Guided Multi-Dimensional Reconstruction from Cross-Sections

Daniel Cohen-Or and David Levin

Abstract. Given parallel $(n-1)$-dimensional cross-sections of an n-dimensional body, one would like to reconstruct the body. The method based upon distance field interpolation gives a robust solution to this problem in its ability to deal with any topology in any dimension. Still this method may give undesired solutions to the problem if the changes from one cross-section to the next are big relative to the size of the details in the cross-sections. In the present work we suggest a way to improve the distance field interpolation method. For this we first assume that the data to the problem are the cross-sections and some specified points on them, prescribing geometric links between the different cross-sections. These links may be manually user-defined, or automatically defined by matching features in the cross-sections. The distance field interpolation method is modified so that the interpolation is guided by those links. We describe some rules for defining a smooth least-distorting mapping which realizes the desired links, and present the corresponding guided interpolation.

§1. Introduction

The problem of reconstructing an n-dimensional body from its cross-sections appears in many applications, from biomedical imaging to 2D and 3D animation. In 3D animation the problem can also be viewed as a surface blending problem. Namely, the continuous evolution of a surface from a source surface, through intermediate surfaces, into a target surface. This is an object-space problem of generating intermediate 3D models [5], to be distinguished from image-space transformations [10].

This paper describes a method that allows the user to create a series of models which form a smooth transition of a given set of general keyframe models. The method deals explicitly with surfaces and its principles are naturally extended from 2D space (contours) to 3D (surfaces) and to hyper-surfaces in higher dimensions.

Advanced Topics in Multivariate Approximation
F. Fontanella, K. Jetter and P.-J. Laurent (eds.) pp. 1–9.

ISBN xxx.

The problem of blending two surfaces, even for polyhedral ones, is not simple. In order to blend between two polyhedral models, a correspondence between their structure (vertices) has to be established [5]. However, a correspondence alone does not guarantee a smooth transition from the source model to the target one, and the *vertices' paths* have to avoid unpleasant situations such as self-intersections. 2D shape blending techniques that deal with the vertices' path problem are not easy to extend to 3D [7,9].

The problem of shape blending can also be considered as a problem of body reconstruction from cross-sections [6,8]. Some of the reconstruction methods use bivariate interpolation by finding a correspondence between the contours of two cross-sections. Other methods use a univariate interpolation between values of corresponding 3D grid points [4]. When the cross-sections are not dense, they may exhibit significant changes in the topology of the boundary contours. A method that handles easily that issue is the Distance Field Interpolation (DFI) presented and analyzed in [6]. The DFI is a general set-valued interpolation method for the reconstruction of an n-dimensional model from a sequence of its $(n-1)$-dimensional cross-sections. It uses a univariate interpolation of the distance field, and it performs well for most general topology in any dimension. The method is also defined in a discrete space and generates intermediate discrete contours of arbitrary complexity. The DFI method has been adapted successfully in biomedical reconstruction applications [4]. Yet, the DFI method is not performing well if the original body, from which the cross-sections are taken, is too 'twisted'. That is, if there are significant differences between adjacent cross-sections. For example, if two adjacent cross-sections are not aligned due to rotation or translation.

The reconstruction procedure presented in this paper is based on the DFI method. However, it is extended in the sense that the interpolation, and thus the blended surface, follow a warp transformation guided by means of a user-defined control. The control is defined by a point-to-point correspondence between prescribed anchor points on the given intermediate cross-sections. In order to achieve least-distorted in-between shapes, we decomposed the warp transformation W into a *rigid* (rotational) transformation R and an *elastic* transformation E, which are separately interpolated. The rigid-elastic decomposition of the warp function, and its particular interpolation are so chosen to minimize the distortion of the intermediate surfaces. The rigid part is used to rotate and translate the source object to a matching general position of the target object, while the finer features of the object are slowly evolved by the elastic part.

In Section 2 we review the DFI method for multi-dimensional reconstruction from cross-sections. In Section 3 we introduce the problem of reconstruction guided by anchor points and consider the application of a warp transformation. The warp transformation, its decomposition and parameterization, and the new warp-guided DFI method are constructed. Implementation details and numerical experiments are described in [2].

