RECOGNITION AND RECONSTRUCTION OF SPECIAL SURFACES FROM POINT CLOUDS

Helmut Pottmann [a], **Stefan Leopoldseder** [a], **Johannes Wallner** [a], **Martin Peternell** [b]

[a] Institute of Geometry, Vienna University of Technology, Wiedner Hauptstr. 8–10, A–1040 Wien, Austria - (pottmann, leopoldseder, wallner)@geometrie.tuwien.ac.at

[b] Advanced Computer Vision GmbH, Wohllebengasse 6/5, A–1040 Wien, Austria - martin.peternell@arcs.ac.at

KEY WORDS: CAD, geometry, engineering, surface recognition, surface reconstruction, three-dimensional modeling.

ABSTRACT

Given a cloud of measurement points from the surface of a 3D object, we address the problem of recognizing and reconstructing special surface types. We survey our work on this problem, which is based on approximation in the space of lines and in the space of planes. Moreover, we discuss new generalizations which also use a recently developed technique for parametric surface fitting with an active contour model.

1 INTRODUCTION

Modern 3D measurement devices (laser range scanners, structured light based measurement, ...) produce a large amount of 3D data of geometric objects. These data are more or less structured point clouds. We have a variety of methods for processing these clouds of points: triangulation, mesh decimation (Garland Heckbert, 1997), reverse engineering through surface fitting (Varady et al., 1998). Together with rapid prototyping and 3D printing, we possess a complete chain for the emerging area of *3D technology*. For the essential steps such as data acquisition, CAD model building, model modification and printing there are already good solutions on the market.

Whereas the basic concepts and algorithms for 3D Vision and Reverse Engineering of geometric objects are available, the degree of automation and intelligence in the systems still has to be increased. A reverse engineering system should not just fit any surface to the data as long as it is within tolerance. For several reasons including functionality and the choice of the right manufacturing tools, it is important to recognize special shapes and build an according CAD model (Varady et al., 1998).

In the present paper, we survey our recent progress on the recognition and reconstruction of special surfaces from point clouds. These surfaces are the basic shapes of any CAD system (plane, sphere, cylinder and cone of revolution, torus) and more general surfaces with a simple kinematic generation. The latter surfaces are sweep surfaces and include surfaces of revolution, helical surfaces, pipe surfaces, developables surfaces, ruled surfaces and translational surfaces. The methodology combines results of classical geometry with techniques of geometric computing. The main idea is to estimate surface normals (or equivalently tangent planes) at the data points and then solve certain approximation problems in the space of lines or the space of planes, respectively.

We will first briefly introduce the basic geometric concepts and then discuss their application to the recognition and reconstruction of special surfaces. Finally, we point to ongoing research which also employs a type of 3D active contours for surface approximation.

2 APPROXIMATION IN LINE SPACE: FITTING A LINEAR COMPLEX TO A SET OF LINES

2.1 The linear line complex

Consider the motion of a rigid body in space. If $\mathbf{x}$ is a point in Euclidean three-space, the symbol $\mathbf{v}(\mathbf{x})$ denotes the velocity vector of that point of the moving body which is at this moment at position $\mathbf{x}$. Thus $\mathbf{v}(\mathbf{x})$ is a time-dependent vector attached to the point $\mathbf{x}$. It is well known that at some instant t, a smooth motion has a velocity vector field of the form

$$\mathbf{v}(\mathbf{x}) = \overline{\mathbf{c}} + \mathbf{c} \times \mathbf{x}, \tag{1}$$

with vectors $\mathbf{c}, \overline{\mathbf{c}}$, see e.g. (Bottema Roth, 1990). Thus the velocity vector field (or the *infinitesimal motion*) at some instant t is uniquely determined by the pair $(\mathbf{c}, \overline{\mathbf{c}})$.

Of special interest are the *uniform* motions, whose velocity vector field is constant over time. It is well known that apart from the trivial uniform motion, where nothing moves at all and all velocities are zero, there are the following three cases:

1. Uniform translations have $\mathbf{c} = \mathbf{o}$, but $\overline{\mathbf{c}} \neq \mathbf{o}$, i.e., all velocity vectors equal $\overline{\mathbf{c}}$.

