An Implicit Method for Interpolating Two Digital Closed Curves on Parallel Planes *

Nikolaos Gabrielides and Laurent Cohen

Centre de Recherche en Mathématique de la Décision,
Université Paris IX, Dauphine,
Place du Maréchal de Lattre de Tassigny, 75775 Paris Cedex 16, France
ngabriel@ceremade.dauphine.fr, cohen@ceremade.dauphine.fr

Abstract. Ardon et al. [2] presented an implicit method for surface segmentation in 3D images. The boundary of the surface is assumed to be constrained by two given curves in the image. In this work we adopt the afore approach to interpolate two given digital curves lying on parallel planes, by introducing an artificial image potential, which is based on a triangular facet surface interpolation technique.

1 Introduction

Let us be given two *digital contours* Γ and Δ, i.e. two closed ordered sets of black voxels on a white background, lying on the planes $z = r_\Gamma$ and $z = r_\Delta$, of a 3D image Ω'_{pqr}, which discretizes the volume $\Omega \subset \mathbb{R}^3$, with p, q and r being the number of voxels distributed equidistantly along the x, y and z axis, respectively. We wish to construct a surface that interpolates the data sets Γ and Δ.

A similar formulation to the afore *digital contour interpolation problem* can be found in the construction of a gradual transformation from the closed polygon, $\mathcal{P}_\Gamma$ to the closed polygon $\mathcal{P}_\Delta$, most widely known as the *morphing problem*. Following Efrat et al. [11], this tranformation can be expressed as a mapping:

$$\mathcal{M}(\mathcal{P}_\Gamma, \mathcal{P}_\Delta) = \{\mu(t),\ t \in [0,1],\ \text{such that } \mu(0) = \mathcal{P}_\Gamma, \mu(1) = \mathcal{P}_\Delta\},$$

which can be computed by solving the following two problems: (a) *The correspondence problem*, where an explicit mapping between $\mathcal{P}_\Gamma$ and $\mathcal{P}_\Delta$, is established, by specifying two functions $\mathbf{c}_\gamma(u) : [0,1] \to \mathcal{P}_\Gamma$ and $\mathbf{c}_\delta(u) : [0,1] \to \mathcal{P}_\Delta$. (b) *The vertex path problem*, where we seek for the trajectory that connects $\mathbf{c}_\gamma(u)$ with $\mathbf{c}_\delta(u)$ (see also [15]). If this path is a straight line, then it is easy to find examples with self intersections. The authors of [11] assert that if one adopts the policy of moving $\mathbf{c}_\gamma(u)$ to $\mathbf{c}_\delta(u)$ along the Euclidean shortest path, from $\mathbf{c}_\gamma(u)$ to $\mathbf{c}_\delta(u)$ that avoids $\mathcal{P}_\Gamma$ and $\mathcal{P}_\Delta$, then it is guaranteed that all intermediate morphs are simple, since the shortest paths do not cross each other, although two such paths may have a common sub-path.

* This work was partially supported by ANR grant SURF -NT05-2_45825.

Hence, in order to achieve a solution to the digital contour interpolation problem, free of self intersections, we seek for a method that constructs surfaces from 3D images, that contain *geodesic paths* connecting the digital contours Γ and Δ. The method presented in [2] might give us the opportunity to solve the problem with implicitly defined surfaces, as it possesses this property.

2 Preliminaries

In order to segment a given 2D or 3D image $I : \Omega \rightarrow \mathrm{I\!R}$, a common approach is to define a Riemannian manifold, called *potential function*, $P = P(I) : \Omega \rightarrow \mathrm{I\!R}$, such that features in I will be captured on P. This, of course, is ensured with an "appropriate" definition of the function P, which takes into account the nature of the features we aim to follow.

