Metric 3D Surface Mesh Generation Using Delaunay Criteria

Tomasz Jurczyk and Barbara Głut

AGH University of Science and Technology, Kraków, Poland
{jurczyk, glut}@agh.edu.pl

Abstract. This paper presents a technique of incorporating anisotropic metric into the Delaunay triangulation algorithm for unstructured mesh generation on 3D parametric surfaces. Both *empty circumcircle* and *inner angles* criteria of Delaunay retriangulation can be successfully used with the developed method of coordinate transformation with little adjustments. We investigate the efficiency of mesh generation process for different criteria and the quality of obtained meshes.

1 Introduction

The concept of a non-Euclidean metric is commonly used for generation of unstructured anisotropic meshes on 3D surfaces [1, 2, 3, 4]. In typical approach the *metric tensor* is used for evaluation of edge length, which is the main geometrical calculation governing the meshing process.

We consider parametric 3D surfaces which can be treated as two-dimensional domains with additional distortion introduced by the parameterization. The triangulation process is performed entirely in this parametric space basing on the Delaunay incremental insertion algorithm[1] with elements evaluation and nodes placement governed by metric. The metric description is stored in a *control space* (covering the whole domain), which can have various structure (e.g. background mesh or quadtree grid) depending on the source of the sizing data[6].

The method of mesh generation is based on Delaunay triangulation which can be obtained using criteria of *empty circumcircle* or *inner angles* [1]. In this paper we present a technique of adaptation of these criteria for generation of meshes with anisotropic metric using coordinate transformation. Utilization of both criteria is investigated with respect to the influence on triangulation efficiency and quality of obtained meshes.

2 Metric Definition

The metric is usually defined as a *metric tensor* $\mathcal{M} = \mathcal{R}\Lambda\mathcal{R}^{-1}$, where $\mathcal{R}$ is the eigenvector matrix (responsible for defining main directions) and $\Lambda = \mathrm{diag}(\lambda_i)$ is the eigenvalue matrix, defining the required length of element edges along the main directions, and $\mathcal{M}$ is a symmetric positive-definite matrix.

V.N. Alexandrov et al. (Eds.): ICCS 2006, Part II, LNCS 3992, pp. 302–309, 2006.

Another method which was used in this work, is to introduce the non-Euclidean metric through an appropriate transformation of the coordinates of points within the domain, using the *transformation matrix* $\mathbf{M}_t$:

$$P_i' = \mathbf{M}_t(P_i)P_i \ . \tag{1}$$

Similar approach was presented in [5] where authors use transformation of coordinates in a different algorithm of constrained Delaunay triangulation.

The relation between the *metric tensor* and the *transformation matrix* is given by formula

$$\mathcal{M}(P) = \mathbf{M}_t(P)\mathbf{M}_t^T(P) \ . \tag{2}$$

Equation (2) defines a family of matrices, but the symmetric one is the most desirable, being much better fitted for metric-related operations[7]. For two-dimensional parametric space such symmetric matrix $\mathbf{M}_t^{(\mathrm{sym})}$ can be calculated as:

$$\mathbf{M}_t^{(\mathrm{sym})} = \begin{bmatrix} m_{11} & m_{12} \\ m_{12} & m_{22} \end{bmatrix} = \begin{bmatrix} \frac{1}{h_1}\cos^2\alpha + \frac{1}{h_2}\sin^2\alpha & (\frac{1}{h_1} - \frac{1}{h_2})\sin\alpha\cos\alpha \\ (\frac{1}{h_1} - \frac{1}{h_2})\sin\alpha\cos\alpha & \frac{1}{h_1}\sin^2\alpha + \frac{1}{h_2}\cos^2\alpha \end{bmatrix} \ , \tag{3}$$

where α denotes the direction of element stretching, h_1 and h_2 are the required lengths of element along the base directions.

During meshing process all geometrical formulas are calculated in metric space (i.e. basing on points coordinates transformed using the transformation matrix). The optimal final mesh should have all edges of unit length in local metric space (so called *unit mesh*[3]).