2. Uniform rotations with nonzero angular velocity about a fixed axis. We have $\mathbf{c} \cdot \overline{\mathbf{c}} = 0$, but $\mathbf{c} \neq \mathbf{o}$.

3. Uniform helical motions are the superposition of a uniform rotation and a uniform translation parallel to the rotation's axis. They are characterized by $\mathbf{c} \cdot \overline{\mathbf{c}} \neq 0$.

   If ω is the angular velocity of the rotation, and v the velocity of the translation, then $p = v/\omega$ is called the *pitch* of the helical motion. We use the convention that ω is nonnegative, that $p > 0$ for right-handed helical motions, and that $p < 0$ for left-handed ones.

   Formally, $p = 0$ means a uniform rotation and $p = \infty$ is a translation.

All possible pairs $(\mathbf{c}, \overline{\mathbf{c}})$ actually occur, so we can use these three cases to classify the type of velocity vector field at one instant of an arbitrary smooth motion: *Infinitesimal translations* are characterized by $\mathbf{c} = \mathbf{o}$, and *infinitesimal*

rotations by $\mathbf{c} \cdot \overline{\mathbf{c}} = 0$. The remaining velocity vector fields are said to belong to *infinitesimal helical motions*. At all instants, the velocity vector field of a smooth motion belongs to one of the three cases, if it is nonzero.

It turns out that it is useful to study *path normals* of motions, i.e. lines that are orthogonal to the velocity vector of one of their points. Here it is convenient to describe lines by their *Plücker coordinates*. If a line G contains a point $\mathbf{p}$ and is parallel to the vector $\mathbf{v}$, then the pair $(\mathbf{g}, \overline{\mathbf{g}}) = (\mathbf{v}, \mathbf{p} \times \mathbf{v})$ is called its Plücker coordinate vector. It is easy to see that $\overline{\mathbf{g}}$ does not depend on the particular choice of $\mathbf{p}$. The Plücker coordinate vector is unique only up to scalar multiples. Its two components $\mathbf{g}$, $\overline{\mathbf{g}}$ are not independent, but fulfill the relation $\mathbf{g} \cdot \overline{\mathbf{g}} = 0$. A point $\mathbf{x}$ is contained in G if and only if $\mathbf{x} \times \mathbf{g}$ equals $\overline{\mathbf{g}}$.

Connections between Plücker coordinates of lines and velocity vector fields are shown by the following two lemmas. For a more detailed treatment, see (Pottmann Wallner, 2001).

Lemma 1 *A line with Plücker coordinates* $(\mathbf{g}, \overline{\mathbf{g}})$ *is a path normal of a smooth motion* $(\mathbf{c}, \overline{\mathbf{c}})$*, if and only if* $\mathbf{c} \cdot \overline{\mathbf{g}} + \overline{\mathbf{c}} \cdot \mathbf{g} = 0$.

Lemma 2 *If* $(\mathbf{c}, \overline{\mathbf{c}})$ *represents the velocity vector field of a uniform rotation or helical motion, then the Plücker coordinates* $(\mathbf{g}, \overline{\mathbf{g}})$ *of the axis, the angular velocity* ω *and the pitch* p *are reconstructed by*

$$p = \mathbf{c} \cdot \overline{\mathbf{c}} / \mathbf{c}^2, \quad \omega = \|\mathbf{c}\|, \quad (\mathbf{g}, \overline{\mathbf{g}}) = (\mathbf{c}, \overline{\mathbf{c}} - p\mathbf{c}). \quad (2)$$

2.2 Approximation of a set of lines by a linear line complex

We consider a set of lines $N_1, N_2, \ldots$, represented by their Plücker coordinates $(\mathbf{n}_i, \overline{\mathbf{n}}_i)$. Here, the direction vectors shall be normalized, $\mathbf{n}_i^2 = 1$.

