates preserve, in the discrete model, the geometric characteristics of those vectors in the continuous model. To our knowledge, there are not known estimators for affine properties in discrete surfaces. This fact motivated this study, given the potential importance of those estimators in areas as Geometric Modeling and Computer Vision, where projective transformations are very common and can take advantage of affine properties, which are closer to projective geometry than euclidean ones.

The main contribution of this paper is the estimation of the affine normal vector in discrete surfaces. Thus, this is a starting point for the study of affine invariants in discrete surfaces such as the Gaussian and mean curvatures.

A. Related Work

It is possible to find in the literature plenty of estimators for euclidean geometric properties in discrete surfaces, from different points of view. In [13], for example, the authors estimate Gaussian and mean curvatures, in a triangular mesh, as an application of Euler’s formula, while in [6] these discrete invariants are obtained based on the Gauss-Bonnet theorem. Taubin [12] estimates some euclidean geometric properties in polygonal meshes from the discrete curvature tensor defined using an integral formula.

Estimates of affine geometric properties on smooth surfaces and discrete curves can be found in [1], [2] and [4]. In [1]

and [2] are presented estimators for the affine curvatures, and affine co-normal and normal vectors in parametric and implicit surfaces. In [4] parabolic polygons are used as models of discrete curves to estimate affine curvature and affine arc length.

In this work, we present estimates for affine normal vectors in discrete surfaces. To do this, we build a representation of these surface which is appropriated for affine geometry. That representation uses paraboloid pieces instead of triangles, allowing us to obtain the estimator of the affine normal from it. This construction is similar to the construction done in [4] for curves.

II. AFFINE CO-NORMAL AND NORMAL VECTOR IN REGULAR SURFACES.

Affine co-normal and normal vectors in regular surfaces are contravariant and covariant, respectively, by equiaffine transformation. In the next subsection, we define these transformations.

A. Affine Transformations

Proposition 1. A transformation $T : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is affine if, and only if, T satisfies $T(u) = L(u) + v_0$, where L is linear and $v_0 \in \mathbb{R}^3$.

Definition 1. An affine transformation $T : \mathbb{R}^3 \rightarrow \mathbb{R}^3$ is called equiaffine if $T(x) = L(x) + v_0$, where $\det(L) = 1$, i.e., if T preserves volume.

In this paper we consider only equiaffine transformations.

B. Affine Co-normal and Normal Vectors

Definition 2. Let S be a regular surface locally convex and $X : U \subset \mathbb{R}^2 \rightarrow S \subset \mathbb{R}^3$ a parametrization of S , the affine co-normal vector is defined by

$$\nu = |K_e|^{-1/4} N_e,$$

where K_e is the euclidean Gaussian curvature and N_e is the euclidean normal vector.

We denote by $[u, v, w]$ the determinant of a 3×3 matrix whose column vectors are u , v and w . We are now able to define the affine first fundamental form.

Definition 3. Let $X : U \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$ be a parametrization of a regular surface locally convex. The affine first fundamental

¹MSc Dissertation. Institute of Mathematics – UFAL, 2014 [9]

form, or metric of Berwald-Blaschke, is the quadratic form given by:

$$I_a = \frac{Ldu^2 + 2Mdu dv + Ndv^2}{|LN - M^2|^{1/4}},$$

where, $L = [X_u, X_v, X_{uu}]$, $M = [X_u, X_v, X_{uv}]$ and $N = [X_u, X_v, X_{vv}]$ are such that the coefficient of metric $d = LN - M^2$ is not null.

Definition 4. Let S be a regular surface and $P \in S$. The affine normal vector to S at P is defined as

$$\xi(P) = [\nu(P), \nu_u(P), \nu_v(P)]^{-1}(\nu_u(P) \times \nu_v(P)).$$

Proposition 2. In the elliptical and hyperbolic paraboloids the affine normal vector is constant.

Definition 5. Let S be a regular surface and G a group of transformations associated with a geometry. We say that a geometric measure is invariant by the group G if $\forall p \in S, \forall A \in G, m(A(p)) = m(p)$, covariant if $m(A(p)) = A(m(p))$ and contravariant if $m(A(p)) = A^{-T}(m(p))$, where $A^{-T} = (A^{-1})^T$.