More specifically, after the classic work of Kass et al. [16] in 2D image segmentation methods the objective is to compute an *active contour*, $\mathbf{C}(s),\ s \in [0, L]$, located on the surface P, such that minimizes the energy functional:

$$E(\mathbf{C}) = \int_0^L P(\mathbf{C}(s))ds. \tag{1}$$

Towards this aim, Cohen & Kimmel, in [8], presented a segmentation method, which computes the active contour connecting two given points, $\mathbf{P}_1$, $\mathbf{P}_2$ on P. The authors show that a globally minimal curve for (1) is obtained by following the opposite gradient direction on the *minimal action map* $U_{\mathbf{P}_1}(\mathbf{Q})$ (see [18]) which is defined by:

$$U_{\mathbf{P}_1}(\mathbf{Q}) = \inf_{\mathbf{C}(0)=\mathbf{P}_1, \mathbf{C}(L)=\mathbf{Q}} \int_0^L P(\mathbf{C}(s))ds, \quad \mathbf{Q} \text{ on } P. \tag{2}$$

The minimal path $\mathbf{C}(s)$, from $\mathbf{P}_1$ to $\mathbf{P}_2$ is then obtained by solving the problem:

$$\frac{d\tilde{\mathbf{C}}(\sigma)}{d\sigma} = -\nabla U_{\mathbf{P}_1}(\tilde{\mathbf{C}}(\sigma)), \text{ with } \tilde{\mathbf{C}}(0) = \mathbf{P}_2, \quad \text{and setting } \mathbf{C}(s) = \tilde{\mathbf{C}}(L - \sigma) \tag{3}$$

According to the analysis in [19] the minimal action map $U_{\mathbf{P}_1}$ is the solution of the following eikonal equation:

$$||\nabla U_{\mathbf{P}_1}|| = P, \quad \text{with } U_{\mathbf{P}_1}(\mathbf{P}_1) = 0. \tag{4}$$

An extension of the above results for 3D images is presented in [1]. Given a 3D image, I, and the corresponding potential, P, the Euler-Lagrange equations of the energy functional E in the 3D space are:

$$\nabla P(\mathbf{C}) \cdot \hat{\mathbf{n}} = P(\mathbf{C})\kappa \quad \text{and} \quad \nabla P(\mathbf{C}) \cdot \hat{\mathbf{b}} = 0, \tag{5}$$

where the vectors $\hat{\mathbf{n}}$, $\hat{\mathbf{b}}$ and the scalar κ denote the normal, the binormal and the curvature of $\mathbf{C}$, respectively. It was proved that if $U_{\mathbf{P}_1}$ is the solution of the eikonal equation (4), then every curve $\mathbf{C}(s)$ that is a solution of the ordinary differential equation (3) is also a solution of the Euler-Lagrange equations (5).

This result paved the way to define and compute the globally minimal path between a point $\mathbf{P}$ and a curve Γ on the Riemannian manifold P. The minimal action map with respect to Γ and P is defined as the function

$$U_\Gamma(\mathbf{P}) = \min_{\mathbf{C}} E(\mathbf{C}) = \min_{\mathbf{C}} \int_0^1 P(\mathbf{C}(t))\, ||\mathbf{C}'(t)||\, dt, \tag{6}$$

where $\mathbf{C}(t),\ t \in [0,1]$ is any curve from the point $\mathbf{P}$ to the curve Γ. Note that, by the definition of $\mathbf{C}$, the minimal action map $U_\Gamma(\mathbf{P})$ is equal to $U_{\mathbf{Q}}(\mathbf{P})$ for some $\mathbf{Q} \in \Gamma$. Thus, U_Γ satisfies the eikonal equation:

$$||\nabla U_\Gamma|| = P, \quad \text{with } U_\Gamma(\mathbf{Q}) = 0, \quad \forall \mathbf{Q} \in \Gamma. \tag{7}$$

