Chapter 6

A triangular tangent-plane-continuous surface

6.1. Introduction

In CAD applications one is often concerned with the problem of generating smooth surfaces being unions of triangular patches, e.g., car bodies. For most purposes, at least tangent-plane continuity is required between adjacent patches. Commonly, the given data are points and associated outward normal vectors (or pairs of tangent vectors) to be interpolated in three-dimensional space. A triangulation of the data points must be known as well.

Triangular methods for the function setting, where points in the plane and associated function and derivative values are given, are discussed in [Barnhill et al.'73], [Barnhill & Farin '81], and [Farin '86]. The more general problem of interpolating arbitrary points in three-dimensional space with prescribed tangent planes has been considered later, e.g., in [Farin '83]. In [Nielson '87] and [Hamann et al.'90] a tangent-plane-continuous surfaces are constructed based on a so-called side-vertex method, originally introduced for bivariate functions (see [Nielson '79]).

Other methods are described in [Herron '85] and [Piper '87]. In [Hagen '89] tangent planes as well as principal curvatures at the data points are interpolated yielding a G^2 surface. A completely different approach is chosen in [Sederberg

'85] and [Dahmen '89], where surface patches are implicitly defined by trivariate functions.

The surface scheme developed here is based on the side-vertex technique and on degree elevated conics as the underlying curve scheme, since the given data may imply curves with and without inflection points. The conic scheme is modified in order to allow more general curves. Ball's generalized conics can be used as an alternative (see [Ball '74,'75,'77] and [Boehm '82]).

In principle, the problem is to interpolate points $\mathbf{x}_i \in I\!R^3$ with given outward (unit) normal vectors $\mathbf{n}_i$. A two-dimensional triangulation $\mathcal{T}$ defining the vertices in $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ forming triangles must be known for the data points. Patch boundary curves are first constructed along the edges of each triangle, then, a radial projector is used to blend from a vertex $\mathbf{v}_i$ to the opposite boundary curve along edge e_i, $i = 1, 2, 3$. The curves used for this blending process are degree elevated conics, being modified such that both convex and rational cubic curves with an inflection point can be generated.

Boundary curves are generated first. Then, three patch building blocks are obtained by calculating degree elevated conics, emanating from a triangle vertex and ending at a point on the opposite boundary curve. Finally, the patch building blocks are blended together in a convex combination defining the complete patch.

The (intrinsic) domain for each patch is the set of triples (u_1, u_2, u_3) of barycentric coordinates for which $\sum_{i=1}^{3} u_i = 1$, $u_i \geq 0$, $i = 1, 2, 3$. Each point on a patch is the image of a triple (u_1, u_2, u_3).

Definition 6.1. The convex combination

$$\mathbf{s}\ (\mathbf{u}) = \sum_{i=1}^{3}\ w_i(\mathbf{u})\ \mathbf{s}_i(\mathbf{u}) \tag{6.1.}$$

interpolating the triangle vertices $\mathbf{v}_1$, $\mathbf{v}_2$, and $\mathbf{v}_3$ and associated outward (unit) normal vectors $\mathbf{n}_1$, $\mathbf{n}_2$, and $\mathbf{n}_3$, where $\mathbf{u} = (u_1, u_2, u_3)$ represents the barycentric coordinates of a point in the triangle and $\sum_{i=1}^{3} \omega_i(\mathbf{u}) = 1$, $\omega_i(\mathbf{u}) \geq 0$, is called a **six parameter patch**.

Each building block $\mathbf{s}_i(\mathbf{u})$ of the patch interpolates the positional data along all three triangle edges and the normal data along the opposite edge e_i. The final patch $\mathbf{s}(\mathbf{u})$ interpolates the positional and normal data prescribed along all three edges. A particular set of weight functions ω_i is needed to solve the interpolation problem.

Theorem 6.1. *The weight functions*

$$w_i\ (\mathbf{u})\ =\ \frac{B^2_{(1,1,1)-e_i}(\mathbf{u})}{B^2_{(0,1,1)}(\mathbf{u}) + B^2_{(1,0,1)}(\mathbf{u}) + B^2_{(1,1,0)}(\mathbf{u})}, \qquad i = 1, 2, 3, \tag{6.2.}$$

where $\mathbf{e}_1 = (1, 0, 0)$, $\mathbf{e}_2 = (0, 1, 0)$, *and* $\mathbf{e}_3 = (0, 0, 1)$, *and* $B^2_{(i,j,k)}$, $i + j + k = 2$, *are the Bernstein polynomials of degree two defined as*

$$B^2_{(i,j,k)}\ (\mathbf{u})\ =\ \frac{2}{i!\ j!\ k!}\ u_1^i\ u_2^j\ u_3^k, \tag{6.3.}$$

have the properties

(i) $\quad \displaystyle\sum_{i=1}^{3} w_i(\mathbf{u})\ =\ 1,$

(ii) $\quad w_i\ (e_k)\ =\ \delta_{i,k}, \quad i, k \in \{1, 2, 3\},$ $\quad$ *and*

(iii) $D_{\mathbf{d}}\ \big(w_i(e_i)\big) \ = \ 0, \quad i \in \{1,2,3\},$

where edge e_1 *is characterized by barycentric coordinates* $(0, u_2, u_3)$, *edge* e_2 *by* $(u_1, 0, u_3)$, *and edge* e_3 *by* $(u_1, u_2, 0)$, $\delta_{i,k}$ *is the Kronecker delta, and* $D_{\mathbf{d}}$ *is a directional derivative in any direction* $\mathbf{d}$, *where* $\mathbf{d}$ *is expressed in barycentric coordinates* (d_1, d_2, d_3), $\sum_{i=1}^{3} d_i = 0$.

