SCATTERED DATA APPROXIMATION WITH KINEMATIC SURFACES

In-Kwon Lee

POSTECH
Information Research Laboratories
Pohang Univ. of Science and Technology
San 31 Hyoja-Dong
Pohang 790-784, South Korea
e-mail: iklee@postech.ac.kr

Johannes Wallner & Helmut Pottmann

Institut für Geometrie
Technische Universität Wien
Wiedner Hauptstraße 8–10
A-1040 Wien, Austria
wallner@geometrie.tuwien.ac.at
pottmann@geometrie.tuwien.ac.at

ABSTRACT

In this paper, we present algorithms to reconstruct surfaces generated by kinematic motions, such as the rotation about an axis or the rolling of a plane on a developlable surface. They share the property that a two-dimensional set of scattered data can be transformed into a one-dimensional one when we are able to detect the kinematics behind the data. We also present a technique to reconstruct a smooth curve from a point cloud, which we use in an essential way.

1. INTRODUCTION

Reconstruction of surfaces from point sets such as data from a 3D scanner has a wide range of applications. The process of transforming real parts into computer models is often called *reverse engineering* of geometric models. For an introduction to the basic concepts of reverse engineering and a survey of the state of the art we refer the reader to Varady et al. [13].

Two important steps in the process of reverse engineering are *segmentation* and *surface fitting*. There, the given point set is grouped into subsets each of which has an appropriate single surface to be fitted to the subset. In this paper, we present algorithms to reconstruct surfaces from sets of points: On the one hand, we study surfaces which are invariant with respect to some group of motions, such as helical surfaces, spiral surfaces, surfaces of revolution (rotational surfaces), and cylindrical (translational) surfaces. On the other hand, we consider surfaces swept by curves during a motion such that they can be well approximated by surfaces of the first type: These include pipe surfaces, profile surfaces and developable surfaces.

Most of our algorithms assume that the given point set does not need more segmentation, i.e., at least one single surface (which is one of the above types) can be fitted to the set.

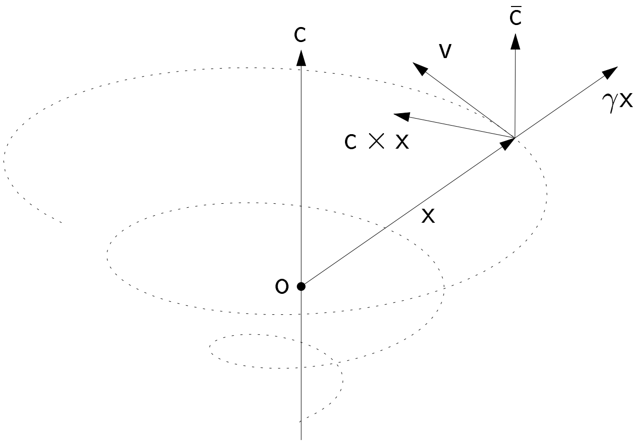


Figure 1: Velocity $\mathsf{v}(\mathsf{x})$ of a spiral motion assuming that the axis passes through the origin (false proportions)

2. EQUIFORM AND EUCLIDEAN MOTIONS

Let $M(t)$ be a one-parameter group of motions of Euclidean three-space. This means that $M(t)$ is a rigid body motion for all real numbers t and that $M(t)M(s) = M(t+s)$. It is well known that $M(t)$ is either a translation along the vector $t\cdot\mathsf{v}$, a rotation about the angle $\omega\cdot t$ or a helical motion, which is a superposition of both. More generally, we consider one-parameter subgroups of equiform motions, i.e., a superposition of a scaling and a rigid body motion. Then $M(t)$ is either one of the above, or a continuous central similarity, or a spiral motion.

The velocity vector field of $M(t)$ is constant, and it is well known that it is of the form

$$\mathsf{v}(\mathsf{x}) = \bar{\mathsf{c}} + \gamma\mathsf{x} + \mathsf{c}\times\mathsf{x}, \qquad (1)$$

where $\bar{\mathsf{c}}$, $\gamma\mathsf{x}$, and $\mathsf{c}\times\mathsf{x}$ represent the translational, scale, and rotational component of $\mathsf{v}(\mathsf{x})$, respectively (see Figure 1).

The following list illustrates the classification of $M(t)$

in terms of its velocity field, and the type of surface generated by a curve by the action of $M(t)$.