We would like to approximate these lines with a linear complex C which consists of all lines $(\mathbf{x}, \overline{\mathbf{x}})$ which satisfy the linear equation $\mathbf{c} \cdot \overline{\mathbf{x}} + \overline{\mathbf{c}} \cdot \mathbf{x} = 0$. C shall be represented by the coefficients $(\mathbf{c}, \overline{\mathbf{c}})$ of this equation. Using as a deviation measure of a line N to a linear complex C the so-called moment $M(N, C)$ (Pottmann Wallner, 2001), the minimization of the squared sum of moments $\sum M(N_i, C)$ amounts to the minimization problem

$$F(\mathbf{c}, \overline{\mathbf{c}}) = \sum (\mathbf{c} \cdot \overline{\mathbf{n}}_i + \overline{\mathbf{c}} \cdot \mathbf{n}_i)^2 \to \min, \quad (\|\mathbf{c}\| = 1). \quad (3)$$

F is a quadratic function of six real arguments, and the side condition $\|\mathbf{c}\| = 1$ is also quadratic. We can therefore rewrite Equ. (3) in the form

$$(\mathbf{c}, \overline{\mathbf{c}})^T \cdot K \cdot (\mathbf{c}, \overline{\mathbf{c}}) \to \min, \quad (\mathbf{c}, \overline{\mathbf{c}})^T \cdot D \cdot (\mathbf{c}, \overline{\mathbf{c}}) = 1, \quad (4)$$

with two (6×6)-matrices K and D. The matrix D has nonzero entries only in its upper left 3×3 corner. The solution of this problem is straightforward. The minimum is assumed for $(\mathbf{c}, \overline{\mathbf{c}})$ which fulfills

$$\begin{array}{l} (K - \lambda D) \cdot (\mathbf{c}, \overline{\mathbf{c}}) = (\mathbf{o}, \mathbf{o}), \quad \|\mathbf{c}\| = 1, \\ \det(K - \lambda D) = 0, \quad \lambda \text{ minimal.} \end{array} \quad (5)$$

This means that we have to choose the smallest solution λ of the cubic equation $\det(K - \lambda D) = 0$ and solve the equation $(K - \lambda D)(\mathbf{c}, \overline{\mathbf{c}}) = (\mathbf{o}, \mathbf{o})$. Details can be found in (Pottmann Randrup, 1998, Pottmann Wallner, 2001).

3 APPROXIMATION IN THE SET OF PLANES

Problems in geometric computing which involve sets of planes can sometimes be transformed to problems for points by application of a projective duality. This is true as long as only projective and algebraic properties are involved. As soon as we perform approximation, we need distance measures, also in the space of planes.

We have shown in earlier papers how to solve this problem by introducing a Euclidean metric in the space of planes (see e.g. (Pottmann Wallner, 2001)): For that, it is necessary to remove a bundle of planes, which in our approach are the planes parallel to the z-axis in an appropriate Cartesian coordinate system (x, y, z). Planes not parallel to the z-axis can be written in the form

$$z = u_0 + u_1 x + u_2 y. \quad (6)$$

We see that (u_0, u_1, u_2) are affine coordinates in the resulting affine space A^* of planes not parallel to the z-axis.

We will now introduce a Euclidean metric in A^*. Thereby we make sure that the deviation between two planes shall be measured within some region of interest. This region shall be captured by its projection Γ onto the xy-plane.

For a positive measure μ in $\mathbb{R}^2$ we define the distance d_μ between planes $A = (a_0, a_1, a_2)$ and $B = (b_0, b_1, b_2)$ as

$$d_\mu(A, B) = \|(a_0 - b_0) + (a_1 - b_1)x + (a_2 - b_2)y\|_{L^2(\mu)}, \quad (7)$$

i.e., the $L^2(\mu)$-distance of the linear functions whose graphs are A and B. This, of course, makes sense only if the linear function which represents the difference between the two planes is in $L^2(\mu)$. We will always assume that the measure μ is such that all linear and quadratic functions possess finite integral.

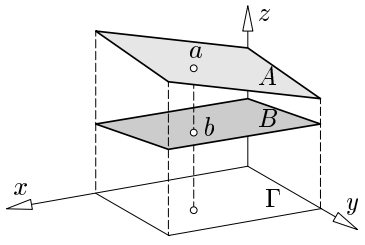

Figure 1: To the definition of the deviation of two planes.