§2. Distance Field Interpolation for n-Dimensional Reconstruction

Reconstruction of 3D objects from cross-sections is of major importance in many applications, particularly in biomedical imaging. One class of reconstruction methods builds a polygonal surface by assuming the cross-sections consist of closed polygonal contours [1]. Another class of methods uses a volumetric approach. The entire volume (3D voxel array) is reconstructed from the gray levels of the cross-sections. The missing data is reconstructed by some interpolation of the gray values of the cross-sections. The surface is then defined and reconstructed by iso-surfacing methods. These methods do suffer from discontinuities in the reconstructed surface, and thus, do not yield smooth blending transformations.

The method of Distance Field Interpolation (DFI) presented in [6] better reconstructs the surface by interpolating distance values rather than the gray values. This method has been adapted successfully in biomedical reconstruction applications [4].

The DFI method can be described in terms of n-dimensional surface reconstruction from $(n-1)$-dimensional cross-sections $(n > 1)$. Consider an object $\Omega \subset I\!R^n$, discretized or not.

Given a finite set of $(n-1)$-dimensional cross-sections of the object,

$$\Omega_{t_j} = \{x = (x^1, \ldots, x^{n-1}) \mid (x^1, \ldots, x^{n-1}, t_j) \in \Omega\} \ , \quad t_0 < t_1 \ldots < t_M \ , \qquad (1)$$

we would like to reconstruct Ω. Throughout the paper we use lower indices for numbering entities and upper indices to denote the components of a vector. Let us define the distance fields at the levels $t_0, t_1 \ldots t_M$: For $x = (x^1 \ldots x^{n-1})$

$$D_{t_j}(x) = \begin{cases} -dist(x, \partial\Omega_{t_j}) & \text{if } x \in \Omega_{t_j} \\ dist(x, \partial\Omega_{t_j}) & \text{otherwise} \ , \end{cases} \qquad (2)$$

where $\partial\Omega_{t_j}$ denotes the boundary of Ω_{t_j}, and *dist* denotes the Euclidean distance in $I\!R^{n-1}$. Now, using univariate interpolation (with respect to the parameter t) we interpolate, between the cross-sections, the distance values of points having the same first $n-1$ coordinates. The resulting interpolant approximates the $n-1$ Euclidean distance between $y = (x^1, \ldots, x^{n-1}, t)$ and $\partial\Omega_t = \partial\Omega \mid_{y^n = t}$.

Once this approximated distance field is available, the surface of the object can be determined by the zero points of the distance field, or, in the discretized version, as the boundary between the positive and negative valued lattice points, while the volume itself (its *interior*) is defined as the set of all negative valued points.

Formally, the DFI method defines an approximate domain $\widetilde{\Omega} \approx \Omega$ as follows: For $y = (x^1, \ldots, x^{n-1}, t)$ we first define the interpolant $d_x(t)$ $(x = (x^1 \ldots x^{n-1}))$ by univariate interpolation of the values $\{D_{t_j}(x)\}_{j=0}^{M}$. Now,

$$\widetilde{\Omega} = \{y \ \mid \ d_x(t) \leq 0 \ \} \ . \qquad (3)$$

Evidently, this method can be adapted for blending, where two or more cross-sections are given, to generate the intermediate models. For more details on the choice of the interpolation method, and for approximation rate analysis see [6]. The DFI method suffers from two major deficiencies: The first one is the low approximation order near boundary points of Ω with a tangent orthogonal to $(0,0,\ldots,1)$. To the best of our knowledge, this problem is not resolved yet. The second problem is also related to the approximation power of the method. Here we refer to the case where the ratio between the distances $|t_{j+1} - t_j|$ and the 'width' of the cross-sections Ω_{t_j} is large with respect to the slope of the boundary of Ω in the direction $(0,0,\ldots,1)$. The approach taken in this work to overcome this problem is the combination of the DFI with a proper point-to-point warp transformation.