- $\gamma = 0$:
  - $\mathsf{c} = 0, \bar{\mathsf{c}} = 0$: $M(t)$ is the identitical motion.
  - $\mathsf{c} = 0, \bar{\mathsf{c}} \neq 0$: $M(t) =$ is a translation along $\bar{\mathsf{c}}$, and S is a cylinder.
  - $\mathsf{c} \neq 0, \mathsf{c} \cdot \bar{\mathsf{c}} = 0$: $M(t)$ is rotation about an axis parallel to c, S is surface of revolution .
  - $\mathsf{c} \neq 0, \mathsf{c} \cdot \bar{\mathsf{c}} \neq 0$: $M(t)$ is helical motion about an axis parallel to c, and S is a helical surface .
- $\gamma \neq 0$:
  - $\mathsf{c} \neq 0$: $M(t)$ is a spiral motion, S is a spiral surface.
  - $\mathsf{c} = 0$: $M(t)$ is a central similarity, S is a conical surface.

The process of reconstructing scattered data points d_i, $i = 1, \ldots, k$, with a surface generated by one of the special motions $M(t)$ in the enumeration above has four phases:

1. estimating the surface normal vector n_i for each d_i
2. computing the velocity field of $M(t)$
3. approximating a profile curve
4. generating a surface by the profile curve.

Estimation of the surface normals from scattered points is a well-studied subject in Computer Aided Geometric Design [2]; we shall omit the details. In this section we focus on step 2.

Let $\mathsf{C} = (\mathsf{c}, \bar{\mathsf{c}}, \gamma) \in \mathbb{R}^7$ represent the velocity field of a motion $M(t)$. We are looking for a motion $M(t)$ whose velocity vectors $\mathsf{v}(\mathsf{d}_i)$ at data points d_i form an angle α_i close to $\pi/2$ with the normal vectors n_i. Thus, we have to solve the nonlinear optimization problem which minimizes

$$G := \sum_{i=1}^{k} \cos^2 \alpha_i, \quad \text{where} \tag{2}$$

$$\begin{aligned} \cos \alpha_i &= \mathsf{v}(\mathsf{d}_i) \cdot \mathsf{n}_i / ||\mathsf{v}(\mathsf{d}_i)|| \\ &= \frac{\bar{\mathsf{c}} \cdot \mathsf{n}_i + \gamma \mathsf{d}_i \cdot \mathsf{n}_i + \mathsf{c} \cdot \bar{\mathsf{n}}_i}{||\bar{\mathsf{c}} + \gamma \mathsf{d}_i + \mathsf{c} \times \mathsf{d}_i||}, \end{aligned}$$

with $\bar{\mathsf{n}}_i := \mathsf{d}_i \times \mathsf{n}_i$.

Instead of solving this nonlinear optimization problem, we proposed [9] to minimize the positive semidefinite quadratic form

$$\begin{aligned} F(\mathsf{C}) &:= \sum_{i=1}^{k} (\bar{\mathsf{c}}\mathsf{n}_i + \gamma \mathsf{d}_i \mathsf{n}_i + \mathsf{c}\bar{\mathsf{n}}_i)^2 && (3) \\ &=: \mathsf{C}^T \cdot N \cdot \mathsf{C} && (4) \end{aligned}$$

subject to the normalization condition

$$1 = ||\mathsf{c}||^2 =: \mathsf{C}^T \cdot D \cdot \mathsf{C}. \tag{5}$$

Equation (5) sets the angular velocity of the rotational part to 1 and excludes pure translations and central similarities.

The solution of $F \to \min$ subject to (5) is a well-known general eigenvalue problem. Using a Lagrangian multiplier λ, we solve

$$(N - \lambda D) \cdot \mathsf{C} = 0, \ \mathsf{C}^T \cdot D \cdot \mathsf{C} = 1. \tag{6}$$

Hence, λ must be a root of the equation

$$\det(N - \lambda D) = 0. \tag{7}$$

Because $D = \operatorname{diag}(1,1,1,0,0,0,0)$, (7) is a cubic equation in λ. For any root λ and corresponding normalized general eigenvector C, we have $F(\mathsf{C}) = \lambda$. Therefore, all roots λ are nonnegative and the *solution* C *is a general eigenvector of N with respect to D, which belongs to the smallest general eigenvalue* λ.

