Curve and surface reconstruction based on the method of evolution

P. Surynková

Charles University in Prague, Faculty of Mathematics and Physics, Department of Mathematics Education, Prague, Czech Republic.

Abstract. This contribution addresses a problem of surface reconstruction from the point clouds which is applied in many scientific and engineering applications. Our aim is to obtain a visualization of data from 3D scanning and document the real objects digitally. We describe the digital reconstruction problem and the methods of surface reconstruction. We mention several known algorithms for surface reconstruction and focus on new methods of reconstruction based on the sequential evolution. We explain a general framework for the evolution-based approximation of a given set of points by a curve. Then we apply it to surfaces. We show the sequential evolution of curves and surfaces on some concrete examples. For the implementation of the algorithms we traditionally use interactive environment MATLAB.

Introduction

In our research we deal with the digital reconstruction problem. We explore the steps which are necessary to convert a physical model or some real object into a computer model. Of course, good geometric and a solid understanding of the procedure are essential to get excellent result. Digital documentation brings the possibility to manipulate with the objects in mathematical and modelling computer software. The digital documentation of real objects is important in many branches. For example, in the architectural engineering it can help with reconstruction and documentation of historical buildings and sculptures with 3D scanners or restoring of monuments. The digital reconstruction is applied in many scientific and engineering applications.

The input is a finite set of points in the three-dimensional space, we know 3D coordinates. The input set is called point cloud in computer graphics. Three-dimensional scanners are used to produce measurement data from a real three-dimensional object. At first, before scanning, we have to think about what we consider as relevant data to be captured. There are many aspects of an object – its surface geometry, its appearance, its materiality and its geometric features. For example, a sculptural object might be best captured with a scattered set of key edges in space rather than with undifferentiated point cloud from a three-dimensional scanner. Figure 1 shows an example of digital reconstruction. You can see the input set of points and the final computer model.

Figure 1. Example of the input set of points obtained by 3D scanning of a sculptural object and the final computer model. (the source: http://www.cs.tau.ac.il/~dcor/)

The measurement data can consist of a large number of points. Real data may contain over million points. Ideally, these data points are precise coordinates of points on the surface of the object. But in real applications there will be measurement errors, we have to deal with. Only the regions of the surface of the real object directly visible from one position of the scanner will be captured. So that a single scan usually contains only measurement data for a part of the real object. We have to produce a number of scans from various positions of the three-dimensional scanner. This number can go into hundreds if great detail is desired. It depends on the type of the surface which is scanning. Each scan produces a point cloud in different coordinate system. All of these obtained point clouds have to be merged into a single point cloud represented in the same coordinate system. This procedure is called *registration*. In the merged point cloud there may be redundant data, some points are useless, don't contain any new or important information or some points are very close to one another. For that reason these redundant data points will be removed. There exist several removal criteria which depend on the underlying application, more detailed information are in [*Iske,* 2007].

In the subsequent *polygon phase*, a triangle mesh is computed that approximates the given data points. This procedure is very difficult. It doesn't exist any general solving method. In the polygon phase we obtain a first surface representation of the object. Several known algorithms for computing triangle mesh are for example alpha-shapes, crust algorithm, cocone algorithm which are based on spatial subdivision (on the dividing of the three-dimensional space). It means that the circumscribed box of the input set of points is divided into disjoint cells – e.g. tetrahedrization, we obtain a tetrahedral mesh. Then we find those parts of mesh which are connected with the surface [*Mencl et al. 1997, Edelsbrunner et al., 2001*].

The final *shape phase* isn't necessary for pure visualization but it will be crucial for architecture. We have to convert the triangle mesh into a CAD representation of an object that is appropriate for further processing. This phase includes edge and feature line detection and decomposition into parts of different nature and geometry – for example planar parts, cylindrical patches, conical patches, freeform patches. This process is called *segmentation*. Then we have to approximate the data regions using surfaces of the correct type which we identified in the segmentation. For example, a region identified as being planar in the segmentation phase will be approximated by part of a plane. Computing such an approximation plane is simple task. This process is known as *surface fitting*. Figure 2 shows an example of a triangle mesh and the final CAD model. More detailed information can be found in [*Pottman et al.,* 2007].