Proposition 3. The affine co-normal and normal vectors are, respectively, contravariant and covariant by affine transformations.

The demonstrations of propositions 2 and 3 can be found in [9].

III. TRIANGULAR BÉZIER PATCHES

In this section we make a brief review of triangular Bézier patches, and show how to compute affine co-normal and normal vectors if the patch is quadratic. Only quadratic patches will be used later to give an alternative representation of discrete surfaces. Further details can be found in references [7], [8] and [10].

A. Representation of triangular Bézier patches

Every polynomial parametric surface b admits a unique representation of Bézier in the form

$$b(u, v, w) = \sum_{|\mathbf{i}|=n} b_{\mathbf{i}} B_{\mathbf{i}}^n(u, v, w),$$

where $u + v + w = 1$.

The coefficients $b_{\mathbf{i}}$ are called *points of Bézier* of b and are the vertices of a mesh called *Bézier mesh* or *control net* of b . Therefore, each control net defines a unique Bézier representation. The number of vertices in the control net is given by $(n+1)(n+2)/2$, where n is the degree of the polynomial (see figure 1).

Restricting the domain of this surface to $T = \{(u, v, w) \mid 0 \leq u, v, w \leq 1 \text{ and } u + v + w = 1\}$ we obtain the corresponding triangular Bézier Patch. Using the change of variables $u = x$, $v = y$ and $w = 1 - x - y$, we obtain a new expression for b in two variables:

$$\varphi(x, y) = \sum_{|\mathbf{i}|=n} b_{\mathbf{i}} B_{\mathbf{i}}^n(x, y, 1 - x - y).$$

The following properties are easy to verify

- (3.1) φ is an affine combination of Bézier points. Consequently **it is covariant by affine transformations**.
- (3.2) For all $(x, y, 1 - x - y) \in T$, $\varphi(x, y)$ is a convex combination of Bézier points (since the Bernstein polynomials are non negative on T). Therefore, the patch satisfies the convex hull property.
- (3.3) The three boundary curves are Bézier curves. As a consequence, the patch interpolates the three extreme points of its control net.

Figure 1 illustrates a quadratic triangular Bézier patch, on the left, and its control net, on the right. Note that each of the three boundary curves is a Bézier parabola defined by three control points in the corresponding side of the control net.

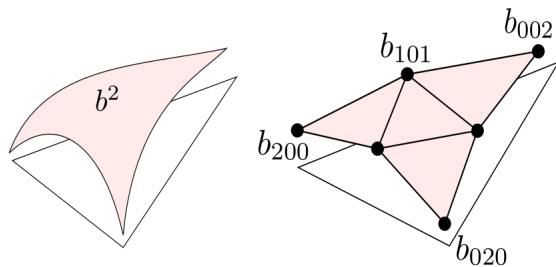


Fig. 1. Triangular Bézier patch with its control mesh.

B. De Casteljau's algorithm

The following recursive algorithm was proposed by de Casteljau [7]. It is very useful to compute the points $b(\mathbf{u}) = \sum_{|\mathbf{i}|=n} b_{\mathbf{i}} B_{\mathbf{i}}^n(\mathbf{u})$ in the triangular patch.

Algorithm 1: de Casteljau's algorithm

Input : Control net vertices $b_{\mathbf{i}} \in \mathbb{R}^3$ and the barycentric coordinates $\mathbf{u} = (u, v, w)$

Output: The point $b(\mathbf{u})$ in the patch

- 1 Consider $\mathbf{i} = (i, j, k)$, with $i, j, k \in \{0, 1, \dots, n\}$ and $|\mathbf{i}| = i + j + k = n$
- 2 Let $e_1 = (1, 0, 0)$, $e_2 = (0, 1, 0)$ e $e_3 = (0, 0, 1)$
- 3 Set $b_{\mathbf{i}}^0(\mathbf{u}) = b_{\mathbf{i}}$
- 4 For $r = 2, 3, \dots, n$, and $|\mathbf{i}| = n - r$ **do**

$$b_{\mathbf{i}}^r(\mathbf{u}) = ub_{\mathbf{i}+e_1}^{r-1}(\mathbf{u}) + vb_{\mathbf{i}+e_2}^{r-1}(\mathbf{u}) + wb_{\mathbf{i}+e_3}^{r-1}(\mathbf{u})$$

- 5 Return $b_{\mathbf{0}}^n(\mathbf{u})$
-

Figure 2 illustrates how a domain point \mathbf{u} is mapped to the corresponding image in the quadratic Bézier patch using de Casteljau's algorithm.