§3. Guided DFI blending

Let us consider for simplicity the blending of two objects, $S = \Omega_0$ and $T = \Omega_1$. The DFI gives us a way for blending by interpolating (linearly) between the distance functions D_0 and D_1, as described in Section (3). We call this blending process the no-warp DFI blending. No-warp DFI blending is quite restricted, and it may produce non-satisfactory results. To demonstrate this, consider the case when T is just a rotation of S, and S is just a thin rod. The no-warp DFI blending gives at $t = \frac{1}{2}$ either a very small object, or an empty object, which is not the naturally-expected result. On the other hand, the no-warp DFI blending works nicely if S is a thin rod and T is just the rod S missing its middle third.

Consider now the possibility that T is obtained by a transformation (warp) of S, i.e., $T = W_1(S)$. Further, let us assume that T is gradually evolving from S, through a continuum of objects $W_t(S)$, where $\{W_t\}_{t\in[0,1]}$ are smooth transformations, smoothly changing with t, where $W_0 \equiv I$ is the identity transformation. In such a case the blending can be the warp itself, this is the pure-warp blending. Pure-warp blending is also very restricted, for example, it does not allow the possibility of changes in the genus of the objects.

We are going to hybridize the two different methods into a more powerful tool. The first step of the hybrid method is finding a smooth warp $\{W_t\}_{t\in[0,1]}$ such that $W_1(S) \approx T$ in some chosen sense. Then the DFI method is applied, now guided by the warp $\{W_t\}$. Note that the warp is operating in $\mathbb{R}^{n-1}$ and the interpolation of the distance field is performed along the $n-th$ dimension. At this point we go back to the case of $M+1$ given objects, considered as cross-sections of Ω, as in Eq. (1). First we find a smooth warp $\{W_t\}_{t\in[0,1]}$ such that

$$W_{t_j}(\Omega_0) \approx \Omega_{t_j} \quad , \; j = 0,\ldots,M \; . \tag{4}$$

The signed distance function at each level are defined here also by Eq. (2). For $x = (x^1,\ldots,x^{n-1})$ we then interpolate the values $\{D_{t_j}(W_{t_j}(x))\}_{j=0}^{M}$, denoting the interpolant $d_x(t)$. The approximated domain $\widetilde{\Omega}$ defined by the hybrid warped DFI

procedure is

$$\widetilde{\Omega} = \{ y = (W_t(x), t) \mid d_x(t) \leq 0 \} . \tag{5}$$

Obtaining the approximated warp W_t.

The way we choose to define the approximated warp W_t is derived from several applications. On each level t_j we assume we are given N points $\{p_{j,i}\}_{i=1}^{N}$ whose mapping is predetermined. Later we refer to these points as anchor points. The warp is to be determined so that

$$W_{t_j}(p_{0,i}) = p_{j,i} , \quad 1 \leq i \leq N . \tag{6}$$

The warp function maps the source (level zero) anchor points to their corresponding target points at levels $\{t_j\}_{j=0}^{M}$, while other points in the source domain comply to those constraints. A warp defined by anchor points is attractive because it is applicable in any dimension in a most natural way. ¿From the user point of view, this method is intuitive and very easy to handle. It also has the advantage of being robust and very stable, in the sense that the results are not very sensitive to the small perturbations in the anchor points.

We now describe the definition of the warp transformation based on anchor points, starting with the case $n = 3$, i.e., the cross-sections are in $\mathbb{R}^2$. As stated earlier in the introduction, in order to achieve control over the shapes of the in-between objects the warp transformation W_t is decomposed into a *rigid* part R and an *elastic* part E. To explain the motivation here, let us view a very simple case: Consider the blending of two objects in $\mathbb{R}^2$, $S = \Omega_0$ and $T = \Omega_1$, where $T = R_\theta S$ and R_θ is a rotation in angle θ. There are two obvious options for defining a warp W_t. One is the *linear interpolation* warp $W_t = (1-t)I + tR_\theta$. The other is the *linear rotation* warp $W_t = R_{t\theta}$. Both warps vary smoothly from the identity I at $t = 0$ to R_θ at $t = 1$. However, the second option is the preferable one since it is an isometry for any t, i.e., non-distorting. Now consider the case where T is obtained from S by a rotation R_θ, a translation c, and an *elastic* transformation E, i.e., $T = E(R_\theta S + c)$. In the following we will choose the rotation R_θ and the translation c so that E is as close as possible to I, in some sense. The warp W_t is defined by