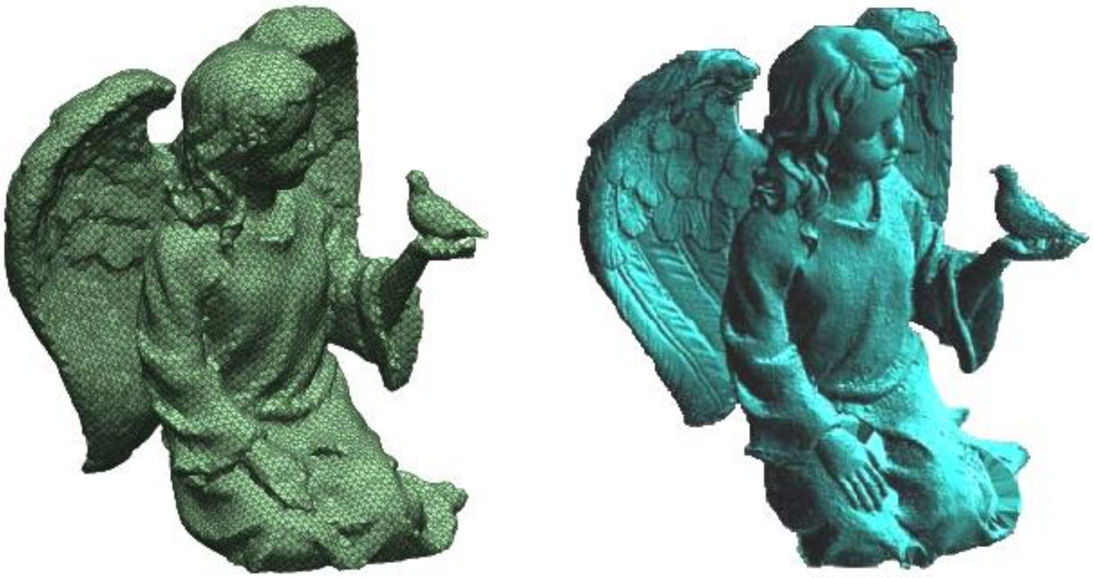

Figure 2. On the left an example of a triangle mesh, on the right the final CAD model. (the source: https://buffy.eecs.berkeley.edu/)

Methods of reconstruction based on the sequential evolution

In this part we introduce some new methods of surface reconstruction. Our aim is using methods which are based on the sequential evolution. Both - curve and surface evolution is available. We explain this problem for curves then we apply it to surfaces.

Curve evolution

The principle of curve evolution is sequential modifying of planar parametric curve from some initial position and shape. The evolution will be stopped if some condition is satisfied. In our case if the final curve has minimal distance from the given data.