A useful choice for μ is the Lebesgue measure $dxdy$ times the characteristic function χ_Γ of the *region of interest* Γ (Fig. 1). If $\mu = dxdy\chi_\Gamma$, we have

$$d_\mu(A,B)^2 = \int_\Gamma ((a_0-b_0)+(a_1-b_1)x+(a_2-b_2)y)^2 dxdy. \tag{8}$$

We write $d_\Gamma(A,B)$ instead of $d_\mu(A,B)$. With $c_i := a_i - b_i$, equation (8) can be written as

$$d_\Gamma(A,B)^2 = (c_0,c_1,c_2)\cdot \begin{pmatrix} \int 1 & \int x & \int y \\ \int x & \int x^2 & \int xy \\ \int y & \int xy & \int y^2 \end{pmatrix} \cdot \begin{pmatrix} c_0 \\ c_1 \\ c_2 \end{pmatrix}. \tag{9}$$

This is a quadratic form, whose matrix depends on the domain of integration Γ for the integrals (where we omitted the symbols $dxdy$ for brevity).

Another possibility is that μ equals the sum of several point masses at points (x_j, y_j). In this case we have

$$d_\mu(A,B)^2 = \sum_j ((a_0-b_0)+(a_1-b_1)x_j+(a_2-b_2)y_j)^2. \tag{10}$$

It can easily be shown (Pottmann Wallner, 2001), that *the distance d_μ defines a Euclidean metric in the set of planes of type (6), if and only if μ is not concentrated in a straight line*.

In this way, approximation problems in the set of planes are transformed into approximation problems in the set of points in Euclidean 3-space, whose metric is based on d_μ.

The introduced metric depends on the *choice of the reference direction*, which we identified with the z-axis of the underlying coordinate system. With decreasing angle between the considered planes and the reference direction, the distance d_μ is becoming geometrically meaningless. Therefore, it might be necessary to use *different reference directions* in order to fully cover the space of planes appropriately. This results in different local mappings of the space of planes to affine 3-space and in different Euclidean metrics. For the application we are dealing with in the next section, this strategy is sufficient.

In a recent paper (Peternell Pottmann, 2001), the metric in the space of planes is investigated further and new applications of this concept are presented. There, it is also discussed how to measure distances for all planes; however, we do no longer obtain a Euclidean metric in the space of planes then.

3.1 Application to the detection and reconstruction of planar faces in point clouds

In the following we are interested in the detection and reconstruction of planar faces in point clouds. A known solution to this problem uses the Gaussian sphere (Varady et al., 1998). For each data point, one locally fits a plane to the point and its nearest neighbors. The unit normal vectors of those planes describe points on the Gaussian unit sphere. Points from a planar face will give rise to nearly identical local regression planes and thus to a point cluster on the Gaussian sphere. The detection of planar faces is reduced to the detection of point clusters on the Gaussian sphere. An obvious disadvantage of this approach is that we lose information when going from the regression plane to the Gaussian sphere. Parallel planar faces cannot be separated directly on the Gaussian sphere. Moreover, the loss of information is critical in case of noisy data and complicated objects.

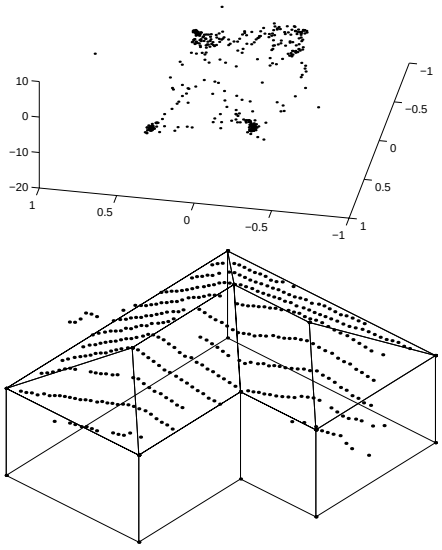

Figure 2: Top: Image points of planes of regression. Bottom: Data points of roof points and the reconstructed building.

These drawbacks can be overcome if we use approximation in the space of planes: The local regression planes determine points in dual space. We use the Euclidean distance introduced there to detect point clusters. For each cluster, we then determine those original data points, which are close to the regression planes that determine the cluster. These data points belong to a planar region. Its plane can easily be computed as a regression plane.