Going one step beyond, let us assume that the point $\mathbf{P}$ belongs to a set Δ. Having solved (7) all the minimal paths from each point in Δ to the curve Γ, can be computed using (3). Let us denote this set of paths by $\mathcal{S}_\Gamma^\Delta$. It can now be undrestood that if the points in Δ form a curve, then the set $\mathcal{S}_\Gamma^\Delta$ consists of all the minimal paths, $\mathbf{C}_\Gamma^\Delta(s)$, between the points of the two curves Γ and Δ. Next, in [2] a function Ψ, was defined on the image domain, such that its zero level set contains all the paths in $\mathcal{S}_\Gamma^\Delta$, i.e. $\Psi(\mathbf{C}_\Gamma^\Delta(s)) = 0$. Assuming that Ψ is continuously differentiable, the following necessary condition was obtained:

$$\Psi(\mathbf{C}_\Gamma^\Delta(s)) = 0 \Longrightarrow \nabla\Psi(\mathbf{C}_\Gamma^\Delta(s)) \cdot \frac{\mathbf{C}_\Gamma^\Delta(s)}{ds} = 0 \Longrightarrow \nabla\Psi(\mathbf{P}) \cdot \nabla U_\Gamma(\mathbf{P}) = 0, \tag{8}$$

for every point $\mathbf{P} \in \mathcal{S}_\Gamma^\Delta$. Demanding that Ψ satisfies a relation similar to (8), everywhere in Ω, a sufficient condition for the minimal paths to be contained in $\Psi = 0$ is given by the following *transport equation*:

$$\nabla\Psi(\mathbf{P}) \cdot \nabla U_\Gamma(\mathbf{P}) + G(\Psi(\mathbf{P})) = 0, \quad \Psi(\mathbf{Q}) = 0, \quad \forall \mathbf{Q} \in \Delta, \tag{9}$$

where the function G is such that $G(0) = 0$ (e.g., $G(\Psi) = a\Psi(\mathbf{P})$). In fact it was proved that if Ψ satisfies (9) then for all points $\mathbf{P}$ of its zero-level set, the minimal path joining $\mathbf{P}$ with the curve Γ is contained in the zero level set of Ψ. This, in turns, proposes to solve equation (7) and then compute Ψ through (9).

Note that the equations (7) and (9) can be solved over the nodes of Ω'_{pqr}, which discretizes Ω. In view of this, the point-sets Γ and Δ form two digital contours, which in turn implies that the afore method establishes essentially an interpolation between the two given digital contours. This allows us to employ it in the digital contour interpolation problem, provided that Γ and Δ lie on the parallel planes $z = r_\Gamma$ and $z = r_\Delta$, and no potential function is given. Since the surface $\Psi = 0$ contains all the minimal paths from the digital contour Γ to the digital contour Δ, we can allegate that solving the problems (7) and (9) we obtain an interpolating surface free of self-intersections.

3 An Artificial Image Potential

The need for an artificial image potential, other than constant, can be explained as follows: if P is constant, then the induced Riemannian manifold is a hyperplane in $\mathbb{R}^4$. Thus, the minimization of the energy functional (1) leads to a set of straight lines in $\mathbb{R}^3$, which start from the point set (contour) $\varDelta$ and end on the points of the contour $\varGamma$, having the minimum length. Suppose now that the contour $\varGamma$ is translated onto the plane $z = r_\varGamma$ until one point p of it is closer to all points of the set $\varDelta$. Then, the surface that contains all minimal paths is conic with its apex at $\mathbf{P}$ and base the set $\varDelta$. In that case all the points of $\varGamma$ but $\mathbf{P}$ are not interpolated by the surface $\varPsi = 0$. Thus, the problem is to introduce an artificial potential function, by using only the given information of $\varGamma$ and $\varDelta$.

Let us suppose that we are given a matching between the two given point sets (pixel sets) $\varGamma$ and $\varDelta$. Then, we can easily define the set of minimal paths $S_\varGamma^\varDelta$ through equation (3) for any potential function, P. If P is constant, then the minimal paths are the straight lines which connect the points of the two point sets (the pixel centers) according to the preassumed matching, thus a C^0 surface containing all the minimal paths can be a triangular facet surface that interpolates $\varGamma$ and $\varDelta$. The main disadvantage with such a construction is that self-intersections cannot be avoided in general (see [14]).