In order to account for distortion introduced by an arbitrary parameterization of the given surface patch $\mathbf{p} : (u, v) \to (x, y, z)$, we use the matrix $\mathbf{G}$ of the first fundamental form:

$$\mathbf{G} = \begin{bmatrix} g_{11} & g_{12} \\ g_{12} & g_{22} \end{bmatrix} = \begin{bmatrix} \langle \mathbf{p}_u, \mathbf{p}_u \rangle & \langle \mathbf{p}_u, \mathbf{p}_v \rangle \\ \langle \mathbf{p}_u, \mathbf{p}_v \rangle & \langle \mathbf{p}_v, \mathbf{p}_v \rangle \end{bmatrix} \ . \tag{4}$$

However, instead of using it directly, the symmetric *parameterization matrix* $\mathbf{M}_P$, where $\mathbf{M}_P\mathbf{M}_P^T = \mathbf{G}$, has to be calculated. Then, the final matrix used for transformation of coordinates is calculated as a product of sizing and parameterization matrices:

$$\mathbf{M}_t = \mathbf{M}_t^{(\mathrm{sym})}\mathbf{M}_P \ . \tag{5}$$

3 Metric in Meshing Algorithms

Introducing the metric into the meshing process requires small changes in all procedures which use geometrical properties. In most cases the only requirement is to calculate the local *transformation matrix* and to translate the coordinates of vertices used in the geometrical equations. The algorithms using the mesh topology information only may remain unchanged.

The coordinate transformation technique depends on the metric to be locally constant (or close to) in the neighborhood from which the points are being taken for calculation in any given meshing step. Usually this condition of small local variation is fulfilled. However, there are situations which can pose problems, mostly at the beginning of the meshing process (when the single mesh transformations cover large areas) or when the control space has (incorrectly) large metric variations. In order to avoid errors, an additional check for metric invariance has to be performed in some algorithms.

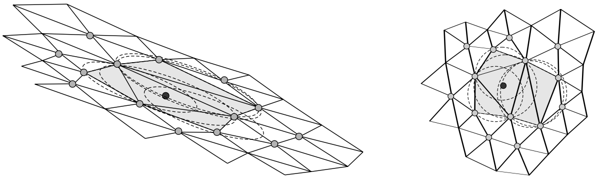
The workflow of used procedure of 3D surface mesh generation:

1. For discretization of contours of surface patches, an iterative placement of nodes for creating edges of unitary length in local metric is used. Procedure runs in parametric space of selected patch, and obtained discretization may be further refined according to control space defined in other patches incident to the given contour.
2. Triangulation of boundary vertices is created separately for each surface patch. Boundary nodes are being inserted one by one with local retriangulation of the mesh in order to maintain the Delaunay property. The retriangulation procedure adjusts the connectivity in the neighborhood of the node inserted into the mesh in order to reinstate the property of Delaunay triangulation. Both *empty circumcircle* and *inner angles* criteria[1] can be successfully adapted to produce proper results with metric introduced via the technique of coordinate transformation. Finally, the missing edges are recovered and obsolete triangles are removed.
3. The mesh is refined according to the prescribed sizing map by introducing new nodes. The selection of triangles to improve is performed basing on theirs metric quality coefficient. The new node is being inserted in the middle of the circumcircle and local retriangulation is performed.
4. Several iterations of mesh smoothing procedures are used (e.g. Laplacian smoothing).
5. If required, the triangular mesh can be further converted to quadrilateral, using one of the implemented algorithms of advancing front triangle merging[8], adapted for the developed technique of coordinate transformation.

3.1 Empty Circumcircle Criterion

The introduction of new node into the existing mesh can be accomplished using the empty circumcircle criterion, which requires locating and removing of all triangles having this node in theirs circumcircle and adding new triangles incident to the new node. Procedure starts with locating the triangle containing the new node. Adjacent triangles are iteratively checked for containing the new node in theirs circumcircles (Fig. 1). Selected elements are removed, and the vertices of the created empty cavity are being connected with the new node.