Proof. See [Nielson '79].

Definition 6.2. The single **patch building blocks** $\mathbf{s}_i(\mathbf{u})$, $i = 1,2,3$, are called **compatible** if each interpolates all three boundary curves of the triangular patch $\mathbf{s}(\mathbf{u})$ and the normals edge e_i, formalized

(iv) $\mathbf{s}_i[\mathbf{c}]\ (e_j) \ = \ \mathbf{c}\ (e_j), \qquad i = 1,2,3,\ j \in \{1,2,3\}$ and

(v) $\mathbf{n}\big[\mathbf{s}_i[\mathbf{c}]\big]\ (e_i) \ = \ \mathbf{n}[\mathbf{c}]\ (e_i), \quad i = 1,2,3.$

A boundary curve $\mathbf{c}$, the patch building blocks $\mathbf{s}_i$, and the normal $\mathbf{n}$ are viewed as operators. The notation "[ ]" means "restricted to." Using the properties (i), (ii), and (iii) from Theorem 6.1. and (iv) and (v) from Definition 6.2., the following interpolation theorem holds.

Theorem 6.2. *The convex combination*

$$\mathbf{s}\ (\mathbf{u}) \ = \ \sum_{i=1}^{3} \ w_i(\mathbf{u})\ \mathbf{s_i}(\mathbf{u}) \tag{6.4.}$$

interpolates all three boundary curves and the patch normals on the boundary.

Proof. a) Positional interpolation:

It is $\mathbf{s}_i[\mathbf{c}]\ (e_j) = \mathbf{c}\ (e_j)$, $i = 1,2,3$, $j \in \{1,2,3\}$, and $\sum_{i=1}^{3} w_i(e_j) = 1$. Therefore,

$\mathbf{s}[\mathbf{c}]\ (e_j) = \mathbf{c}\ (e_j)$ holds.

b) Normal interpolation:

To show interpolation of the boundary normals one calculates two non-parallel tangent vectors for a point on a boundary curve, calculates the cross product and shows that it coincides with the prescribed patch normal at that point. Let the directions in which tangent vectors are computed be $\mathbf{d}_1 = (-1, 1, 0)$, $\mathbf{d}_2 = (0, -1, 1)$, and $\mathbf{d}_3 = (1, 0, -1)$. $\mathbf{D}_{\mathbf{d}_i}$ is the vector valued derivative operator determining tangent vectors in direction $\mathbf{d}_i$. Using the product rule and taking the properties (ii) and (iv) into account (Theorem 6.1., Definition 6.2.), one obtains

$$\mathbf{D}_{\mathbf{d}_i}\mathbf{s}[\mathbf{c}]\ (e_i)$$

$$= \ w_1(e_i)\ \mathbf{D}_{\mathbf{d}_i}\mathbf{s}_1[\mathbf{c}]\ (e_i) + D_{\mathbf{d}_i}w_1(e_i)\ \mathbf{s}_1[\mathbf{c}]\ (e_i)$$

$$+w_2(e_i)\ \mathbf{D}_{\mathbf{d}_i}\mathbf{s}_2[\mathbf{c}]\ (e_i) + D_{\mathbf{d}_i}w_2(e_i)\ \mathbf{s}_2[\mathbf{c}]\ (e_i)$$

$$+w_3(e_i)\ \mathbf{D}_{\mathbf{d}_i}\mathbf{s}_3[\mathbf{c}]\ (e_i) + D_{\mathbf{d}_i}w_3(e_i)\ \mathbf{s}_3[\mathbf{c}]\ (e_i)$$

$$= \ \mathbf{D}_{\mathbf{d}_i}\mathbf{s}_i[\mathbf{c}]\ (e_i) + \mathbf{c}\ (e_i)\ \Big(D_{\mathbf{d}_i}\big(w_1(e_i) + w_2(e_i) + w_3(e_i)\big)\Big)$$

$$= \ \mathbf{D}_{\mathbf{d}_i}\mathbf{s}_i[\mathbf{c}]\ (e_i).$$

Considering the result from a), the tangent vector along a boundary curve is given by

$$\mathbf{D}_{\mathbf{d}_{1+(i \bmod 3)}}\mathbf{s}_i[\mathbf{c}]\ (e_i) \ = \ \mathbf{D}_{\mathbf{d}_{1+(i \bmod 3)}}\mathbf{s}[\mathbf{c}]\ (e_i),$$