However, since there are interpolation techniques which can easily construct triangular facet surfaces that interpolate the given point sets, the above remarks make us think that it is preferable to compute the potential P through the construction of such a surface, say $\mathbf{S}$. Since $\mathbf{S}$ consists of triangles, it can easily be implicitized on the grid, $\varOmega'_{pqr}$. This can be achieved, for example, by computing the euclidean distance function, D, of $\mathbf{S}$, on the grid nodes $\mathbf{P}$, i.e.

$$D(\mathbf{P}, \mathbf{S}) = \min ||\mathbf{P} - \mathbf{S}|| \,, \ \forall \mathbf{P} \in \varOmega' \tag{10}$$

Then, regardless the matching we chose between the points of $\varGamma$ and $\varDelta$, if one traverses the minimal path from a point on $\varDelta$ to some point on $\varGamma$ and the surface intersects itself, the minimal path is chosen so as to have a common sub-path after the intersection point, thus avoiding self-intersections. This suggests that the surface $\mathbf{S}$ could be the Riemannian manifold on which the minimal paths lie, i.e. the unsigned distance D can play the role of the discrete potential P at the image discretized domain $\varOmega'_{pqr}$.

3.1 Interpolating two Polygons with C^0 Triangular Facet Surfaces

Previous Work. The construction of the surface $\mathbf{S}$ can be formulated as follows:

Problem 1. Given the ordered closed planar point sets:

$\mathcal{P}_\varGamma = \{\mathbf{P}_{\varGamma,j} \in \mathbb{E}^3, \ j = 0, \ldots, n-1\}$ and $\mathcal{P}_\varDelta = \{\mathbf{P}_{\varDelta,k} \in \mathbb{E}^3, \ k = 0, \ldots, m-1\}$,

which belong to the planes $z = r_\varGamma$ and $z = r_\varDelta$, respectively, construct a C^0 surface interpolating them and consists of triangles with vertices in $\mathcal{P}_\varGamma$ and $\mathcal{P}_\varDelta$.

The total number of such triangulations is $\frac{(n+m)!}{(n-1)!(m-1)!}$. Among them, one has to compute the optimal, according to some objective function, which quantifies the quality of these triangulations. Apparently, the quality of such a surface depends mainly on the relative twist between the points of the two contours. This in turns lets us entitle the objective function as a *twist minimization criterion.*

Keppel introduced in [17] a representation of all continuous solutions, with the aid of a *toroidal graph*, i.e., a binary matrix, $\mathbb{K}_{n\times m}$, where the indices j, k of its elements are regarded as $j = mod(j, n)$ and $k = mod(k, m)$. If $\mathbb{K}_{jk} = 1$, then the points $\mathbf{P}_{\Gamma,j}$ and $\mathbf{P}_{\Delta,k}$ are connected. If $\mathbb{K}_{jk} = 1$ and $\mathbb{K}_{j+1,k} = 1$, then the points $\mathbf{P}_{\Gamma,j}$, $\mathbf{P}_{\Gamma,j+1}$ and $\mathbf{P}_{\Delta,k}$ form a triangle. (Analogously, if $\mathbb{K}_{jk} = 1$ and $\mathbb{K}_{j,k+1} = 1$, then the points $\mathbf{P}_{\Gamma,j}$, $\mathbf{P}_{\Delta,k}$ and $\mathbf{P}_{\Gamma,k+1}$ form a triangle). Each triangle arrangement is represented by a set of unitary elements in this matrix. Keppel proved that for acceptable triangulation, these elements form a monotone path in the graph. Thus, the optimum surface can be obtained by searching among all monotone paths in the toroidal graph $\mathbb{K}_{n\times m}$.