$$W_t(x) = ((1-t)I + tE)(R_{t\theta}x + tc) . \tag{7}$$

This approach can be applied to the case of $M+1$ objects in $\mathbb{R}^2$, $\{\Omega_{t_j}\}_{j=0}^{M}$. First we find a sequence of rotations $\{R^{(j)} = R_{\theta_j}\}_{j=0}^{M}$, a sequence of elastic transformations $\{E_j\}_{j=0}^{M}$, and translations $\{c_j\}_{j=0}^{M}$, with $\theta_0 = 0$, $E_{t_0} = I$ and c_0=0. Now we define

$$\theta(t) = \sum_{j=0}^{M} b_j(t)\theta_j , \quad E_t = \sum_{j=0}^{M} b_j(t)E_j , \quad c(t) = \sum_{j=0}^{M} b_j(t)c_j , \tag{8}$$

where $\{b_j(t)\}_{j=0}^{M}$ are Lagrange-type basis functions for interpolation at the levels $t_0, \ldots, t_M$. Finally, the warp transformation W_t is defined as

$$W_t(x) = E_t(R_{\theta(t)}x + c(t)) \ . \tag{9}$$

In the following we describe a way for obtaining the rotations, the elastic transformations and the translations, and we consider the main case $n = 4$, namely blending of 3D objects.

The Rigid Transformation in $\mathbb{R}^3$.

We start with given sets of anchor points in $\mathbb{R}^3$, $\{p_{j,i}\}_{i=1}^{N}$, corresponding to the levels $\{t_j\}$, $j = 0, \ldots, M$. The rigid part of the transformation from level t_0 to level t_j is defined by the rotation $R^{(j)}$ and the translation c_j which minimizes

$$Q = \sum_{i=1}^{N} \| R^{(j)}p_{0,i} + c_j - p_{j,i} \|^2 \ . \tag{10}$$

Imagine that the points $\{p_{0,i}\}$ are connected by identical elastic springs to the corresponding points $\{p_{j,i}\}$. The form Q represents the elastic energy of this system, and the rigid transformation which minimizes Q, brings Ω_0 to an equilibrium position in the springs' system.

The above least-square fitting problem of the two sets of anchor points in $\mathbb{R}^3$ is solved using an explicit algorithm, which involves the *singular value decomposition* (SVD) of a 3×3 matrix. As a result, we obtain a 3×3 rotation matrix characterized by three angles, $R^{(j)} = R_{\theta_j,\phi_j,\psi_j}$, and a translation vector c_j. The warp transformation in the 3D case is defined by

$$W_t(x) = E_t(R_{\theta(t),\phi(t),\psi(t)}x + c(t)) \ , \tag{11}$$

with

$$\phi(t) = \sum_{j=0}^{M} b_j(t)\phi_j \ , \quad \psi(t) = \sum_{j=0}^{M} b_j(t)\psi_j \ . \tag{12}$$

The Elastic Warp in $\mathbb{R}^3$.

Once we have computed the rigid transformation, say in $\mathbb{R}^3$, we move to calculating the elastic transformation. In accordance with Eq. (6) and Eq. (9), we look for a transformation E_j, in general a non-linear transformation, $E_j : \mathbb{R}^3 \rightarrow \mathbb{R}^3$, such that

$$E_j(R^{(j)}p_{0,i} + c_j) = p_{j,i} \ , \ 1 \leq i \leq N \ . \tag{13}$$

This is a multivariate scattered data interpolation problem, which we suggest to solve by using *Radial Basis Functions*, abbreviated RBF [3]. In $\mathbb{R}^3$ this means

solving three interpolation problems in $\mathbb{R}^3$, for each component of the vector equation defined in Eq. (13).