$i = 1, 2, 3$. Choosing two arbitrary directions $\mathbf{d}_i$ and $\mathbf{d}_{1+(i \bmod 3)}$, $i = 1, 2, 3$, the patch normal along a boundary is determined by the cross product

$$\mathbf{n}[\mathbf{c}]\ (e_i) \;=\; \mathbf{n}\big[\mathbf{s}_i[\mathbf{c}]\big]\ (e_i)$$

$$= \mathbf{D}_{\mathbf{d}_i}\mathbf{s}_i[\mathbf{c}]\ (e_i) \times \mathbf{D}_{\mathbf{d}_{1+(i \bmod 3)}}\mathbf{s}_i[\mathbf{c}]\ (e_i) \;=\; \mathbf{D}_{\mathbf{d}_i}\mathbf{s}[\mathbf{c}]\ (e_i) \times \mathbf{D}_{\mathbf{d}_{1+(i \bmod 3)}}\mathbf{s}[\mathbf{c}]\ (e_i)$$

$$= \mathbf{n}\big[\mathbf{s}[\mathbf{c}]\big]\ (e_i),$$

proving normal interpolation along the boundaries.

q.e.d.

The concept of barycentric coordinates for a triangle is shown in Figure 6.1.

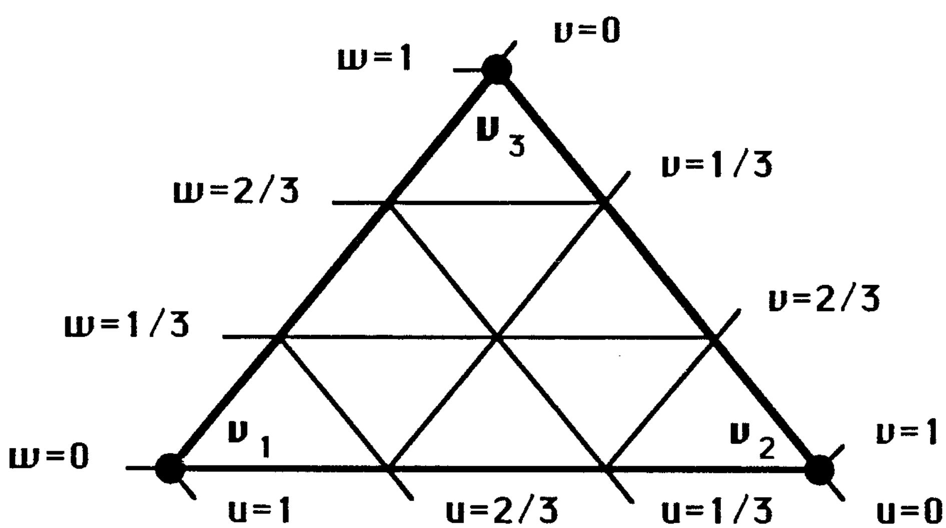

Fig. 6.1. Concept of barycentric coordinates for a triangle.

6.2. The conic curve scheme

A planar curve scheme is needed for the interpolation of two points $\mathbf{b}_0 \in I\!R^3$ and $\mathbf{b}_3 \in I\!R^3$ and two associated outward unit normal vectors, $\mathbf{n}_0 \in I\!R^3$ and $\mathbf{n}_3 \in I\!R^3$. A plane containing unit tangent vectors for the end points of the curve must be defined.

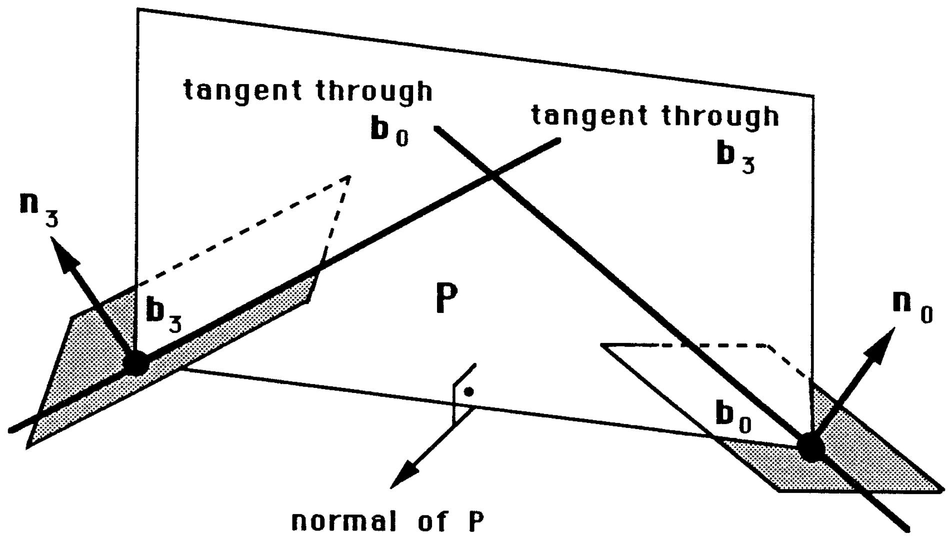

Fig. 6.2. Conic in Bézier representation.