The methods for computing such paths can be divided into two categories: the exhaustive search methods (e.g, [17, 13]) that evaluate the final surface according to some global criterion, and the methods based on weighted graphs (e.g., [6, 4, 12]) according to which a weight is assigned on each graph node and then starting from the least one, the whole path is computed by choosing in each step, among the neighboring nodes, the one with minimum weight.

The methods based on weighted graphs reduce effectively the computational cost, but since they are depending on the selection of the nodal weights, may yield surfaces that do not interpolate all points in $\mathcal{P}_\Gamma$ and $\mathcal{P}_\Delta$. Our intension is to propose a nodal weight definition, which resolves such ambiguities.

Our Method. In order to introduce our method, let us further restrict ourselves to *convex contour data sets.* In [6, 12] the weight at the node $\mathbb{K}_{jk}$ of the toroidal graph, is the length $||\mathbf{P}_{\Gamma,j} - \mathbf{P}_{\Delta,k}||$. Thus, by definition, the final result depends on the relative position of the sets $\mathcal{P}_\Gamma$ and $\mathcal{P}_\Delta$. The method of [4] proposes a translation of the polygons so as their centers, $\mathbf{A}_\Gamma$ and $\mathbf{A}_\Delta$ coinside. Thus, the square of the afore defined distance for the translated polygons, with respect to the initial points is equal to $||(\mathbf{P}_{\Gamma,j} - \mathbf{A}_\Gamma) - (\mathbf{P}_{\Delta,k} - \mathbf{A}_\Delta)||^2 = ||(\mathbf{P}_{\Gamma,j} - \mathbf{A}_\Gamma)||^2 + ||(\mathbf{P}_{\Delta,k} - \mathbf{A}_\Delta)||^2 - 2(\mathbf{P}_{\Gamma,j} - \mathbf{A}_\Gamma) \cdot (\mathbf{P}_{\Delta,k} - \mathbf{A}_\Delta)$. Then, setting $-(\mathbf{P}_{\Gamma,j} - \mathbf{A}_\Gamma) \cdot (\mathbf{P}_{\Delta,k} - \mathbf{A}_\Delta)$ as nodal weight, the path is computed by choosing the minimum weight at each step. We propose as weight function the dimensionless quantity:

$$-\frac{(\mathbf{P}_{\Gamma,j} - \mathbf{A}_\Gamma) \cdot (\mathbf{P}_{\Delta,k} - \mathbf{A}_\Delta)}{||\mathbf{P}_{\Gamma,j} - \mathbf{A}_\Gamma||\,||\mathbf{P}_{\Delta,k} - \mathbf{A}_\Delta||}, \tag{11}$$

which is equal to the negative cosine of the angle formed by the vectors: $\mathbf{P}_{\Gamma,j} - \mathbf{A}_\Gamma$, $j = 0, \ldots, n-1$ and $\mathbf{P}_{\Delta,k} - \mathbf{A}_\Delta$, $k = 0, \ldots, m-1$, in $[0, \pi]$. Since the cosine is a decreasing function in $[0, \pi]$, the proposed weight can equivalently be defined as the least angle, $\phi(\theta_{\Gamma,j}, \theta_{\Delta,k})$ formed by two lines with directions given by $\mathbf{P}_{\Gamma,j} - \mathbf{A}_\Gamma$ and $\mathbf{P}_{\Delta,k} - \mathbf{A}_\Delta$, where $\theta_{\Gamma,j}$ denotes the polar angle of the point $\mathbf{P}_{\Gamma,j}$ with respect to a coordinate system whose origin is $\mathbf{A}_{\Gamma,j}$. (Analogous definition holds for $\theta_{\Delta,k}$).