C. Affine normal vector in the quadratic Bézier patch

In order to compute the normal vector in a quadratic triangular Bézier patch we need to know its parameterization in two variables $\varphi : T \subset \mathbb{R}^2 \rightarrow \mathbb{R}^3$. Given the control net $\{b_{200}, b_{020}, b_{002}, b_{110}, b_{101}, b_{011}\}$, we set $\mathbf{u} = (x, y, 1 - x - y)$ and using the degree 2 Bernstein polynomials

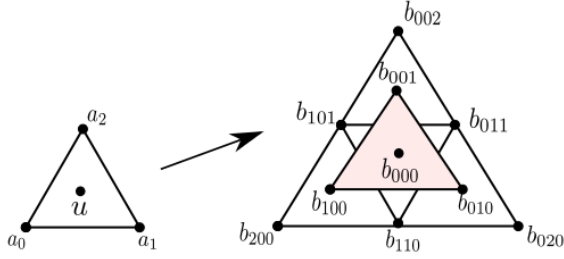


Fig. 2. De Casteljau's algorithm

$$\begin{aligned} B_{200}^2(\mathbf{u}) &= u^2 & B_{020}^2(\mathbf{u}) &= v^2 & B_{002}^2(\mathbf{u}) &= w^2 \\ B_{110}^2(\mathbf{u}) &= 2uv & B_{101}^2(\mathbf{u}) &= 2uw & B_{011}^2(\mathbf{u}) &= 2vw \end{aligned}$$

we obtain:

$$\begin{aligned} \varphi(x, y) &= (b_{200} + b_{002} - 2b_{101})x^2 + (b_{020} + b_{002} - 2b_{011})y^2 \\ &\quad + (2b_{002} - 2b_{011} - 2b_{101} + b_{110})xy \\ &\quad + 2(b_{101} - b_{002})x + 2(b_{011} - b_{002})y + b_{002}. \end{aligned}$$

We are now able to compute the affine co-normal –hence the affine normal– vector in a quadratic patch. Note that

$$K_e = \frac{LN - M^2}{\|\varphi_x \times \varphi_y\|^4} \Rightarrow |K_e|^{-1/4} = \frac{\|\varphi_x \times \varphi_y\|}{|LN - M^2|^{1/4}}.$$

So, considering $\delta = LN - M^2$, we have:

$$\nu = |K_e|^{-1/4} N_e = \frac{\varphi_x \times \varphi_y}{|\delta|^{1/4}}.$$

Define the points $a, b, c, d, e, f \in \mathbb{R}^3$ as:

$$\begin{aligned} a &= b_{200} + b_{002} - 2b_{101}; & d &= 2(b_{101} - b_{002}); \\ b &= b_{020} + b_{002} - 2b_{011}; & e &= 2(b_{011} - b_{002}); \\ c &= 2(b_{002} - b_{011} - b_{101} + b_{110}); & f &= b_{002}. \end{aligned}$$

As the quadratic triangular Bézier patch is a paraboloid [7], and the affine normal vector is constant in paraboloids, it is sufficient to compute it in the point $(0, 0)$. We have:

$$\nu(0, 0) = \frac{1}{|\delta|^{1/4}} (d_2e_3 - d_3e_2, d_3e_1 - d_1e_3, d_1e_2 - d_2e_1);$$

$$\nu_x(0, 0) = \frac{1}{|\delta|^{1/4}} (2a_2e_3 - 2a_3e_2 + d_2c_3 - d_3c_2, 2a_3e_1 - 2a_1e_3 + d_3c_1 - d_1c_3, 2a_1e_2 - 2a_2e_1 + d_1c_2 - d_2c_1);$$