Referring to Figures 6.2. and 6.3., the construction proceeds as follows:

(i) Define a plane P through $\mathbf{b}_0$ and $\mathbf{b}_3$ containing the desired curve. This plane is specified by the requirement that the vector $\frac{1}{2}(\mathbf{n}_0 + \mathbf{n}_3)$ lies in it (special care necessary for the case $\mathbf{n}_0 = -\mathbf{n}_3$).

(ii) Construct the intersection of the conic plane P and the tangent plane P_0 at $\mathbf{b}_0$ and of P and the tangent plane P_3 at $\mathbf{b}_3$. Each of the straight lines obtained defines the tangent of the desired curve at $\mathbf{b}_0$ and $\mathbf{b}_3$, respectively. The cross product between the normal of P and the two given normal vectors at the end points of the curve define unit tangent vectors for the end points, denoted by $\mathbf{t}_0$ and $\mathbf{t}_3$.

(iii) The resulting conic is written as a degree elevated rational Bézier curve of degree three,

$$\mathbf{c}\ (t) \ = \ \frac{\sum_{i=0}^{3} \omega_i\ \mathbf{b}_i\ B_i^3(t)}{\sum_{i=0}^{3} \omega_i\ B_i^3(t)}, \tag{6.5.}$$

where $t \in [0,1]$, $\omega_0 = \omega_3 = 1$, $\omega_1 = \omega_2 = \omega$, and

$$B_i^3\ (t) \ = \ \binom{3}{i}\ (1-t)^{3-i}\ t^i, \qquad i = 0...3. \tag{6.6.}$$

Referring to Figure 6.3., the interior Bézier points, $\mathbf{b}_1$ and $\mathbf{b}_2$, lie on a line parallel to the line through $\mathbf{b}_0$ and $\mathbf{b}_3$. Using the law of sines one obtains

$$l_{0,1} \ = \ \frac{\sin\beta}{\sin\gamma}\ L_{0,3}. \tag{6.7.}$$

Degree elevation requires the following ratio to hold (see [Farin '90]):

$$\frac{L_{0,1}}{l_{0,1} - L_{0,1}} \ = \ 2\ \omega. \tag{6.8.}$$

Therefore,

$$L_{0,1} \ = \ \frac{2\ \omega}{1 + 2\ \omega}\ \frac{\sin\beta}{\sin\gamma}\ L_{0,3}. \tag{6.9.}$$

Thus, one gets

$$\mathbf{b}_1 \ = \ \mathbf{b}_0 + L_{0,1}\mathbf{t}_0. \tag{6.10.}$$

The same construction is carried out for $\mathbf{b}_2$.

Theorem 6.3. *Choosing the weight* $\omega = \omega_1 = \omega_2$ *as*

$$\omega \ = \ \sin\frac{\gamma}{2} \ = \ \cos\alpha \tag{6.11.}$$

determines a finite value $L_{0,1}$ for γ approaching zero (parallel tangents at end points) and defines a circular arc for the case $\alpha = \beta$.

Proof. Obviously, the scheme yields circular arcs for an isosceles triangle as Bézier polygon (see [Boehm et al.'84]).

The choice for ω also guarantees a finite value for $L_{0,1}$, since

$$\lim_{\gamma \to 0} L_{0,1} = \lim_{\gamma \to 0} \frac{2 \sin \frac{\gamma}{2}}{1 + 2 \sin \frac{\gamma}{2}} \frac{\sin \beta}{\sin \gamma} L_{0,3}$$

$$= \lim_{\gamma \to 0} \frac{2 \frac{\gamma}{2}}{1 + 2 \frac{\gamma}{2}} \frac{\sin \beta}{\gamma} L_{0,3} = L_{0,3} \sin \beta. \qquad (6.12.)$$

q.e.d.

In order to avoid consistency problems between patches and to reduce input information all weights associated with interior Bézier points should be chosen automatically as proposed in Theorem 6.3. Of course, they can also be specified by the user, considering the consistency constraints. Figure 6.3. illustrates the degree elevation process for conics.

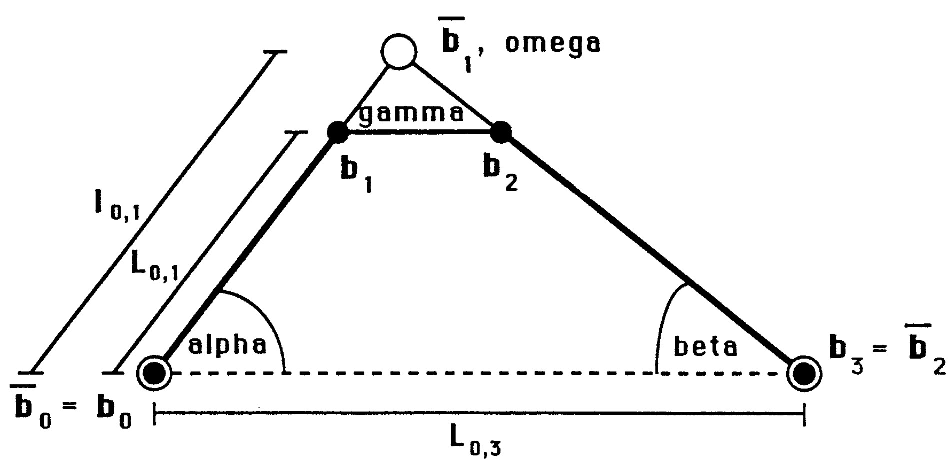

Fig. 6.3. Degree elevation for a conic in Bézier representation.