For each inserted node local metric is retrieved from control space once. The coordinates of new node and vertices of triangles being checked for empty circumcircle test have to be transformed into metric space (on average 7 triangles are checked and 16 vertices are transformed). If this procedure is used during the phase of insertion of inner nodes, the quality of all new triangles (on average



(a) insertion of new node (b) metric space

Fig. 1. Insertion of new node using empty circumcircle criterion

6) has to be recalculated which requires additional calls to control space and transformation of vertices coordinates (in total about 24 calls and 18 transformations).

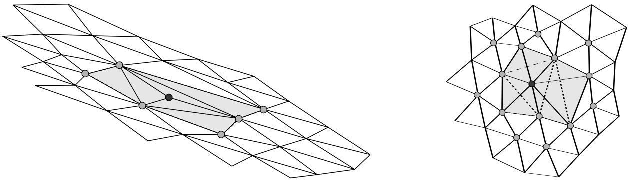
3.2 Inner Angles Criterion

Procedure starts with locating the triangle containing the new node. This triangle is then substituted by three new triangles (Fig. 2). If the node is located at some edge, two triangles have to be removed and four new elements are inserted instead. Then the edges in the adjacent elements are swapped according to inner angles criterion. The swapping procedure starts with checking the new triangles and theirs neighbors for fulfillment of the criterion, then in case of swap further pairs of triangles have to be checked iteratively.

For each inserted node local metric is retrieved from control space once. The coordinates of new node and vertices of each pair of adjacent triangles being checked for inner angles test have to be transformed into metric space (on average 13–16 pairs of triangles are checked and 2–4 edges are swapped which requires about 9–12 transformation of coordinates). If this procedure is used during the phase of insertion of inner nodes, the quality of all modified triangles (on average 6) has to be recalculated which requires additional calls to control space and transformation of vertices coordinates (in total about 24 calls and 18 transformations).

3.3 Insertion of Inner Nodes

This phase starts with reorganizing the triangles into heap-list which requires calculation of quality (proper size and shape) of all elements. Then, in each step



(a) insertion of new node (b) metric space

Fig. 2. Insertion of new node using inner angles criterion

of the main loop the worst triangle is taken from the heap and an additional node is inserted in its circumcenter. The mesh is locally retriangulated and quality (and position in the heap) of affected elements is updated.

The local metric is retrieved from the control space for barycenter of the triangle selected for refinement. This metric is used for transformation of triangle vertices for calculation of new node and for reverse-transformation of this node before insertion into mesh.

Introduction of new nodes in circumcenter of triangle allows to obtain good quality of elements, but may be unacceptable in some cases (e.g. if the new node lies too close to boundary or outside of it). Using metric approach, additional conditions have to be checked. First, the new node can't be too close to one of vertices of the containing triangle. Then it must be checked whether the insertion of the node and retriangulation according to the metric defined in this node would at all affect the improved triangle. In order to reduce number of canceled insertions, at the beginning phase of this procedure inner nodes are inserted in the middle of the longest edge of triangle.

The quality of triangle is calculated basing on the the area (in metric space) of the circle circumscribed on the given triangle, which controls both size and shape of created triangles. All vertices of the evaluated triangle have to be transformed to the metric space. The transformation matrix is calculated as intersection[7] of metric matrices retrieved from the control space for each vertex of the triangle and for its barycenter.

4 Examples

Several meshes with different size and structure are presented. Meshes **AN0** (Fig. 3(a)) and **AN1** were generated on a plane with rectangular boundary $u \in [0,2], v \in [0,1]$ and metric defined by functions $f_m^i : (u,v) \rightarrow (\alpha, h_1, h_2)$:

$$\begin{array}{l} f_m^0 = (\pi/6, 0.01, 0.11 - 0.05u) \\ f_m^1 = (\pi/6, 0.01, 0.0101 - 0.005u) \end{array} \tag{6}$$