We have so far excluded the case $\mathsf{c} = 0$, i.e., data that fit well a *cone* or a *cylinder*. If we apply our algorithm to data exactly fitting a cone or cylinder, the matrix $N - \lambda D$ is singular for all λ and thus all coefficients in the characteristic equation (7) vanish; for data close to a cone or cylinder we are facing a numerical instability. In order to detect this in advance, we first minimize F with the normalization $\mathsf{C}^2 = \mathsf{c}^2 + \bar{\mathsf{c}}^2 + \gamma^2 = 1$, which leads to an ordinary eigenvalue problem in $\mathbb{R}^7$. If the solution exhibits $||\mathsf{c}||, \gamma \ll ||\bar{\mathsf{c}}||$, we will search for a *translation* $M(t)$ by letting $\mathsf{c} = 0, \gamma = 0$ and minimizing $F = \sum_{i=1}^{k} (\bar{\mathsf{c}} \cdot \mathsf{n}_i)^2$ with the normalization $\bar{\mathsf{c}}^2 = 1$. This is equivalent to $G \to \min$ and requires the solution of an eigenvalue problem in $\mathbb{R}^3$ (see [11] for more details). Analogously $||\mathsf{c}|| \ll ||\bar{\mathsf{c}}||$ indicates that the data fit a conical surface, with M being a central similarity.

Let us now continue with the case $\mathsf{c} \neq 0$ and assume we have found a solution as indicated above. If γ turns out to be small compared to $||\mathsf{C}||$, we let $\gamma = 0$ and obtain a helical motion $M(t)$ by minimizing (4), which is a a general eigenvalue problem in $\mathbb{R}^6$ (see [9]). If γ is nonzero, $M(t)$ is a spiral motion. There is exactly one line invariant under the action of $M(t)$, which is called the *spiral axis* and which passes through the only invariant point, the *spiral center* z. This axis is parallel to the vector c.

Multiply invariant surfaces

There are surfaces which are invariant under more than one one-parameter subgroup. Let us briefly address some of the possible cases: A cylinder of revolution is invariant whith respect to all translations along its axis, all rotations about its axis, and all of their superpositions (helical motions).

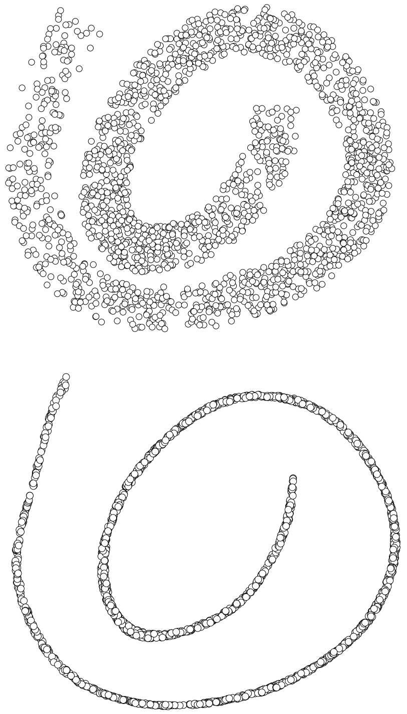


Figure 2: Source point cloud (top) and the result of improved moving least squares (bottom)

This situation is characterized by $\gamma = 0$ and two small eigenvalues (see [9]).

A sphere and a plane are invariant with respect to a three-parameter set of motions, but planar and spherical data are easily detected by the property that the surface normals must either be parallel or intersect in the center of the sphere.