6.3. Computing the patch building blocks

The computation of a point $\mathbf{s}_i(\mathbf{u})$, $i = 1, 2, 3$, for given parameter values u_1, u_2, and u_3 is based on generating two separate curves. The first curve is associated with the edge e_i of the domain triangle interpolating the vertices associated with this edge and the computed tangent vectors at those vertices.

This curve is evaluated to obtain a point on the boundary along edge e_i. The second curve constructed is the result of blending from the vertex $\mathbf{v}_i$ to the point on the curve along e_i, thus interpolating the vertex $\mathbf{v}_i$, the point on the curve along e_i, and the two tangent vectors prescribed for these two points. This curve finally determines a point on this building block.

The planar conic scheme is used for the computation of points on a building block $\mathbf{s}_i(\mathbf{u})$ in the following way:

(i) *Curve scheme for the boundary curves*

Using the planar curve scheme based on degree elevated conics it is easy to generate the three boundary curves $\mathbf{c}_1(t)$, $\mathbf{c}_2(t)$, and $\mathbf{c}_3(t)$. For the computation of patch building block $\mathbf{s}_1(\mathbf{u})$ at $\mathbf{u} = (u_1, u_2, u_3)$ one evaluates the curve $\mathbf{c}_1(t)$ for $t = u_3/(1 - u_1) \in [0, 1]$ along edge e_1. The input for the conic scheme are the vertices $\mathbf{v}_2$ and $\mathbf{v}_3$, the normals $\mathbf{n}_2$ and $\mathbf{n}_3$, and the parameter t. This results in a particular point on the patch boundary.

(ii) *Surface scheme for a point on* $\mathbf{s}_i(\mathbf{u})$

Having computed a point $\mathbf{c}_i(t)$, $i = 1, 2, 3$, on a boundary curve, the next step is the estimation of the surface normal of the final patch at a particular point on $\mathbf{c}_i(t)$. The surface normal $\mathbf{n}_i^S(t)$ along $\mathbf{c}_i(t)$ must be perpendicular to this curve itself,

$$\mathbf{n}_i^S(t) \;\cdot\; \dot{\mathbf{c}}_i(t) \;=\; 0, \tag{6.13.}$$

$i = 1, 2, 3$, where $\dot{\mathbf{c}}_i(t)$ denotes the tangent vector of the conic. Requiring

$$\mathbf{n}_i^S(t) \;\cdot\; \mathbf{n}_i^C(t) \;=\; \gamma\;(t), \tag{6.14.}$$

where $\mathbf{n}_i^C(t)$ denotes the unit normal vector to the conic in its plane, $\mathbf{n}_i^S(t)$ is determined. The value $\gamma\;(t)$ is chosen according to the following interpretation: At $t = 0$,

$$\mathbf{n}_i^S(0) \;\cdot\; \mathbf{n}_i^C(0) \;=\; \gamma\;(0) \tag{6.15.}$$

denotes the cosine of the angle formed by the surface normal $\mathbf{n}_i^S(0)$ and the conic normal $\mathbf{n}_i^C(0)$. At $t = 1$, $\gamma\;(1)$ has the analogous interpretation. If one sets

$$\gamma\;(t) \;=\; (1-t)\;\gamma(0) + t\;\gamma(1), \tag{6.16.}$$

the cosine of the angle formed by $\mathbf{n}_i^S(t)$ and $\mathbf{n}_i^C(t)$ varies linearly along the edge.

This process guarantees that the surface normals are the same as the given ones at the vertices and, moreover, that the surface normals along an edge are the same for this triangular surface patch as well as for a neighbor patch sharing the edge. Thus, tangent-plane continuity between adjacent patches is assured.

Remark 6.1. The choice of the boundary surface normal considers the boundary curve itself, a simple linear interpolation of the given normal vectors at the two vertices is avoided.

The process of determining the surface normal along the boundary conic $\mathbf{c}_1(t)$ is illustrated in Figure 6.4.