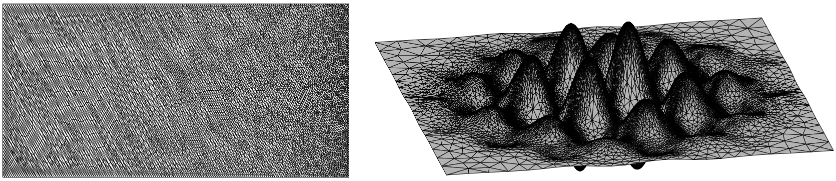
Meshes **SUR0** (Fig. 3(b)), **SUR1** and **SUR2** were created on surface patch

$$\mathbf{p} : (u,v) \rightarrow (u, v, 2.5e^{-0.1(u^2+v^2)) \sin(2u) \cos(2v)}) \tag{7}$$

where $u, v \in [-6, 6]$. The metric was automatically recognized from the curvature of the surface and boundary[6] (maximum anisotropy ratio $d_{\max} = 10$, lengths of elements are proportional to radii of curvature with factors respectively $c_0 = 0.1, c_1 = 0.025$) and was stored in the control space with structure of regular grid. Mesh **SUR2** was created as isotropic ($d_{\max} = 1$) with factor $c_2 = 0.1$.

4.1 Results

Table 1 presents metric utilization statistics for all tested meshes. As can be seen in columns $\overline{N}_c$, $\overline{N}_{pm}$ and $\overline{N}_{mp}$ the average number of control space calls



(a) planar mesh AN0

(b) surface mesh SUR0

Fig. 3. Anisotropic meshes

Table 1. Utilization of control space and metric during mesh generation using (c) – *empty circumcircle* and (a) – *inner angles* criteria. NP is number of points in the final mesh, N_c – number of calls to control space, N_{pm} – number of transformation of coordinates to metric space, N_{mp} – reverse transformations. $\overline{N}_c$, $\overline{N}_{pm}$ and $\overline{N}_{mp}$ are additionally divided by number of points.

	NP	N_c	N_{pm}	N_{mp}	$\overline{N}_c$	$\overline{N}_{pm}$	$\overline{N}_{mp}$
AN0 (c)	5069	115321	217728	4099	22.8	42.95	0.8
(a)	5139	117899	183426	4174	22.9	35.69	0.8
AN1 (c)	104447	2475058	4661639	81994	23.7	44.63	0.8
(a)	105013	2543917	3934650	82994	24.2	37.47	0.8
SUR0 (c)	23854	580469	1136499	20674	24.3	47.64	0.9
(a)	23707	599628	974884	21039	25.3	41.12	0.9
SUR1 (c)	250031	5773210	11052111	214788	23.1	44.20	0.9
(a)	249628	5874957	9324457	215831	23.5	37.35	0.9
SUR2 (c)	53404	1225174	2350930	45785	22.9	44.02	0.9
(a)	53265	1244957	1982100	45580	23.4	37.21	0.9

and metric transformations per point of the final mesh is fairly constant for all meshes.

Using *empty circumcenter* criterion tends to require lower number of control space calls, while for *inner angles* there are noticeably less transformation of coordinates. However, it must be noted that the operation of control space call for local metric retrieving is much more costly, especially for structures like background mesh.

Meshing times (2.6GHz Intel P4) for created meshes are presented in Table 2. The triangulation speed is close to linear for larger meshes and the triangulation process with *empty circumcircle* criterion is consistently faster, although the difference is not that large.

Table 3 presents quality evaluation of generated meshes (with two iterations of Laplacian smoothing in metric space). All presented coefficients (μ_{EL}, μ_T and μ_α) should be equal to 1 for ideal mesh, μ_T evaluates only size and μ_α only shape of created triangles.

Both criteria of retriangulation allow to produce meshes with comparably good quality. In case of example **SUR0** the metric variation supposedly is too large in comparison with the prescribed sizes of element, which results in creation smaller elements than expected.

Table 2. Meshing time (NT – number of triangles in final mesh, t_b – time of triangulation of boundary nodes, t_i – insertion of inner nodes, t_s – two iterations of Laplacian smoothing, τ_t – speed of triangulation (boundary and inner nodes) and τ_{ts} – speed of triangulation with smoothing)