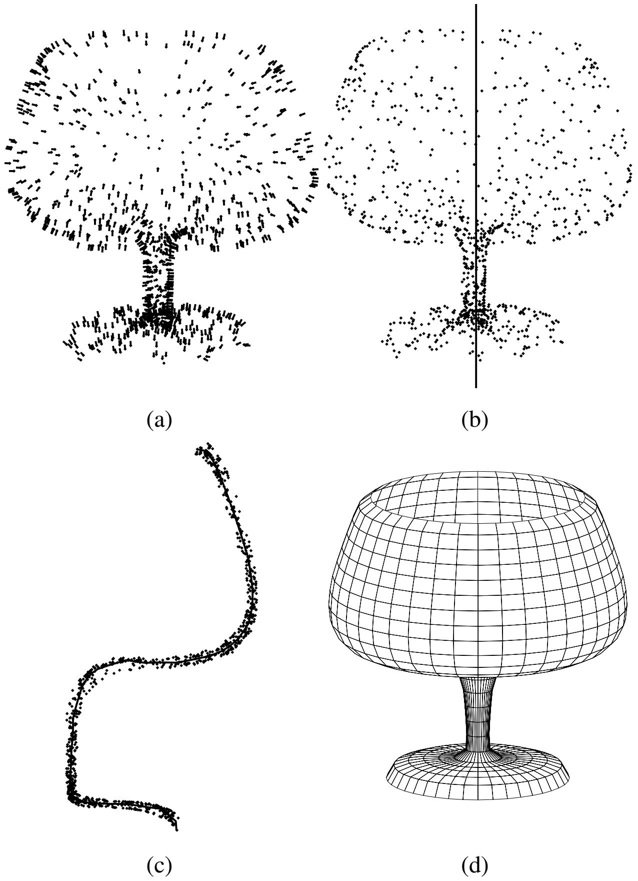
If $\gamma \neq 0$, three special cases which all may lead to numerical instabilities should be mentioned. First, data may be close to a *cone of revolution*. These cones are invariant by a two-parameter family of spiral motions, which also contains rotations and central similarities. Thus, two small eigenvalues will occur. We let $\gamma = 0$ and obtain a (stable) general eigenvalue problem in $\mathbb{R}^6$ when minimizing F. A weight iteration allows to solve $G \rightarrow \min$ (see [7] for more details of weight iteration).

Another case of surfaces which are invariant under a family of spiral motions are spiral cylinders, which are traced out by lines parallel to the spiral axis. Here we let $\gamma = 0$, $\mathsf{c} = 0$ and approximate our data by a cylinder as above.

3. FROM MOTION TO SURFACE

3.1. Simple kinematic surfaces

Having determined a generating motion $M(t)$, we look for a plane P which data points d_i can be projected into by intersecting P with their trajectories under the action of $M(t)$.



(a)

(b)

(c)

(d)

Figure 3: Reconstruction of surface of revolution: (a) data points and estimated normal vectors, (b) data points and computed axis, (c) points projected onto a plane and a curve approximating the point set, (d) final surface of revolution

If $M(t)$ is a rotation, we choose P such that it contains the rotation axis. In the case of a helical or spiral motion, we choose P orthogonal to an appropriate motion trajectory [9]. Maybe some trajectories intersect P more often than once. In this case we have to find a domain in P which intersects the trajectories of the point cloud exactly once.

In this plane we find a curve fitting the resulting thin cloud of data points. Once the curve is computed, the surface is reconstructed by applying $M(t)$ to the curve for all t. (see Figure 3).

To find a smooth curve fitting a set of points, we use a method based on moving least squares [3]. The moving least squares method [5, 6] is a powerful tool to reduce a point cloud to a thin curve-like shape which is a near-best approximation of the point set. The basic idea of moving least squares is to compute a simple regression curve or surface C_i for each data point d_i which locally fits a certain neighborhood of d_i, by using a weighted regression scheme.

Then d_i is moved to a new position d_i' on d_i. In [3], an improved moving least squares technique is suggested using the concepts of minimum spanning tree, region growing and refining iteration for point clouds in $\mathbb{R}^2$ and $\mathbb{R}^3$. (see Figure 2, showing the result of applying our moving least-squares technique to a point cloud). After thinning the point cloud, we can easily reconstruct a smooth curve [3].

3.2. Pipe Surfaces

As an application of curve reconstruction, we introduce an algorithm for reconstructing a *pipe surface*. A pipe surface is generated as envelope of a sphere with a constant radius whose center runs along a *spine curve*. When reconstructing the pipe surface, the sphere radius can be computed by using a technique described in section 2; we can compute a torus which locally fits the region after collecting a cluster of points from the given point set. Once the radius r is found, we translate each data point d_i by a vector $r\mathsf{n}_i$, where n_i is an estimated unit normal vector at d_i, always taking care of the directions of normal vectors. If a given point set is likely to represent a pipe surface, the translated points form a reasonably thin point cloud, which corresponds to the spine curve, and which our reconstruction algorithms can be applied to (see Fig. 4).