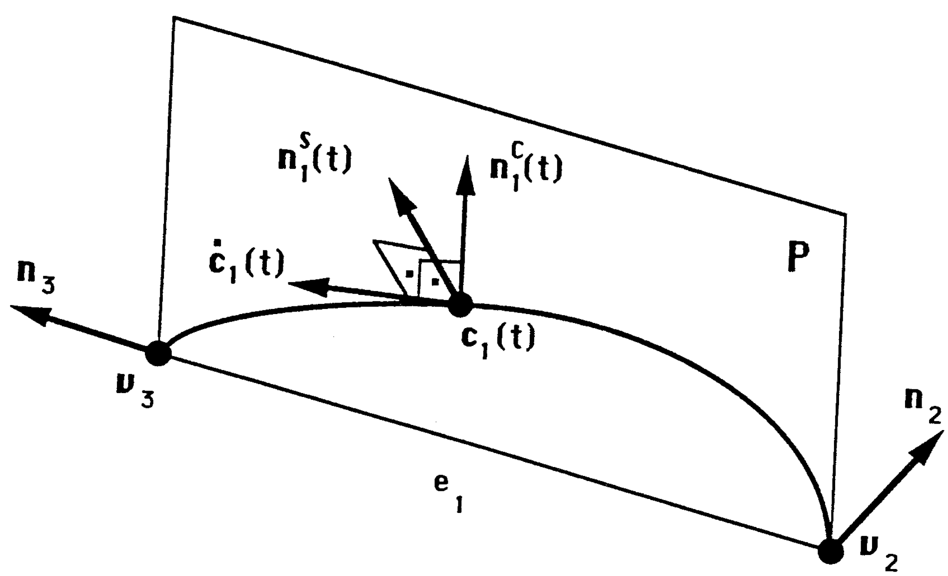

Fig. 6.4. Generating patch normal along edge e_1.

Using the idea of radial projectors, the blending from a vertex to the opposite boundary curve is done next. This is described for the vertex $\mathbf{v}_1$ and its associated boundary curve $\mathbf{c}_1(t)$. The generation of a point on a curve, emanating from $\mathbf{v}_1$ and ending at a point on $\mathbf{c}_1(t)$, follows the same principle as the generation of a point on the boundary conic $\mathbf{c}_1(t)$, $t = u_3/(1-u_1) \in [0,1]$.

To obtain a point on the patch building block $\mathbf{s}_1(\mathbf{u})$ at $\mathbf{u} = (u_1, u_2, u_3)$ one has to construct a curve $\mathbf{c}(\bar{t})$, $\bar{t} = (1 - u_1) \in [0, 1]$. The input data for the curve scheme are the vertices $\mathbf{v}_1$ and $\mathbf{c}_1(t)$, the normals $\mathbf{n}_1$ and the constructed surface normal $\mathbf{n}_1^S(t)$ along edge e_1. The computation of a point on $\mathbf{s}_1(\mathbf{u})$ is shown in Figure 6.5.

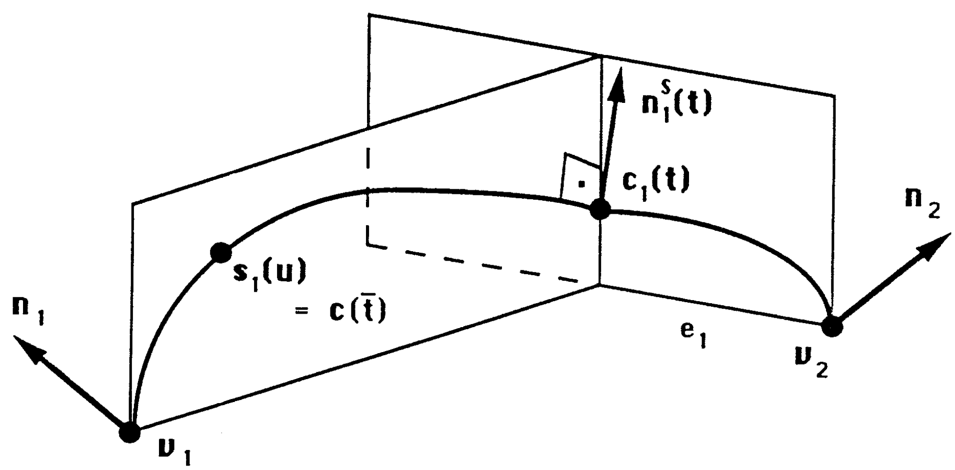

Fig. 6.5. Evaluating first patch building block.

Repeating this process for the other two building blocks $\mathbf{s}_2(\mathbf{u})$ and $\mathbf{s}_3(\mathbf{u})$ finally yields the point $\mathbf{s}(\mathbf{u})$ on the surface. The weights for the interior Bézier points of all conics can be interpreted as tension parameters, allowing the generation of patches approaching the triangle $\{\mathbf{v}_1, \mathbf{v}_2, \mathbf{v}_3\}$ for small weights ($\omega \ll 1$).

So far, only convex data configurations have been considered, making it possible to use conics. Generalized conics were introduced in [Ball '74,'75,'77] as rational curves of degree three, including conics as a subset. The concept of generalized conics allows to model input data implying curves with and without an inflection point. A criterion must be given that allows to decide, whether the presented conic scheme can be used for a planar data configuration.

Definition 6.3. The line through the vertices $\mathbf{v}_1$ and $\mathbf{v}_2$ divides the plane into two half-planes. The tangent vectors $\mathbf{t}_1$ and $\mathbf{t}_2$ associated with $\mathbf{v}_1$ and $\mathbf{v}_2$ define a **convex configuration** if the tangent vectors are directed into opposite half-planes and a **non-convex configuration**, otherwise.