We provide an example from the reconstruction of 3D urban models from airborne laser scanner data. This is currently an important research topic in geodesy and photogrammetry. We do not review the literature on this problem here, but just point to the very recent paper (Vosselmann Dijkman, 2001). There, an extension of the Hough transform to 3 dimensions is used to recognize planar faces of the buildings' roofs. This approach has some similarity to ours, since the Hough transform is also a special duality. However, in our method the *metric* in the space of planes plays a crucial role and thus it can be expected that the present approach is more reliable.

The example displayed in Fig. 2 shows both the point cluster in dual space and the reconstructed building. Our

method just gives the roof. The vertical walls have been taken from a given top view.

4 RECOGNITION AND RECONSTRUCTION OF SPECIAL SURFACES

4.1 Surfaces of revolution and helical surfaces

The detection and reconstruction of rotational and helical surfaces has been studied in (Pottmann Randrup, 1998). One first estimates surface normals of the data points and then fits a linear line complex to those normals (Pottmann Wallner, 2001). As we have seen in the previous section, this approximation problem in line space requires the solution of a general eigenvalue problem. In the case that a good fit is possible, the characteristics of the linear complex allow us to compute the kinematic generation of the underlying shape, i.e., the rotational axis or the helical axis plus pitch or the translational direction in case of a cylinder surface. It is then rather simple to compute a generating profile curve and finally the approximating surface.

Special surfaces such as sphere and cylinder of revolution can be detected via special distributions of the general eigenvalues in the eigenvalue problem (Pottmann Randrup, 1998). For solutions of the problem of fitting special surfaces (sphere, cylinder and cone of revolution, torus) based on their representation as algebraic varieties, we refer to (Lukacs et al., 1998).

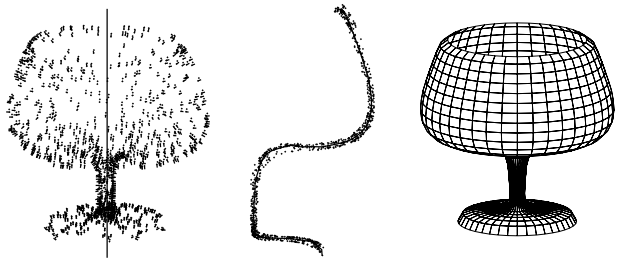

Figure 3: Reconstruction of a surface of revolution: Left: data points, estimates of normal vectors, and axis computed from this estimation. Center: points projected onto a plane and a curve approximating this point set, Right: final surface of revolution.

As an example we consider scattered data (e.g. obtained by a laser scanner) from an object whose boundary is a surface of revolution. The surface normals at the data points are estimated (see Fig. 3, left) using local quadric fits as in (Varady et al., 1998). The pitch p of an approximating linear line complex in this case is nearly zero, which shows that the original data come from a surface of revolution. We let $p = 0$ and project the input data into a half-plane which contains the axis (Fig. 3, center). The curve which fits these points was found by a moving least squares method according to (Lee, 2000).

4.2 Moulding surfaces, in particular pipe surfaces and delopable surfaces

There are surfaces which are locally well approximated by surfaces of revolution. One class of such surfaces are smooth surfaces which have a kind of 'osculating' simpler surface analogous to an osculating circle. *Pipe surfaces*, which are generated as the envelope of a moving sphere, are locally well approximated by tori. *Moulding surfaces*, which are generated by a planar curve, whose plane is rolling on a developable surface, are locally well approximated by surfaces of revolution (do Carmo, 1976).

A second class are surfaces composed of several different pieces of simple surfaces. This includes most surfaces of parts used e.g. in mechanical engineering. Surfaces which do not consist of pieces of planes, cylindrical surfaces, spheres, surfaces of revolution, and helical surfaces are rare in many areas of application.

To reconstruct either type of surface in a satisfactory manner, we have to consider the problem of deciding which subsets of a given point cloud are well approximated by the simple surfaces mentioned above. A solution is provided by a suitable *region growing* algorithm, which grows an initially small subset until no simple surface fits well enough.

An application of this is the recovery of pipe surfaces (see Fig. 4). Parts of such surfaces appear as constant radius rolling ball blends in reverse engineering (Kós et al., 2000). The reconstruction of pipe surfaces is based on locally approximating tori.