$$\nu_y(0, 0) = \frac{1}{|\delta|^{1/4}} (2d_2b_3 - 2d_3b_2 + c_2e_3 - c_3e_2, 2d_3b_1 - 2d_1b_3 + c_3e_1 - c_1e_3, 2d_1b_2 - 2d_2b_1 + c_1e_2 - c_2e_1).$$

where,

$$\begin{aligned} \delta &= [\varphi_x(0, 0), \varphi_y(0, 0), \varphi_{xx}(0, 0)][\varphi_x(0, 0), \varphi_y(0, 0), \\ &\quad \varphi_{xy}(0, 0)] - [\varphi_x(0, 0), \varphi_y(0, 0), \varphi_{yy}(0, 0)]^2. \end{aligned}$$

Hence, the affine normal ξ_R is given by

$$\xi_R = [\nu(0, 0), \nu_x(0, 0), \nu_y(0, 0)]^{-1} (\nu_x(0, 0) \times \nu_y(0, 0)).$$

Note how this vector depends only of the six control points of the patch. Figure 3 shows a patch together with its normal (red) and co-normal (blue) vectors.

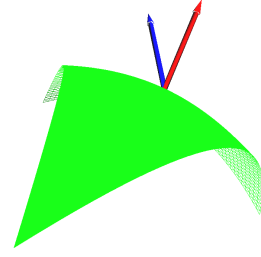


Fig. 3. Affine normal (red) and co-normal (blue) vectors in a quadratic patch

IV. DISCRETE AFFINE NORMAL VECTOR

In this section we introduce our discrete affine normal vector estimator. To this end, we first represent the discrete surface using triangular quadratic Bézier patches, instead of planar triangles.

A. Representation of discrete surfaces using triangular patches

Given a triangle mesh, we associate a patch with each of its faces. So, we can consider each triangular face T_i of the mesh as the domain of the corresponding paraboloid R_i , with $i \in \{1, 2, \dots, n\}$ and n the number of mesh faces.

To define the patch R_i we must specify its control net $\{b_{200}^i, b_{020}^i, b_{002}^i, b_{110}^i, b_{101}^i, b_{011}^i\}$. For the sake of simplicity, in the following we omit supra-index i . The three extreme points b_{200} , b_{020} and b_{002} are the vertices of triangle T_i , so the patch interpolates mesh vertices, as stated by property 3.2.3. The other three control points b_{110} , b_{101} and b_{011} , are defined with two objectives in mind: to obtain a continuous surface, and to guarantee an affine invariant construction. For each mesh triangle $T(b_{200}, b_{020}, b_{002})$ we make the following construction.

- 1) Consider the neighboring triangles $T'(b_{002}, b_{020}, b'_{200})$, $T''(b_{200}, b_{002}, b''_{020})$ and $T'''(b_{020}, b_{200}, b'''_{002})$ as shown in figure 4.

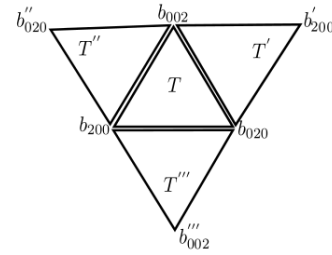


Fig. 4. Neighboring triangles of T

- 2) For each edge of T , we consider four tetrahedra, two of them defined each by one of its adjacent triangles and a point M , and two defined in the same way, after an edge flip. See figure 5 for an example. The point M is computed as the center of mass of the union of the k -neighborhoods of both edge extremes. The “radius” k of the neighborhood depends on the mesh.