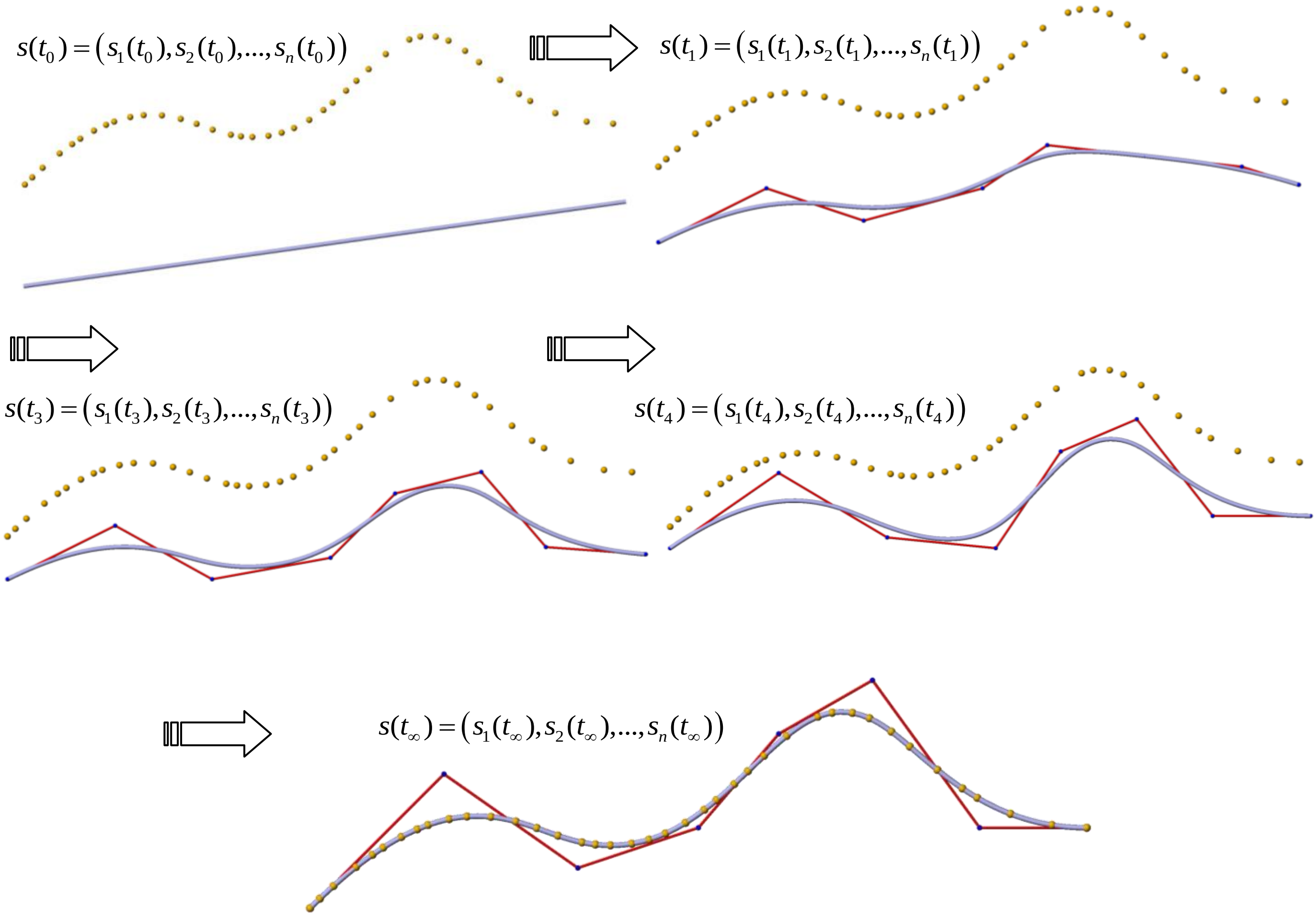

Figure 3. Curve evolution in time. First picture shows initial position and shape of the curve and the input set of points, last picture shows the final curve.

We identify a curve that approximates a given set of data points $\{p_j\}_{j=1..N}$ in the least square sense. We consider a planar parametric curve

$$c(u)=\sum_{i=0}^{m}\beta_i(u)\cdot V_i\ ,$$

where u is the curve parameter, V_i are control points and β_i are basis functions. We are looking for the curve such that

$$\sum_{j=1}^{N}\min_{x_j\in c}\left\|p_j-x_j\right\|^2\rightarrow\min\ ,$$

where x_j are points on the curve. If we denote $c_s(u):=c(s,u)$ then two different kinds of parameters appear in the representation of the curve; the curve parameter u and a vector of shape parameters $s=(s_1,s_2,...,s_n)$ where s_j denotes the coordinates of control points. It means that we are looking for the vector of shape parameters that defines the curve. We let the shape parameters s depend smoothly on an evolution parameter t, $s(t) = (s_1(t), s_2(t), ..., s_n(t))$. The parameter t can be identified with the time. Starting with certain initial values, these parameters are modified continuously in time such that a given initial curve moves closer to the data points. Figure 3 shows curve evolution in time.

For each point $\{p_j\}_{j=1..N}$ we compute the closest points $f_j = c(u_j)$ on the initial curve. During the evolution of a curve $c_{s(t)}(u)$ each point f_j travels with velocity

$$v_s(u_j) = \dot{c}_s(u_j) = \sum_{i=1}^{n} \frac{\partial c_s(u_j)}{\partial s_i} \dot{s}_i$$

or with normal velocity

$$(v_s(u_j))^{\mathsf{T}} n_s(u_j) = \sum_{i=1}^{n} \left(\frac{\partial c_s(u_j)}{\partial s_i} \dot{s}_i \right)^{\mathsf{T}} n_s(u_j) .$$