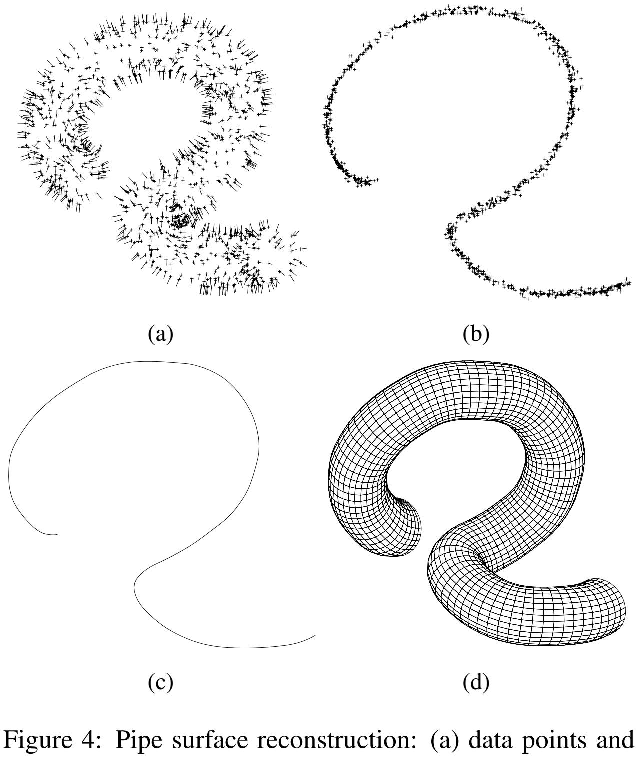


Figure 4: Pipe surface reconstruction: (a) data points and estimated normals, (b) data points translated by the radius of the swept sphere, (c) approximated spine curve, (d) reconstructed pipe surface

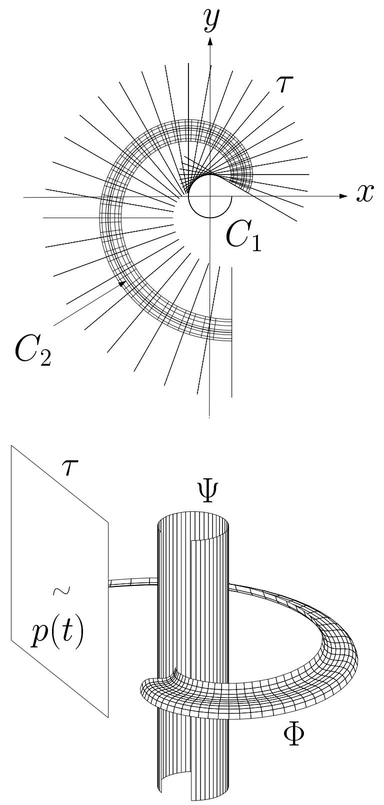


Figure 5: Example of a profile surface Φ generated from the cylinder axode Ψ (top) and top view (bottom)

4. PROFILE SURFACES AND DEVELOPABLE SURFACES

4.1. Profile surfaces

A *profile surface* Φ is traced out by a planar curve p in the tangent plane τ of a developable surface Ψ when τ is rolling on Ψ. A simple example can be seen in Fig. 5 (here Ψ is a cylinder).

The basic idea used in reconstructing a profile surface from given data points uses the fact that any profile surface possesses a family of osculating surfaces of revolution $\sigma(t)$ (see [7]). The algorithm finds a finite sequence of rotation axes A_i of the corresponding surfaces of revolution σ_i, where consecutive axes either intersect or are parallel. Then, the profile curves, at which consecutive surface segments σ_i, σ_{i+1} meet, lie in the planes α_i containing both A_i and A_{i+1}. (see Figure 6).

For collecting a point subset generating a σ_i, we use a technique called *region growing* [12]. Starting from a set of points called *seed region*, which is small both in number and in diameter, the region grows until it includes all points which can be approximated within a given tolerance by a

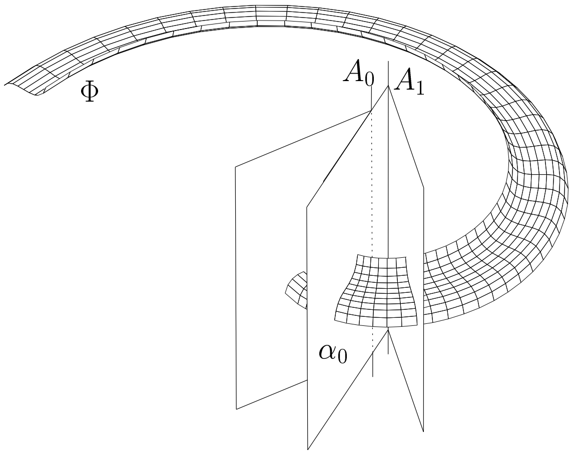


Figure 6: Rotation axes and boundary planes for two consecutive surface of revolution segments

single surface of revolution.