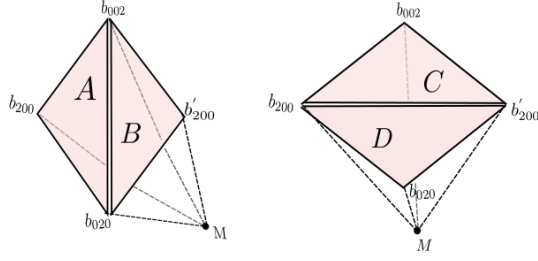


Fig. 5. Tetrahedra of one edge of T

- 3) For each edge of T , we compute a coefficient α as the ratio between the volume of the two tetrahedra before and after the flip. So, if we call the tetrahedra before the flip A and B , and after the flip C and D , we have

$$\alpha = \frac{Vol(A) + Vol(B)}{Vol(C) + Vol(D)}.$$

Note how α tells us whether the surface is convex ($\alpha > 1$) or concave ($\alpha < 1$) in that edge. Given an extra parameter $\varepsilon \in (0, 0.5]$, we define a linear interpolation parameter β as

$$\beta = \begin{cases} \alpha(1 + \varepsilon), & \text{if } \alpha > 1 \\ \alpha(1 - \varepsilon), & \text{if } \alpha < 1 \end{cases}$$

The control point corresponding to the edge is computed as the linear interpolation between M and edge midpoint, with parameter β . Parameter ε controls how far will be this control point from the edge. The computation is done by the following formulas:

$$\begin{aligned} b_{110} &= \frac{\beta'}{2}(b_{200} + b_{020}) + (1 - \beta')M'; \\ b_{011} &= \frac{\beta''}{2}(b_{020} + b_{002}) + (1 - \beta'')M''; \\ b_{101} &= \frac{\beta'''}{2}(b_{002} + b_{200}) + (1 - \beta''')M''' \end{aligned}$$

Note how the above construction is affine invariant, since it depends on means and linear interpolations, which are affine combination of points, so preserved by affine applications.

Figure 6 (a) shows a quadratic Bézier patch, while in (b) we have the whole mesh representation given by such patches.

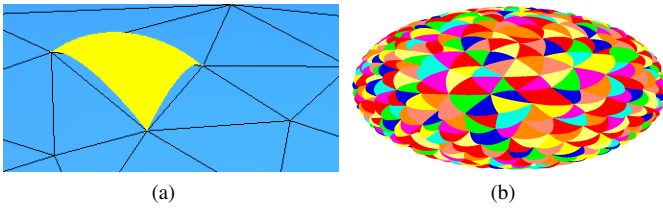


Fig. 6. Representation of an ellipsoid mesh model by quadratic patches

B. Estimating affine normal vector at mesh Vertices

Representing mesh triangles as parabolic patches has the advantage that each mesh face (a parabolic patch) has a constant affine normal vector. We are now able to estimate affine normal vector at a vertex as the mean of the normals of its incident faces, pondered by patch areas. This construction is similar to the approach usually taken in the computation of (euclidean) normals in triangular meshes. As triangle meshes have planar faces, it is impossible to estimate affine normals for them, as is done in the euclidean case, but paraboloids play the role of planes in affine geometry. The affine normal is then computed as

$$\xi_i = \frac{\sum A_{R_i} \xi_{R_i}}{\sum A_{R_i}}.$$

Areas are preserved by affine transformations, so the computed affine normal are also preserved. Patch areas are estimated as the area of a, sufficiently refined, triangular mesh approximating the patch. This mesh can be easily computed using subdivision together with de Casteljau's algorithm.

V. RESULTS AND FINAL THOUGHTS

A. Affine normal estimator:

In figure 7 we show the estimated affine normals in a sphere and an ellipsoid models. In both cases we expect the lines containing the affine normals to pass through the quadric center. In both cases we used the 7-neighborhood to compute M and $\varepsilon = 0.1$. By our experiments, the k -neighborhood size depends on mesh discretization and further study is necessary to select k automatically. Figure 8 shows the influence of the selection of k in the normal estimation, in a model with 6000 faces. Good estimates were obtained using $k = 10$.