The dot denotes the derivative with respect to the time variable t, $n_s(u_j)$ denotes the unit normal of the curve in the point $c_s(u_j)$. We denote $d_j := p_j - f_j$. If a closest point is one of the two boundary points then we consider first velocity. We can compute for each point f_j the velocity or normal velocity on the one hand and the expected velocity on the other hand. The following condition has to satisfy

$$\sum_{\substack{j=1 \\ u_j \notin \{a,b\}}}^{N} \left\| (v_s(u_j) - d_j)^{\mathsf{T}} n_s(u_j) \right\|^2 + \sum_{\substack{j=1 \\ u_j \in \{a,b\}}}^{N} (v_s(u_j) - d_j)^2 \to \min_{\dot{s}} .$$

We compute $(\dot{s}_1, \dot{s}_2, ..., \dot{s}_n)$ and update the vector of shape parameters using the Euler-steps $(s_1 + \varepsilon \dot{s}_1, s_2 + \varepsilon \dot{s}_2, ..., s_n + \varepsilon \dot{s}_n)$. More detailed information can be found in [*Aigner et al.*, 2006].

Surface evolution

We can apply the methods of curve evolution to surfaces. The principle of surface evolution is the same. We assume that a set of data points $\{p_j\}_{j=1..N}$ in the space is given, see Figure 4. We identify a surface that approximates a given set of data points in the least square sense. Surface evolution in time is shown in Figure 5.

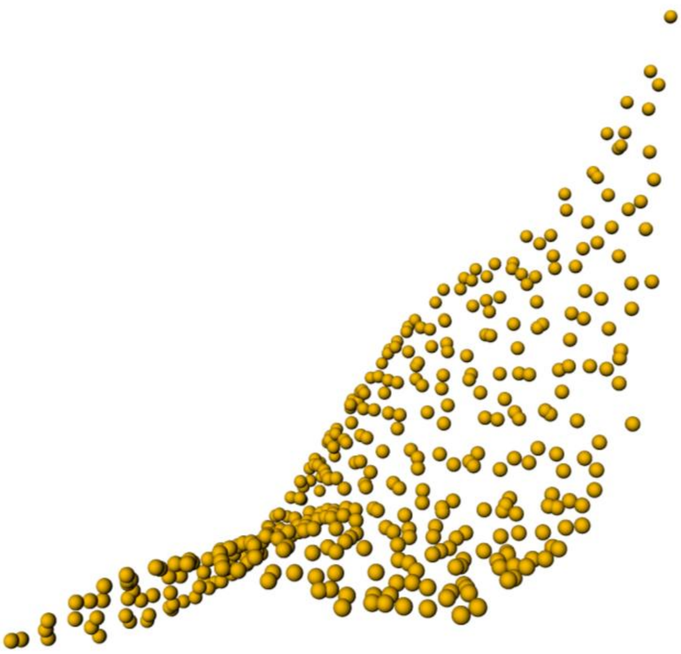

Figure 4. The input set of data points in the space.