If given data are convex the planar scheme for degree elevated conics can be used as described above. In the case of a non-convex configuration, the two interior Bézier points for the curve scheme must be constructed in a way that a rational curve of degree three with an inflection point is obtained.

Assuming the tangent vectors $\mathbf{t}_1$ and $\mathbf{t}_2$ are directed into the same half-plane, the prescribed tangents through $\mathbf{v}_1$ and $\mathbf{v}_2$ are reflected with respect to the axis given by the line through $\mathbf{v}_1$ and $\mathbf{v}_2$. Therefore, two pairs of tangents are obtained, one pair per vertex. The degree elevation procedure is then carried out in both half-planes. The interior Bézier points must be chosen in a way such that $(\mathbf{b}_1 - \mathbf{b}_0) = \alpha_1 \mathbf{t}_1,\ \alpha_1 > 0$, and $(\mathbf{b}_3 - \mathbf{b}_2) = \alpha_2 \mathbf{t}_2,\ \alpha_2 > 0$.

The four different data configurations possible are illustrated in Figure 6.6. Two curves with and two without inflection points can be obtained.

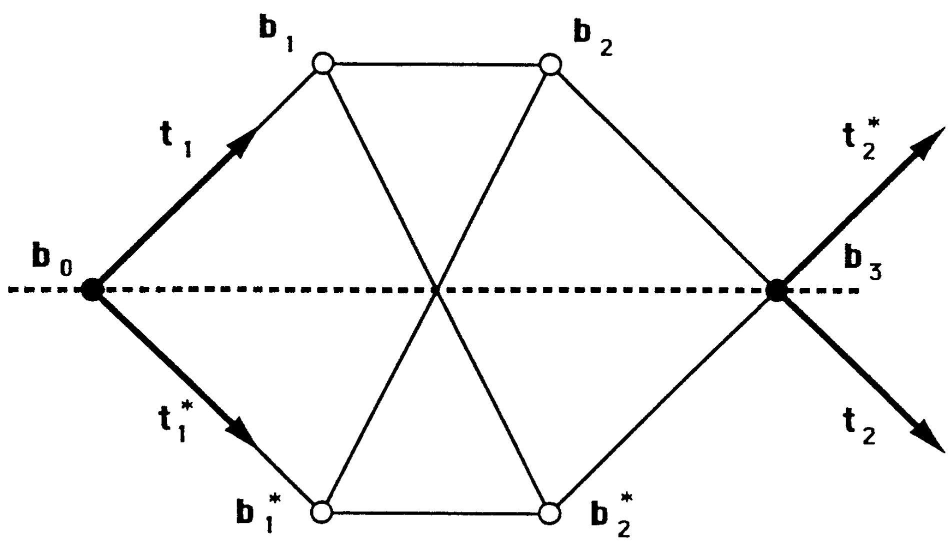

Fig. 6.6. Convex and non-convex data configurations defined by end points and end tangents in a plane.

Thus, the concept of degree elevated conics allows us to handle both convex and non-convex data. The continuity inside a single patch building block is guaranteed, since the construction of the interior Bézier points is continuous with respect to the involved angles.

Remark 6.2. If all data points $\mathbf{x}_j$, $j \neq i$, are lying in the same half-space determined by the plane through $\mathbf{x}_i$ with normal $\mathbf{n}_i$, and this is true for all i, a convex surface is implied and "usual" conics can be used everywhere as curve scheme.

Remark 6.3. Choosing the weights automatically, as described in Theorem 6.3., yields circular arcs when the data configuration implies this. Choosing points and normals from a unit sphere as input data produces a surface rather well approximating a sphere. However, the presented scheme does not have spherical precision. This is the result of the convex combination. Each single building block has spherical precision, but each one generates a different point on the sphere.

Remark 6.4. Using degree elevated conics instead of parametric cubic curves guarantees that one does not obtain inflection points or loops unless the prescribed normals at the two end points of a curve imply an inflection point.

In Figure 6.7., three different surfaces are shown using increasing weights. The four vertices of an equilateral tetrahedron inscribed in a unit sphere with the associated normals of the sphere at these points are given as input. For the first surface the weight ω is chosen automatically (Theorem 6.3.). In this case, the maximal distance of the resulting surface from the unit sphere is about 0.01. The other two surfaces both have lower weights.