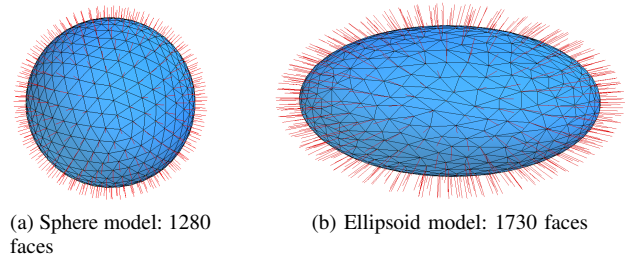


Fig. 7. Affine normals in sphere and ellipsoid

B. Preservation of geometric properties:

The affine normal and co-normal vectors are, respectively, covariant and contravariant by equiaffine transformations. Figure 11 shows some experiments in the computation of such vectors before and after randomly generated equiaffine transformations. Besides, we show the mean error in such computations. Lets call the affine normal vectors (before and after transformation) as ξ_{b_i} and ξ_{a_i} respectively, $i \in \{1, 2, \dots, n\}$. Similarly, call co-normal vector as ν_{b_i}, ν_{a_i} . The error may be computed as

$$e_\xi = \sum_{i=1}^n \frac{\|\xi_{b_i} - \xi_{a_i}\|}{n}, \quad e_\nu = \sum_{i=1}^n \frac{\|\nu_{b_i} - \nu_{a_i}\|}{n},$$

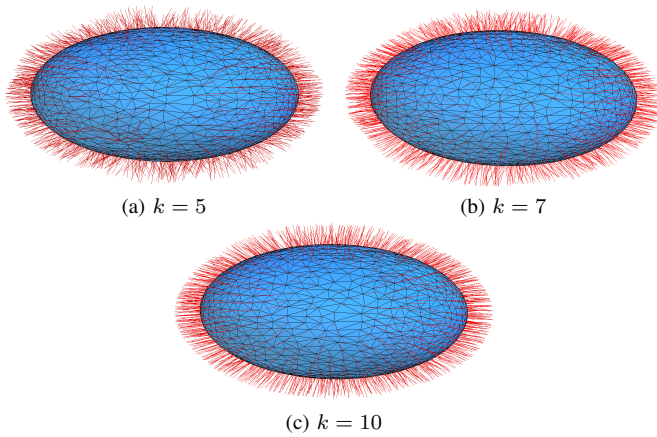


Fig. 8. Influence of k -neighborhood size in normal estimation

where e_ξ denotes error in normal computations and e_ν is the error in co-normal computations.

C. Limitations

This work is restricted to closed surfaces, with no planar or parabolic points, since affine normal is not defined on points with zero Gaussian curvature. In figure 9, for example, affine normals were not well estimated in regions where euclidean Gaussian curvature change signal. Also, we do not know yet how to choose the appropriated neighborhood size when computing M . Our experiments suggest it should depend on the mesh and this is subject of future work. Meshes containing triangles with obtuse angles have been a source of problems. In such triangles the patches may be going too far from the mesh, resulting in poor mesh approximation. Figure 10 shows an example of this behavior. We believe this is not a real issue, since it can be addressed by mesh preprocessing.

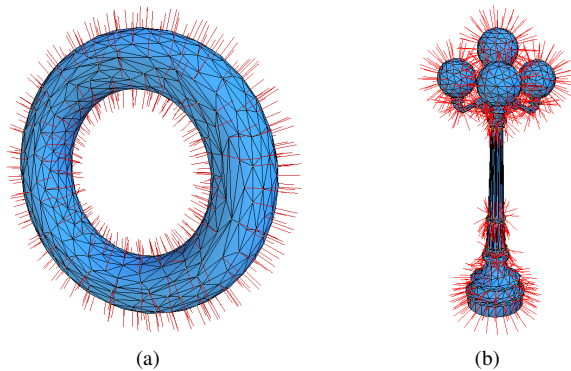


Fig. 9. Normal estimates in points with zero Gaussian curvature

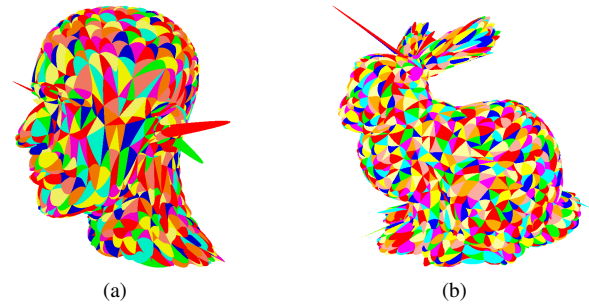


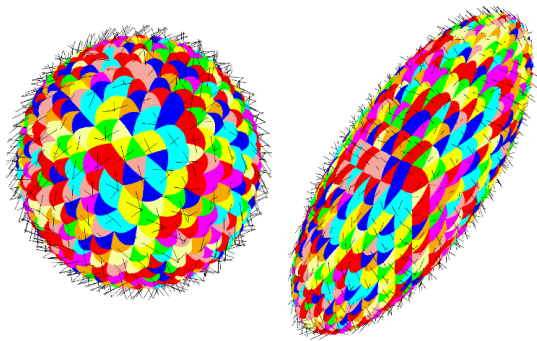
Fig. 10. Patches on obtuse-angle triangles