For data from a pipe surface, locally approximating tori have nearly the same pipe radius. We use the mean of the computed radii as radius r of the pipe surface. Offsetting the data points by a distance r in inward normal direction, we should ideally end up at points of the spine curve. Due to various errors (data, normal estimates, estimation of r), we get a thin cloud of points along the spine curve. Fitting a curve to these points (see Fig. 4, middle), we obtain the spine curve and together with r the pipe surface is finally determined (see Fig. 4, right).

Using locally approximating general surfaces of revolution, we can also reconstruct moulding surfaces (Lee et al., 1999).

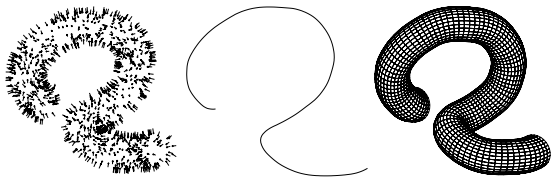

Figure 4: Pipe surface: Left: data points and estimates of normal vectors, Center: approximate spine curve, Right: reconstruction of pipe surface.

4.3 Developable surfaces

Developable surfaces are special moulding surfaces, namely those with a straight line as profile curve. Specializing the strategy for general moulding surfaces, the reconstruction of developable surfaces may be performed with local fits by right circular cones or cylinders (Chen et al., 1999).

Note that a developable surface is the envelope of a one-parameter family of planes. Given scattered data points, we may estimate tangent planes at data points and view them as points in dual space. Using a metric in the space of planes as discussed in section 3, we can fit a curve to the resulting point cloud and interpret it as dual model of an approximating developable surface (Peternell Pottmann, 2001).

4.4 Active contours for the reconstruction of ruled or translational surfaces

An efficient approach to various approximation problems for curves and surfaces are *active contour models*, which are mainly used in Computer Vision and Image Processing. The origin of this technique is the seminal paper (Kass et al., 1988), where a variational formulation of parametric curves, called *snakes*, is presented for detecting contours in images. There are various other applications and a variety of extensions of the snake model (see e.g. (Blake Isard, 1998, Malladi et al., 1995)).

Recently we have developed an active contour type strategy for approximating a point cloud or a surface in any representation ('model shape') by a B-spline surface or another surface type which can be written as linear combination of bivariate basis functions (Pottmann Leopoldseder, 2002). This technique is based on local quadratic approximants of the squared distance function to curves and surfaces (Pottmann Hofer, 2002). There it is described how to compute for any point $\mathbf{p} \in \mathbb{R}^3$ such a local quadratic approximant $F_{d,\mathbf{p}}$. The surface approximation method proceeds in the following steps:

1. Initialize the 'active' B-spline surface and determine the boundary conditions. This requires the computation of an initial set of control points, the proper treatment of boundaries (e.g. by fixing vertices of a patch) and the avoidance of model shrinking during the following steps.

2. Repeatedly apply the following steps a.–c. until the approximation error or change in the approximation error falls below a user defined threshold:

a. With the current control points, compute a set of points $\mathbf{s}_k$ of the active surface, such that the shape of the active surface is well captured. For each of the points $\mathbf{s}_k$ determine a local quadratic approximant $F_{d,\mathbf{s}_k} =: F_d^k$ of the squared distance function to the model shape at the point $\mathbf{s}_k$. In an appropriate coordinate system, this has to be the graph of a nonnegative quadratic function, $F_d^k(\mathbf{x}) \geq 0, \forall \mathbf{x} \in \mathbb{R}^3$.

b. Compute displacement vectors for the control points by minimizing the functional

$$F = \sum_{k=1}^{N} F_d^k(\mathbf{s}_k^*) + \lambda F_s, \tag{11}$$

where $\mathbf{s}_k^*$ denote the displaced surface points (which depend linearly on the unknown displacement vectors of the control points) and F_s denotes a smoothing functional which shall be quadratic in the unknown displacement vectors. Thus, our goal is to bring the new surface points $\mathbf{s}_k^*$ closer to the model shape than the old surface points $\mathbf{s}_k$. Since the points $\mathbf{s}_k^*$ depend linearly on the unknown displacement vectors of the control points, both F_d^k and F are quadratic in the unknowns.

We see that this step requires the minimization of a function F which is quadratic in the displacement vectors of the control points. This amounts to the solution of a linear system of equations.

c. With the displacement vectors from the previous step, update the control points of the active surface.