Radial basis functions have proven to be an effective tool in multivariate interpolation problems of scattered data: Given the scattered d-dimensional data

$$(x_i, F_i), \quad x_i \in \mathbb{R}^d, \quad F_i \in \mathbb{R} \quad , \quad i = 1, 2, \ldots, N,$$

we look for an interpolatory function $S(x)$ of the form

$$S(x) = \sum_{i=1}^{N} a_i g\left(\| x - x_i \|\right), \tag{14}$$

where $\| \cdot \|$ denotes the usual Euclidean norm on $\mathbb{R}^d$ and $g : \mathbb{R}^+ \to \mathbb{R}$. A function of this type is usually referred to as a *pure radial sum*. Using radial functions reflects the fact that the scattered data has no preferred orientation, and their choice is dimension-dependent [3]. Other multivariate approximations, generalizing the univariate splines, use augmented radial sums, where the sum in Eq. (14) is augmented by a low degree polynomial.

Our discussion is mainly concerned with the 3-dimensional case ($d = 3$). We define the elastic transformation $E : \mathbb{R}^3 \to \mathbb{R}^3$ as:

$$E(q) = L(q) + Z(q) \ , \tag{15}$$

where L is a linear transformation and Z is of the form:

$$Z(q) = (S_1(q), S_2(q), S_3(q)) \tag{16}$$

where

$$S_k(q) = \sum_{i=1}^{N} a_i^k g(\|q - q_i\|) \ , \ 1 \leq k \leq 3 \ , \ q = (q^1, q^2, q^3) \in \mathbb{R}^3 \ .$$

$g : \mathbb{R}^+ \to \mathbb{R}$ is a univariate function, $\| \cdot \|$ is the Euclidean norm on $\mathbb{R}^3$, and $\{q_i\}$ for $i = 1, 2, \ldots, N$ are the source anchor points *after* the rigid transformation has been applied to them.

The above definition of the transformation E, includes a linear part L. This linear part is added because pure radial sums often yield poor approximation of the transformation for points away from the anchor points. Moreover, the linear part is natural here since we would like to reconstruct exactly those transformations E which are linear, and especially the identity transformation.

A proper choice of g is also important. Choosing g as a function with a fast decay results in a finer local influence of the radial part, which can be used in order to obtain different effects for various parts of the object. Other natural choices of

the function g are $g(r) = r^2 \log(r)$ in $\mathbb{R}^2$ and $g(r) = r^3$ in $\mathbb{R}^3$ [3]. In the following we describe the computation of the elastic warp in $\mathbb{R}^3$.

After computing the rigid part, namely, the rotation $R^{(j)}$ and the translation c_j, the elastic mapping E_j is defined by the following N interpolation conditions:

$$E_j(R^{(j)} p_{0,i} + c_j) \equiv E_j(q_i) = p_{j,i} \equiv y_i \quad , \quad i = 1, \ldots, N \ , \tag{17}$$

where $q_i = (q_i^1, q_i^2, q_i^3)$, and $y_i = (y_i^1, y_i^2, y_i^3)$. This interpolation problem is always solvable if we use an augmented radial approximation (with a properly chosen g) of the form

$$E_j(q) = \sum_{i=1}^{N} a_i g\left(\|q - q_i\|\right) + Aq + \alpha_4 \ , \tag{18}$$

where

$$A = (\alpha_1, \alpha_2, \alpha_3)^T \ , \ a_i \in \mathbb{R}^3 \ , \ 1 \le i \le N \ , \ \alpha_\ell \in \mathbb{R}^3 \ , \ 1 \le \ell \le 4 \ .$$