Having found an initial surface of revolution σ_i, this step is repeated recursively for an adjacent surface of revolution, σ_{i+1}, with the constraint that A_i and A_{i+1} intersect in the projective sense. Figure 7 shows the algorithm of reconstructing a profile surface.

4.2. Developable surfaces

A *developable surface* can be reconstructed by an algorithm similar to profile surface reconstruction. Before we describe this algorithm (see [1]), we want to tell the reader some basic facts about developable surfaces and *distances* between them, in order to make clear in what sense we approximate.

A developable surface, which by definition is a surface which can be developed unto a plane in an isometric way, is shown to consist of pieces of planes, cones, cylinders, and tangent surfaces of twisted space curves. In any case it carries straight lines (*rulings*) and the surface tangent plane is constant along such a ruling. This is in contrast to the non-developable ruled surfaces, such as the one-sheeted hyperboloid, and it makes it possible convert the surface, which we would naturally think of as a two-parameter set of points, into something one-parametric, namely its one-parameter family of tangent planes.

This makes it possible to define a distance between developable surfaces, more precisely, between two parametrizations of the family of tangent planes. We first define a distance between two planes $u : z = u(x, y) = u_0+u_1x+u_2y$, and $v : z = v(x, y) = v_0 + v_1x + v_2y$ by

$$d(u, v) = \int_D (u(x, y) - v(x, y))^2 dxdy,$$

where D is some region of interest in the xy-plane, and two developable surfaces, or rather their families of tangent planes $U = u(t)$, $V = v(t)$, then can be given the distance

$$d(U, V) = \sum_{i=0}^{m} d(u(t_i), v(t_i))^2,$$

or

$$d(U, V) = \int d(u(t), v(t))g(t)dt$$

with some weight function $g(t)$. For a more detailed discussion of these distance functions, and also a suitable choice of the coordinate system, see [10]. If we determine a plane by its equation as above, this means that we consider planes as points in some three-space. The distance function between planes defines a positive definite scalar product between them, and the distance function between families of planes defines accordingly a positive semidefinite scalar product in some function space. Thus we are able to reduce the problem of approximating developable surface to approximation problems in well-studied function spaces.

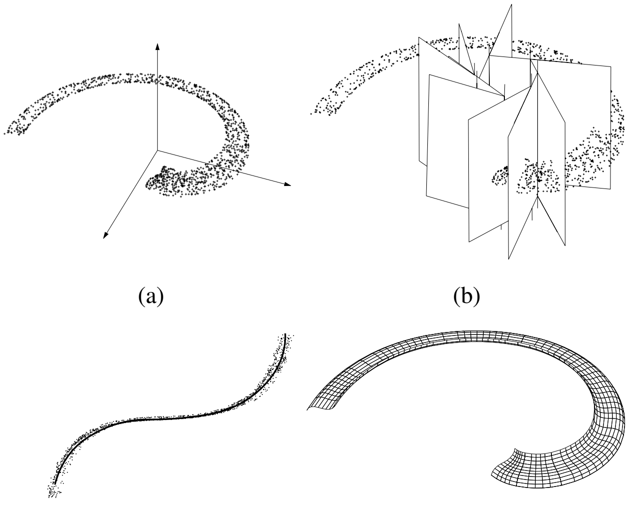


Figure 7: Profile surface reconstruction: (a) Data points, (b) seven wedges are classified, each of which is corresponding to a surface of revolution, (c) profile curve constructed from the set of projected data points, (d) resulting profile surface approximation having seven different surface of revolution patches

In order to be able to apply these distance functions to scattered data we first have to find the tangent planes or some approximations of them. We use the well-known fact that a developable surface possesses a family of *osculating cones*, which are locally fitted well by the given data. Thus we can apply a procedure similar to the reconstruction of profile surfaces, and after an appropriate segmentation we approximate the point cloud by a sequence of cones. [4] shows how to find, for a given sequence of cones, a sequence of auxiliary cones which completes the given set to a tangent-plane continuous developable surface. Its tangent planes define a continuous, piecewise smooth curve $U = u(t)$ which can be approximated by a spline curve of a suitable degree of smoothness, which in turn is nothing but a smooth developable surface approximating the original scattered data (see [1]). An example is shown in Fig. 8.