D. Future work

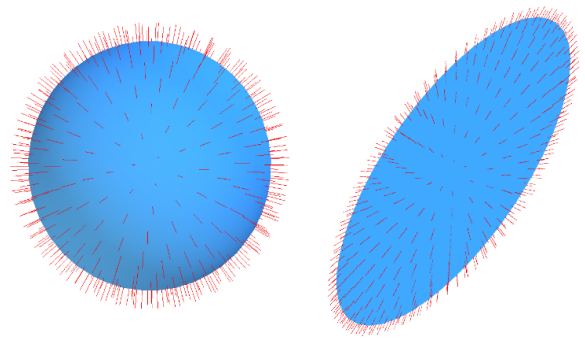
In this work we give estimates of affine normal vectors as well as affine co-normal vectors. We see it as a first step in the search for robust estimators for other affine properties as surface curvatures. Some issues are to be carefully studied as an automatic method to decide the appropriated neighborhood size in the computation of M .

REFERENCES

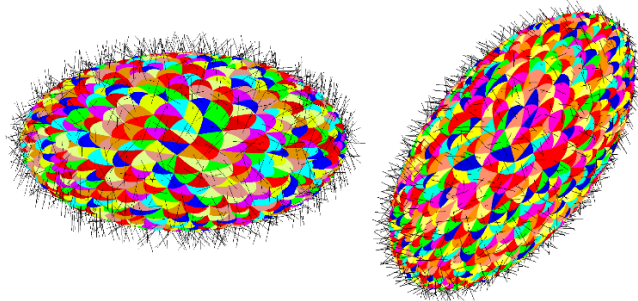
- [1] Andrade, M. Cálculo de Estruturas Afins e Aplicação às Isossuperfícies. PhD Theses. Departamento de Matemática, Pontifícia Universidade Católica do Rio de Janeiro, Rio de Janeiro. 2011.
- [2] Andrade, M. and Lewiner, T. *Cálculo e Estimação de Invariantes Geométricos: Uma Introdução às Geometrias Euclidianas e Afim*. 28 Colóquio Brasileiro de Matemática, Rio de Janeiro, 2011.
- [3] Calabi, E. Hypersurfaces with maximal affinely invariant area. American Journal of Mathematics, 104(1): 91 - 126, 1982.
- [4] Craizer, M. and Lewiner, T. and Morvan J. Parabolic polygons and discrete affine geometry. Sibgrapi 2006, pp.19-26.
- [5] Do Carmo, M. *Geometria Diferencial de Curvas e Superfícies - 4. ed.* Rio de Janeiro: SBM, 2010.
- [6] Kim, S. J. and Kim, C. H. and Levin, D., Surface simplification using a discrete curvature norm, Computers and Graphics 26 (5) (2002) 657-663.
- [7] Farin, G. *Curves and Surfaces for Computer-Aided Geometric Design: A Practical Guide*. páginas 279 - 307, 1996.
- [8] Farin, G. Triangular Bernstein - Bézier patches. Computer Aided Geometric Design. pp. 83 - 127, 1986.
- [9] Freitas, N., Estimativa do Vetor Normal Afim em Superfícies Discretas, 2014. 45 f. MsC Dissertation - Instituto de Matemática, Universidade Federal de Alagoas, Maceió. 2014.
- [10] Paluszny, M. and Prautzsch, H., and Boehm W. *Métodos de Bézier y B-Splines*, 2005.
- [11] Sichacá, B. M. Dinâmica das Linhas de Curvatura de Superfícies no Espaço Afim. 2013. MsC Dissertation - Instituto de Matemática e Estatística, Universidade Federal de Goiás, Goiânia. 2013.
- [12] Taubin G., Estimating the tensor of curvature of a surface from a polyhedral approximation, in: Proceedings of the Fifth International Conference on Computer Vision, 1995, pp. 902-907.
- [13] Watanabe, K. and Belyaev G. A., Detection of salient curvature features on polygonal surfaces, Computer Graphics Forum 20 (3) (2001) 385-392.