Thus E_j is determined by $N + 4$ coefficients in $\mathbb{R}^3$. The computation of those coefficients involves the solution of 3 square linear systems of size $N+4$ each, where N conditions are derived by the interpolation requirements, as defined in (17), and the additional compatibility conditions are

$$\sum_{i=1}^{N} a_i^k = \sum_{i=1}^{N} a_i^k q_i^1 = \sum_{i=1}^{N} a_i^k q_i^2 = \sum_{i=1}^{N} a_i^k q_i^3 = 0 \ , \ k = 1, 2, 3 \ .$$

These conditions guarantee that the transformation is affine reducible, i.e., the transformation is affine whenever possible. The system of equations for the vectors of unknowns $u_k = (a_1^k, ..., a_N^k)^T$ and $v_k = (\alpha_1^k, \alpha_2^k, \alpha_3^k, \alpha_4^k)^T$, k=1,2,3, is

$$\begin{cases} Gu_k + Hv_k = b_k \\ H^T u_k = 0 \ . \end{cases} \tag{19}$$

where $b_k = (y_1^k, ..., y_N^k)$, $G = \{\ g(\|q_i - q_j\|)\ \}_{i,j=1}^{N}$ and H is an $N \times 4$ matrix with an i-th row $\{1, q_i^1, q_i^2, q_i^3\}$, $1 \le i \le N$.

Now that the warp transformation W_t is fully defined. The warped DFI approximation is obtained by (5), that is, the approximated domain $\widetilde{\Omega}_t$ at time t, defined by the hybrid warped DFI procedure, is defined by

$$\widetilde{\Omega}_t = \{ y = (W_t(x), t) \ \mid \ d_x(t) \le 0 \ \} \ , \tag{20}$$

with W_t given by (12) and d_x as defined in Section 2.

Practical implementation.

The formal definition (20) seems rather simple, however, it is quite involved to realize. To make the warped DFI approximation an applicable tool in computer graphics, efficient algorithms for computing distance fields and for boundary detection, in 2D and in 3D, should be developed and employed. Implementation details and numerical experiments with the warped DFI method are described in [2]. Examples of 2D and 3D animations generated by the presented method can be found in the URL address: http://www.math.tau.ac.il/~daniel/Morph/morph.html/.

References

1. G. Barequet and M. Sharir, Piecewise linear interpolation between polygonal slices, in *Proc. 10th Ann. ACM Symp. on Computational Geometry*, Stony Brook, NY, 1994, 93-102.
2. D. Cohen-Or, D.Levin and A. Solomovici, Surface metamorphosis, Technical Report 309/96, Computer Science Department, Tel-Aviv University, January 1996.
3. N. Dyn, Interpolation and Approximation by Radial and Related Functions, in *Approximation Theory VI*, C.K. Chui, L.L. Schumaker and J.D. Ward (eds.), Academic Press, 1987, 85-100.
4. G.T. Herman and J. Zheng and C.A. Bucholtz, Shape-Based Interpolation, IEEE Computer Graphics and Applications, May 1992, 69-79.
5. J.R. Kent and W.E. Carlson and R.E. Parent, Shape Transformation for Polyhedral Objects, Computer Graphics **26** (1992), 47-54.
6. D. Levin, Multidimensional Reconstruction by Set-valued Approximation, IMA J.Numerical Analysis **6** (1986), 173-184.
7. T.W. Sederberg and P. Gao and G. Wang and H. Mu, 2-D Shape Blending: An Intrinsic Solution to the Vertex Path Problem, in *Proceedings of SIGGRAPH '93*, **27** (1993), 15-18.
8. T.W. Sederberg and E. Greenwood, A Physically Based Approach to 2D Shape Blending, in *Proceedings of SIGGRAPH '92*, **26** (1992), 25-34.
9. M. Shapira and A. Rappopor, Shape Blending Using the Star-Skeleton Representation, IEEE Computer Graphics and Applications, March 1995, 44-50.
10. G. Wolberg, in *Digital Image Warping* IEEE Computer Society Press, Washington, DC, 1990.

Daniel Cohen-Or and David Levin
School of Mathematical Sciences
Tel-Aviv University
Tel-Aviv 69978
Israel

daniel@math.tau.ac.il
levin@math.tau.ac.il