An important advantage of the new technique is that it is not necessary to deal with the correspondence between points in the parameter domain and the data points. Thus problems where this correspondence is crucial can now be easier handled.

One of these problems concerns the approximation of a given surface or point cloud by a ruled surface. *Ruled surfaces* are interesting from various points of view (Pottmann Wallner, 2001). Because they carry a one-parameter family of straight lines, their use in architecture is much simpler than that of more general freeform shapes. They can be manufactured with wire EDM (Yang Lee, 1996), and the approximation with ruled surfaces also appears in the context of NC machining with a peripheral milling strategy and a cylindrical milling tool (Lee Koc, 1998). There is prior work on ruled surface approximation (Chen Pottmann, 1999, Hoschek Schwanecke, 1998, Pottmann Wallner, 2001). In the new surface approximation strategy we just have to use tensor product B-splines of bidegree $(1, n)$ as active contours because these are ruled surfaces.

Another application of the new approximation technique concerns the approximation of a given surface or point cloud by a translational surface. A *translational surface* $\mathbf{x}(u, v)$ is generated by a translatory motion of a curve $\mathbf{c}(u)$ along another curve $\mathbf{d}(v)$. Assuming that the two curves share a common point $\mathbf{a} = \mathbf{c}(0) = \mathbf{d}(0)$, the surface parameterization is given by

$$\mathbf{x}(u, v) = \mathbf{c}(u) + \mathbf{d}(v) - \mathbf{a}. \tag{12}$$

Translational surfaces are very well studied in classical geometry. Because of the simple generation, they are used for various applications, e.g. in architecture.

For the reconstruction of a translational surface from a point cloud, or the approximation of a given (not exactly translational) surface by a translational surface, the concepts of (Pottmann Leopoldseder, 2002) are again applicable. We note that a translational B-spline surface has control points which satisfy the constraints

$$\mathbf{d}_{i,j} = \mathbf{d}_{i,0} + \mathbf{d}_{0,j} - \mathbf{d}_{0,0}.$$

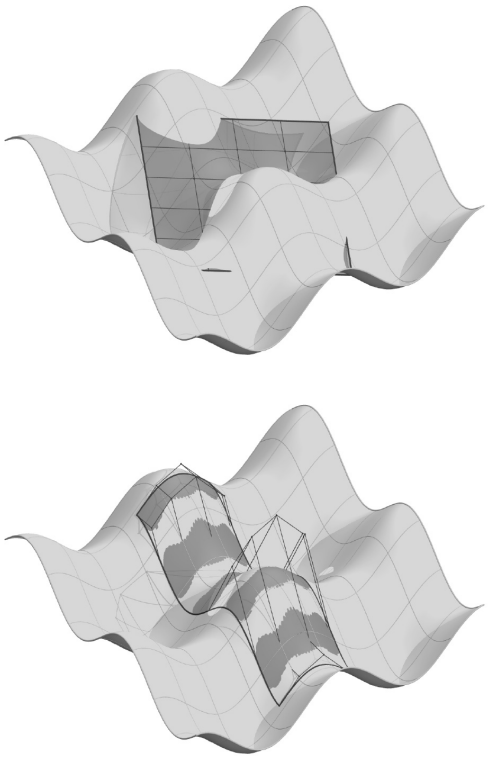

Figure 5: Surface approximation with translational surface

As an example, Fig. 5 shows the initial (top) and the final position (bottom) of a translational B-spline surface (dark surface) that approximates a given translational surface (light surface). The control points are iteratively recomputed as described above, such that the B-spline surface 'flows' towards the target shape. As a boundary condition, only two diagonally opposite vertices of the moving surface are kept fixed. Note that no information on the direction and shape of the translated curves $\mathbf{c}(u)$ and $\mathbf{d}(v)$ of the target surfaces have been used. These characteristic curves have been 'detected' by our algorithm.

Future research on this topic will include the recognition of ruled surfaces and translational surfaces, applying methods of both projective and affine differential geometry (Blaschke, 1923, Bol, 1950).

ACKNOWLEDGEMENTS

This work has been carried out in part within the K plus Competence Center *Advanced Computer Vision* and was funded from the K plus program.