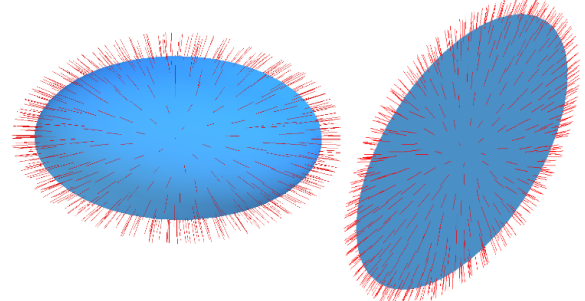
(a) $e_\nu = 1.254288e^{-15}$



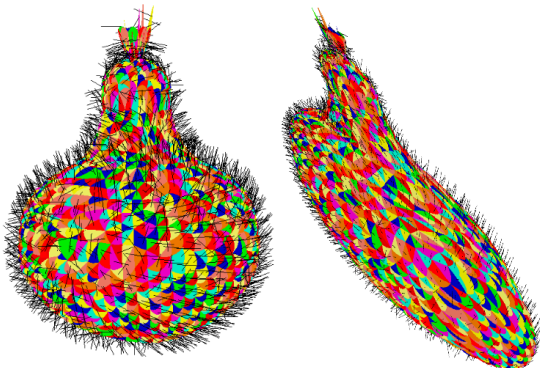
(b) $e_\xi = 8.4281e^{-15}$



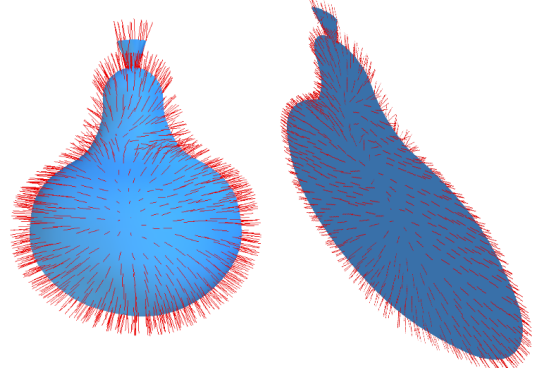
(c) $e_\nu = 8.257393e^{-15}$



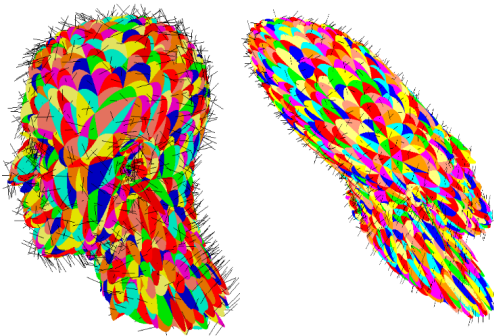
(d) $e_\xi = 6.253971e^{-14}$



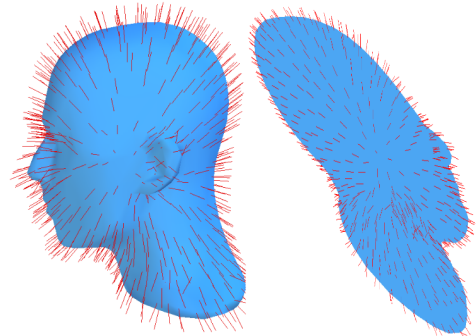
(e) $e_\nu = 4.250836e^{-12}$



(f) $e_\xi = 3.250841e^{-08}$



(g) $e_\nu = 8.35405e^{-12}$



(h) $e_\xi = 1.26713e^{-14}$

Fig. 11. Affine normal (right) and co-normal (left) vectors before and after equiaffine transformation.