	NT $[10^3]$	$t_b[s]$	$t_i[s]$	$t_s[s]$	$\tau_t[10^3/s]$	$\tau_{ts}[10^3/s]$
AN0 (c)	11.6	0.02	0.17	0.03	61.5	52.8
(a)	11.7	0.02	0.19	0.05	57.7	46.9
AN1 (c)	237.6	5.64	4.81	0.89	22.7	20.9
(a)	238.9	6.09	4.91	0.89	21.7	20.1
SUR0 (c)	53.5	1.49	1.81	0.39	16.2	14.5
(a)	53.6	1.49	1.91	0.39	15.8	14.2
SUR1 (c)	589.6	0.95	20.27	4.45	27.8	23.0
(a)	589.8	0.97	20.52	4.42	27.5	22.8
SUR2 (c)	126.6	1.49	3.47	0.91	25.6	21.6
(a)	126.9	1.49	3.56	0.89	25.1	21.4

Table 3. Mesh quality (μ_{EL}, σ_{EL} – average length and standard deviation of edges length in metric space, $N_{EL}[\%]$ – number of edges shorter than 0.5 or longer than 2, μ_T and σ_T – triangle size quality (ratio of optimal area to current one, in metric space), μ_α and σ_α – alpha criterion[9] of triangle shape quality in metric space)

	μ_{EL}	σ_{EL}	$N_{EL}[\%]$	μ_T	σ_T	μ_α	σ_α
AN0 (c)	1.079	0.165	0.03	0.912	0.244	0.961	0.042
(a)	1.072	0.167	0.03	0.927	0.252	0.959	0.042
AN1 (c)	1.042	0.175	0.14	0.969	0.325	0.952	0.054
(a)	1.039	0.175	0.14	0.974	0.326	0.952	0.054
SUR0 (c)	0.802	0.255	16.66	1.577	1.275	0.913	0.098
(a)	0.806	0.255	16.39	1.568	2.151	0.913	0.101
SUR1 (c)	0.989	0.195	1.69	1.045	0.503	0.953	0.054
(a)	0.990	0.195	1.68	1.046	0.659	0.952	0.055
SUR2 (c)	0.999	0.173	0.22	0.982	0.292	0.957	0.042
(a)	1.000	0.173	0.22	0.979	0.293	0.958	0.042

5 Conclusion

In this paper there were described details of the developed technique of metric introduction into meshing process via coordinate transformation. The presented examples show potential of this method to generate unstructured, possibly anisotropic meshes on 3D parametric surfaces. The running time is close to linear which allows to create meshes with very large number of elements. The quality of meshes in most cases is good with respect to both size and shape of elements, although results for the mesh **SUR0** suggest that additional processing of control space would be advisable to adjust its variation to prescribed size of elements.

Acknowledgments. The partial support of the Polish Ministry of Scientific Research and Information Technology (MNII) Grant No. 3T11F00828 is gratefully acknowledged.

References

1. George, P.L., Borouchaki, H.: Delaunay Triangulation and Meshing. Applications to Finite Elements. Hermès, Paris (1998)
2. Dolejsi, V.: Anisotropic mesh adaptation for finite volume and element methods on triangular meshes. Comput. Visual Sci. **1** (1998) 165–178
3. Frey, P.: Anisotropic metrics for mesh adaptation. In: Proc. ECCOMAS 2004, Jyvaskyla, Finland (24–28 July 2004)
4. Bottasso, C.: Anisotropic mesh adaptation by metric-driven optimization. Int. J. Numer. Meth. Engng. **60** (2004) 597–639
5. Vigo, M., Pla, N.: Computing directional constrained Delaunay triangulations. Computers & Graphics **24** (2000) 181-190
6. Głut, B., Jurczyk, T.: Definition and interpolation of discrete metric for mesh generation on 3d surfaces. Computer Science, Annual of University of Mining and Metallurgy (2005) in press.
7. Jurczyk, T., Głut, B.: Metric 3D surface mesh generation using coordinate transformation method. In: Proc. of CMS'05 Conference on Computer Methods and Systems. Volume 1., Kraków, Poland (2005) 395–405
8. Jurczyk, T., Głut, B.: Generation of good quality quadrilateral elements for anisotropic surface meshes. In: Proc. of ADMOS2003 Conference on Adaptive Modeling and Simulation, Goeteborg, Sweden (2003)
9. Lo, S., Lee, C.: Generation of gradation meshes by the background grid technique. Computers & Structures **50**(1) (1994) 21–32