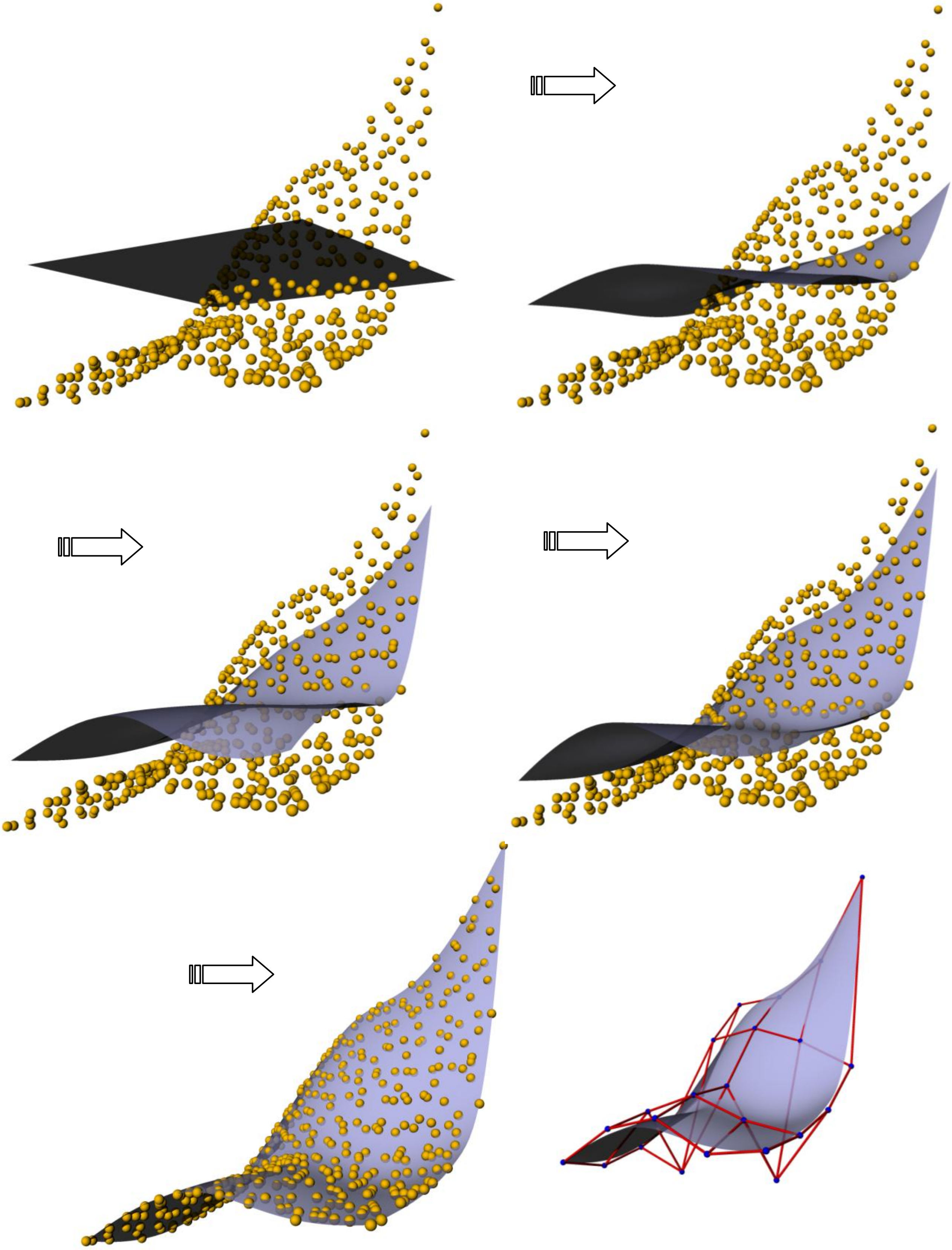

Figure 5. Surface evolution in time. First picture shows initial position and shape of the surface and the input set of points, last two pictures show the final surface and the final surface with control points which are modified in time.

Choice of software

We choose MATLAB for implementation of the algorithms. MATLAB is modern language and interactive environment for technical computing. It includes many 2D and 3D graphics functions for data visualization and it is commonly used for these purposes. We also use RHINOCEROS (NURBS modeling for Windows) for 3D modelling. Figures 3, 4, 5 are created using RHINOCEROS.

Conclusion and future work

In our contribution we described the digital reconstruction problem and the process of surface reconstruction. We introduced new methods of reconstruction based on the sequential evolution. Curve evolution was explained and an example of surface evolution was shown.

In future work we will focus on further methods for surface reconstruction. We want to approximate surface of building practice (for example vaults) using quadratic patches. We plan to work with detection and decomposition into parts of different nature and geometry using the curvature of surfaces.

References

Aigner M., Šír Z., Jüttler B., Least-squares approximation by Pythagorean hodograph spline curves via an evolution process, GMP, Springer LNCS, Vol. 4077, 2006.

Edelsbrunner, H., Geometry and topology for mesh generation, Cambridge University Press, Cambridge, United Kingdom, 2001.

Iske, A., Multiresolution Method in Scattered Data Modelling, Technische Universität München, Germany, 2004.

Mencl, R., et al., Interpolation and Approximation of Surfaces from Three-Dimensional Scattered Data Points, State of the Art Report, EURORAPHICS'98, 1997.

Pottman H., Asperl A., Hofer M., Kilian A., Architectural geometry, Bentley Institute Press, USA, 2007.