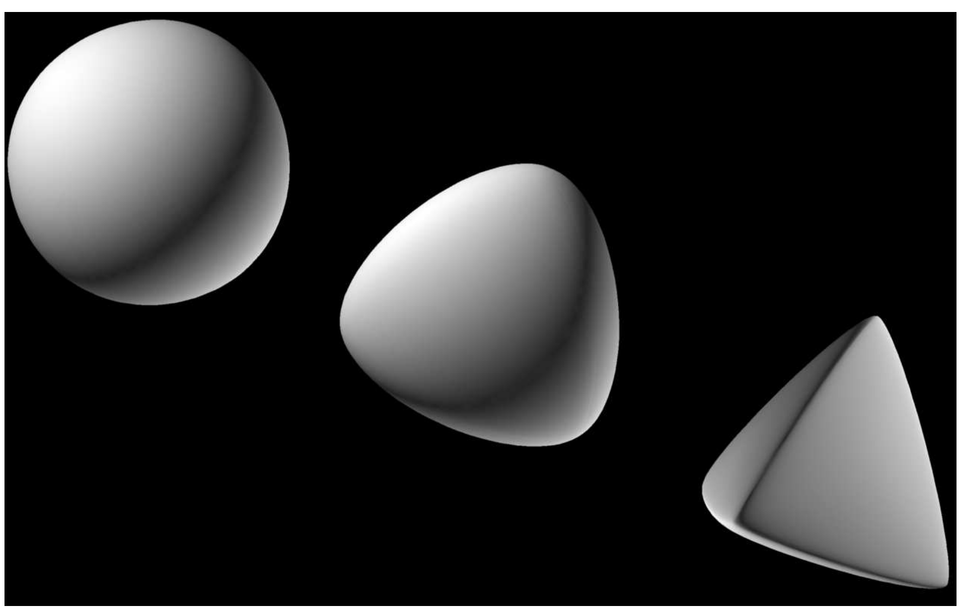

Fig. 6.7. Triangular surfaces obtained from spherical data using increasing weights (from left to right).

In Figure 6.8., the surface scheme has been applied to the reduced triangulation approximating a human skull (about 6,000 triangles, see Figure 5.11.). All weights ω are chosen automatically (Theorem 6.3.).

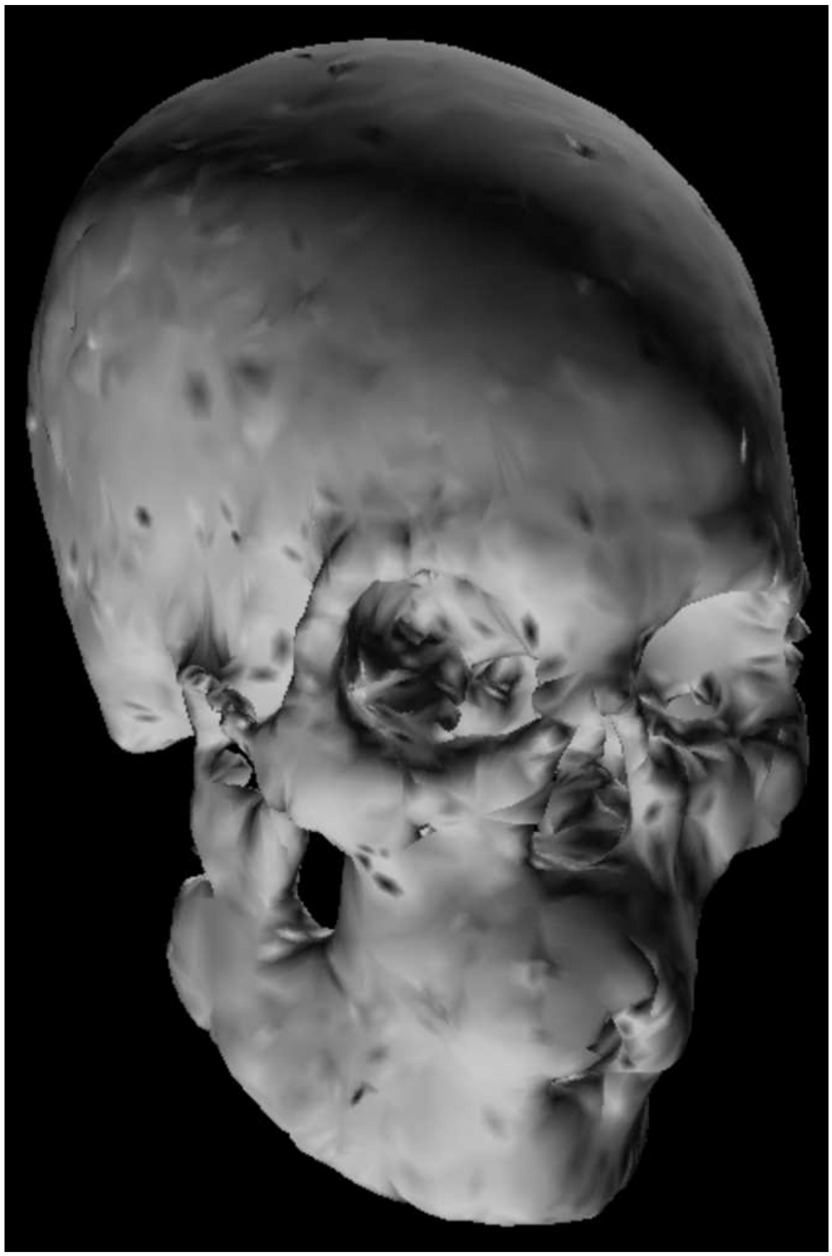

Fig. 6.8. Triangular surface for reduced skull triangulation, weights chosen automatically.