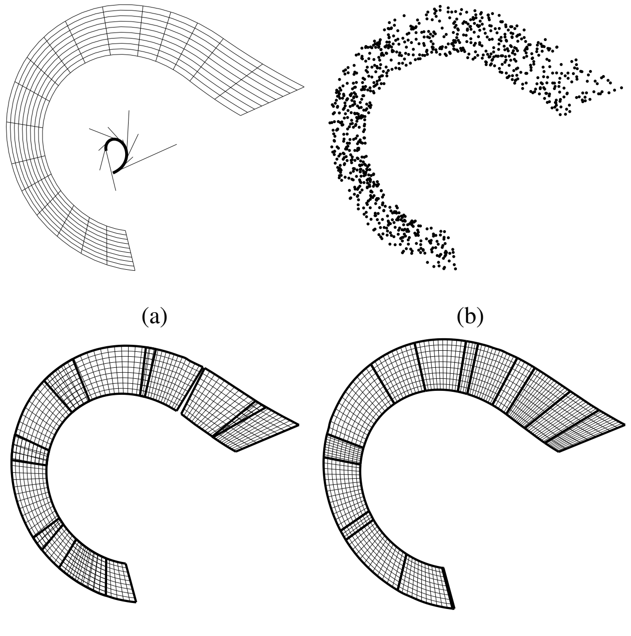

(a) (b) (c) (d)

Figure 8: Developable surface reconstruction: (a) original developable surface (tangent surface), (b) sample points from tangent surface with some perturbation, (c) 8 approximated cones, (d) reconstructed developable surface — cone spline surface

5. CONCLUSION

In this paper, we presented various algorithms to reconstruct surfaces generated by kinematic motions as well as a technique to reconstruct a smooth curve from a point cloud. Further interesting topics are the treatment of other subgroups of the group of affine motions, especially those with algebraic trajectories. The segmentation of data point clouds into subsets which fit well to kinematic surfaces is a tricky problem which is still being investigated.

Acknowledgements

This research was supported in part by Project No. P12252-MAT of the Austrian Science Foundation.

6. REFERENCES

[1] H.-Y. Chen, I.-K. Lee, S. Leopoldseder, H. Pottmann, T. Randrup, J. Wallner: On Surface Approximation Using Developable Surfaces. to appear in: *Graphical Models and Image Processing*, 1999.

[2] J. Hoschek and D. Lasser. *Fundamentals of Computer Aided Geometric Design*. A.K.Peters, 1993.

[3] I.-K. Lee. Curve reconstruction from unorganized points. Technical Report No. 55, Institute of Geometry, Vienna University of Technology, August 1998.

[4] S. Leopoldseder, H. Pottmann: Approximation of Developable Surfaces with Cone Spline Surfaces. *Computer Aided Design* **30** (1998), 571–582.

[5] D. Levin. The approximation power of moving least-squares. *Mathematics of Computation* **67** (1998), 1517–1531.

[6] D. Levin. Mesh-independent surface interpolation. Private communication.

[7] H. Pottmann, H.-Y. Chen and I.-K. Lee. Approximation by profile surfaces. In: A. Ball et al. (Eds.): *The Mathematics of Surfaces VIII*, Information Geometers, 1998., pp. 17–36.

[8] H. Pottmann, I.-K. Lee, and T. Randrup. Reconstruction of kinematic surface from scattered data. In: *Proceedings of Symposium on Geodesy for Geotechnical and Structural Engineering*, Eisenstadt, Austria, Feb. 1998, pp. 483–488.

[9] H. Pottmann and T. Randrup. Rotational and helical surface approximation for reverse engineering. *Computing* **60** (1998), 307–322.

[10] H. Pottmann and J. Wallner. Approximation Algorithms for Developable Surfaces. to appear in *Computer Aided Geometric Design*, 1999.

[11] T. Randrup. Approximation by cylinder surfaces. *Computer-Aided Design* **30** (1998), 807–812.

[12] N. Sapidis and P. Besl. Direct construction of polynomial surfaces from dense range images through region growing. *ACM Transactions on Graphics* **14** (1995), 171–200.

[13] Varady T., R. R. Martin, and J. Cox. Reverse engineering of geometric models — an introduction. *Computer-Aided Design* **29** (1997), 255–268.