We connect the point $\mathbf{P}_{\Gamma,j}$ with the point $\mathbf{P}_{\Delta,k}$ (analogously the point $\mathbf{P}_{\Delta,k}$ with $\mathbf{P}_{\Gamma,j}$), when the index k (index j) is such that solves the following problems:

$$\min_{k=0,\ldots,m-1} \phi(\theta_{\Gamma,j}, \theta_{\Delta,k}) \quad \text{and} \quad \min_{j=0,\ldots,n-1} \phi(\theta_{\Gamma,j}, \theta_{\Delta,k}). \tag{12}$$

We set the weight at every node $\mathbb{K}_{jk}$ equal to the angle $\phi(\theta_{\Gamma,j}, \theta_{\Delta,k})$. Then, $\mathbb{K}_{jk} = 1$ for all couples of points that constitute the set of solutions of the problems (12). Now, we can easily establish that the solution has the following properties (see, e.g., Fig.3.1):

i. In every row and every column of the toroidal graph there exists at least one unitary node, since $\forall j$ we have computed the corresponding index k and $\forall k$ we have computed the corresponding j.

ii. The unitary nodes of the graph are ordered monotonically. The proof is simple, if one realizes that for each particular connection between $\mathbf{P}_{\Gamma,p_1}\mathbf{P}_{\Delta,q_1}$ and $\mathbf{P}_{\Gamma,p_2}\mathbf{P}_{\Delta,q_2}$, every point $\mathbf{P}_{\Gamma,j}$ which is in between $\mathbf{P}_{\Gamma,p_1}$ and $\mathbf{P}_{\Gamma,p_2}$ must be connected with a point which is in between $\mathbf{P}_{\Delta,q_1}$ and $\mathbf{P}_{\Delta,q_2}$, since both polygons share the same orientation and are convex.

iii. Solving the problems (12) does not imply that all the nodes of the monotone path in the graph have been computed. It is possible to be left with couples (p_1, q_1) and (p_1+1, q_1+1) but none of (p_1, q_1+1) and (p_1+1, q_1).

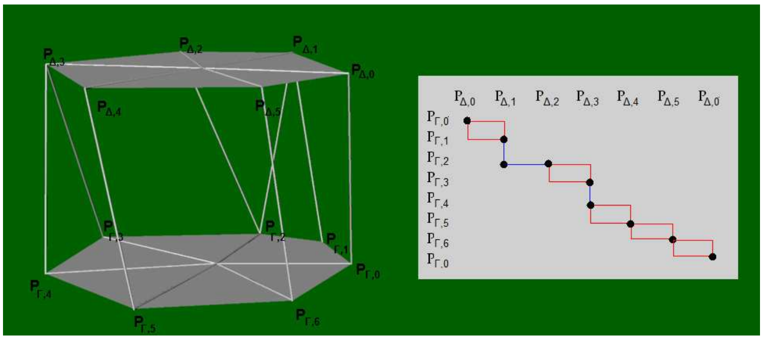

Fig. 1. Left: The connections between the points of two convex polygons, as obtained by solving the problems (12). Right: The toroidal graph of the connections. The unitary nodes are illustrated by spheres, the computed triangle edges by blue lines and the possible triangle edges by red lines.

If we interprete geometrically the afore properties, we may assert that up to this point we have constructed a surface which interpolates the point sets $\mathcal{P}_\Gamma$ and $\mathcal{P}_\Delta$ and consists of triangular and rectangular patches. The final triangulation can be obtained by tracking all the rectangular patches (i.e. where the property

(iii) holds) and triangulating them, based on the least nodal weight. Constructing the surface in this way, $O(nm)$ operations need to be performed, but this cost can effectively be reduced.

Towards this aim, we define the circular lists: $L_{\mathit{\Gamma}} = \{\theta_{\mathit{\Gamma},j}\}_{j=0}^{n-1}$ and $L_{\mathit{\Delta}} = \{\theta_{\mathit{\Delta},k}\}_{k=0}^{m-1}$ of the polar angles of the points of the two initial point sets, with respect to their centers. Note that the elements of these lists have a circularly increasing order. We find the element of the list $L_{\mathit{\Gamma}}$ with the least value and we set the head of $L_{\mathit{\Gamma}}$ at its position. Then, we compute the index ℓ which solves (12) for $j = 0$ and we set the head of $L_{\mathit{\Delta}}$ at ℓ. (We also reorder accordingly the elements of the point sets $\mathcal{P}_{\mathit{\Gamma}}$ and $\mathcal{P}_{\mathit{\Delta}}$). Now, we know that the element $\mathbb{K}_{00}$ of the graph belongs to the set of solutions of the problem. Note that up to this point, the operations done are $O(n+m)$.