REFERENCES

Blake, A., Isard, M., 1998. *Active Contours*. Springer.

Blaschke, W., 1923. *Vorlesungen über Differentialgeometrie II: Affine Differentialgeometrie*. Springer, Berlin.

Bol, G., 1950, 1954, 1967. *Projektive Differentialgeometrie I, II, III*. Vandenhoeck & Ruprecht, Göttingen.

Bottema, O., Roth, B., 1990. *Theoretical Kinematics*. Dover Publ., New York.

Chen, H.-Y., Lee, I.-K., Leopoldseder, S., Pottmann, H., Randrup, T., Wallner, J., 1999. On surface approximation using developable surfaces. *Graphical Models and Image Processing*, 61, pp. 110–124.

Chen, H.-Y., Pottmann, H., 1999. Approximation by ruled surfaces. *J. Comput. Applied Math.*, 102, pp. 143–156.

do Carmo, M. P., 1976. *Differential Geometry of Curves and Surfaces*. Prentice Hall, Englewood Cliffs, New York.

Garland, M., Heckbert, P., 1997. Surface simplification using quadric error metrics. *SIGGRAPH 97 Proc.*, pp. 209–216.

Hoschek, J., Schwanecke, U., 1998. Interpolation and approximation with ruled surfaces. In: *The Mathematics of Surfaces VIII (R. Cripps, ed)*. Information Geometers, pp. 213–231.

Kass, M., Witkin, A., Terzopoulos, D., 1988. Snakes: Active contour models. *Intern. J. Computer Vision*, 1, pp. 321–332.

Kós, G., Martin, R. R., Várady, T., 2000. Methods to recover constant radius rolling ball blends in reverse engineering. *Computer Aided Geom. Design*, 17, pp. 127–160.

Lee, I.-K., 2000. Curve reconstruction from unorganized points. *Computer Aided Geom. Design*, 17, pp. 161–177.

Lee, Y.S., Koc, B., 1998. Ellipse-offset approach and inclined zig-zag method for multi-axis roughing of ruled surface pockets. *Computer-Aided Design* 30(12), pp. 957–971.

Lee, I.-K., Wallner, J., Pottmann, H., 1999. Scattered data approximation with kinematic surfaces. *Proceedings of SAMPTA'99 Conference*, Loen, Norway, pp. 72–77.

Lukács, G., Marshall, A. D., Martin, R. R., 1998. Faithful least-squares fitting of spheres, cylinders, cones and tori for reliable segmentation. In: *Computer Vision — ECCV '98, Lecture Notes in Computer Science*, Springer.

Malladi, R., Sethian, J. A., Vemuri, B. C., 1995. Shape modeling with front propagation: A level set approach, *IEEE Trans. Pattern Anal. and Machine Intell.*, 17, pp. 158–175.

Peternell, M., Pottmann, H., 2001. Approximation in the space of planes — Applications to geometric modeling and reverse engineering. Technical Report 87, Institute of Geometry, Vienna Univ. of Technology.

Pottmann, H., Hofer, M., 2002. Geometry of the squared distance function to curves and surfaces. Technical Report 90, Institute of Geometry, Vienna Univ. of Technology.

Pottmann, H., Leopoldseder, S., 2002. A concept for parametric surface fitting which avoids the parametrization problem. Preprint, Insitute of Geometry, Vienna Univ. of Technology.

Pottmann, H., Randrup, T., 1998. Rotational and helical surface reconstruction for reverse engineering. *Computing*, 60, pp. 307–322.

Pottmann, H., Wallner, J., 2001. *Computational Line Geometry*. Springer-Verlag.

Várady, T., Benkő, P., Kós, G., 1998. Reverse engineering regular objects: simple segmentation and surface fitting procedures. *Int. J. Shape Modeling*, 4, pp. 127–141.

Vosselman, G., Dijkman, S., 2001. 3D building model reconstruction from point clouds and ground plans. *Intl. Archives of Photogrammetry and Remote Sensing*, Vol. XXXIV-3/W4, Annapolis, MD, pp. 37–43.

Yang, M., Lee, E., 1996. NC verification for wire-EDM using an R-map. *Computer Aided Design*, 28, pp. 733–740.