Say now that the node $\mathbb{K}_{jk}$ belongs to the solution set of the problems (12), i.e. $\mathbb{K}_{jk} = 1$. We consider only the possible connection of this to the nodes $\mathbb{K}_{j+1,k}$, $\mathbb{K}_{j,k+1}$ and $\mathbb{K}_{j+1,k+1}$, knowing that due to the properties (i)-(iii), at least one of them belongs to the solution nodes. Thus, we begin from the node $\mathbb{K}_{00}$, which is already computed, and at each step we compare the weights given by the function $\phi(\cdot,\cdot)$, only for the afore mentioned three neigboring nodes. In case the least node is $\mathbb{K}_{j+1,k+1}$, we also insert in the path the one of the other two that has the least weight. Apparently, the path computed this way will traverse the nodes of the solution of the problems (12), and since the nodes to be computed are exactly $(n+m)$, it readily follows that the complexity of the algorithm is $O(n+m)$. Now, we can state the following result:

Lemma 1. *A C^0 triangular faced surface that interpolates any two convex planar polygons, with n and m points and satisfies the criteria (12) can be computed after $O(n+m)$ operations. Moreover, the space needed for the whole process is of $O(n+m)$.*

If one or both polygons are not convex, we can map them onto their convex hulls and apply the algorithm to the trasformed polygons. The output of the alogrithm is actually a point matching, thus the final surface can be constructed by adopting this matching. The use of such a technique was first proposed and implemented in [12] but their method increases the computational cost. Alternatively, in order to eliminate the cost of this mapping, we project all the points of the non-convex segments, $\mathbf{P}_j, j = S+1, \ldots, E-1$, to the corresponding convex hull segment $\mathbf{P}_S\mathbf{P}_E$, according to rule given by: $\mathbf{P}'_j = (1-t_j)\mathbf{P}_S + t_j\mathbf{P}_E, \quad t_j = \frac{\sum_{k=S}^{j-1} ||\mathbf{P}_{k+1} - \mathbf{P}_k||}{\sum_{k=S}^{E-1} ||\mathbf{P}_{k+1} - \mathbf{P}_{k-1}||}$. Computing the convex hull of a polygon by using the algorithm of [21] needs $O(n)$ operations, hence we can state that the results of the Lemma 1 still hold in the general case of non-convex polygons. It is worth to remark that this algorithm although is of linear complexity, the criterion is not local (in the sense that the same result is obtained following the exhaustive search procedure) in constrast to all up today published algorithms, except of the one given in [24] also for convex polygons. Finally, the result, i.e. the point matching, is independent of any translation of the initial data and moreover independent of an isotropic scaling of the initial data sets, thus it satisfies the criteria given by [24]. Note also that the whole method emulates the algorithmic procedure proposed by [5].

The Discrete Potential Function. Since the surface $\mathbf{S}$ consists of $(n+m)$ triangles, the minimum Euclidean distance (10) from every point of a grid Ω'_{pqr} to $\mathbf{S}$ can be found in $(n+m)$ operations, thus the total number of calculations for the discrete image potential becomes of $O(pqr(n+m))$.

4 Numerical Solution of the Eikonal and the Transport Equation

Both equations (7) and (9), belong to the class of *Hamilton-Jacobi stationary equations* and shall be considered simultaneously. The conditions under which the solution of a numerical approximation of any Hamilton-Jacobi equation converges towards the so-called *viscosity solution* can be found in [9] and [10].

In [25, 2] a first order *upwind scheme* employed in order to solve equation (9). According to them, the *numerical Hamiltonian* of (9) can be written, for $G(\Psi) = \alpha\Psi$, as

$$\left(\Psi_x^{i,j,k}, \Psi_y^{i,j,k}, \Psi_z^{i,j,k}\right) \cdot \left((U_\Gamma)_x^{i,j,k}, (U_\Gamma)_y^{i,j,k}, (U_\Gamma)_z^{i,j,k}\right) + \alpha\Psi^{i,j,k} = 0. \tag{13}$$

where the subscripts denote the partial differentation with respect to x, y and z. Approximating the derivatives by biasing the finite difference stencil in the direction where the characteristic information is coming from, lets us write the product $\Psi_x^{i,j,k}(U_\Gamma)_x^{i,j,k}$ as:

$$\Psi_x^{i,j,k}(U_\Gamma)_x^{i,j,k} = \begin{cases} \Psi_{+x}^{i,j,k}(U_\Gamma)_x^{i,j,k} = -(U_\Gamma)_x^{i,j,k}\frac{\Psi^{i,j,k}-\Psi^{i+1,j,k}}{\Delta x}, \text{ if } (U_\Gamma)_x^{i,j,k} < 0 \\ \Psi_{-x}^{i,j,k}(U_\Gamma)_x^{i,j,k} = \quad (U_\Gamma)_x^{i,j,k}\frac{\Psi^{i,j,k}-\Psi^{i-1,j,k}}{\Delta x}, \text{ if } (U_\Gamma)_x^{i,j,k} > 0 \end{cases}$$

or

$$\Psi_x^{i,j,k}(U_\Gamma)_x^{i,j,k} = |(U_\Gamma)_x^{i,j,k}|\frac{\Psi^{i,j,k}-\Psi^{I,j,k}}{\Delta x}, \text{ where } I = \begin{cases} i+1, \text{ if } (U_\Gamma)_x^{i,j,k} < 0 \\ i-1, \text{ if } (U_\Gamma)_x^{i,j,k} > 0 \end{cases} \tag{14}$$

Applying the above to (13) and solving it with respect to $\Psi_x^{i,j,k}$ we obtain:

$$\Psi^{i,j,k} = \frac{\left[\Psi^{I,j,k}\frac{|(U_\Gamma)_x^{i,j,k}|}{\Delta x} + \Psi^{i,J,k}\frac{|(U_\Gamma)_y^{i,j,k}|}{\Delta y} + \Psi^{i,j,K}\frac{|(U_\Gamma)_z^{i,j,k}|}{\Delta z}\right]}{\left[\frac{|(U_\Gamma)_x^{i,j,k}|}{\Delta x} + \frac{|(U_\Gamma)_y^{i,j,k}|}{\Delta y} + \frac{|(U_\Gamma)_z^{i,j,k}|}{\Delta z} + \alpha\right]} \tag{15}$$

for $i = 0, \ldots, p-1$, $j = 0, \ldots, q-1$ and $k = 0, \ldots, r-1$, with I, J and K being defined in analogous to (14) manner, according to the sign of the nodal derivatives of $U_\Gamma^{i,j,k}$ with respect to x, y and z, respectively.

For the eikonal equation (7) the scheme proposed by Rouy & Tourin [22],

$$\sum_{X=\{x,y,z\}} \max\left(\max\left((U_\Gamma)_{-X}^{i,j,k}, 0\right), -\min\left((U_\Gamma)_{+X}^{i,j,k}, 0\right)\right)^2 = (P^{i,j,k})^2, \tag{16}$$

leads to a quadratic equation, with respect to $(U_\Gamma)^{i,j,k}$. Both equations (15) and (16) can be solved iteratively by updating their grid values until they converge, according to some predefined accuracy.

An ultimately efficient approach to solving them is based on the so-called *fast marching method*, which was introduced by Sethian [23] for the eikonal equation (16). Realizing that the solution of the eikonal equation represents the *distance map* on the (hyper)-surface P from the boundary curve Γ (see [19] and [7]) it is to be expected that the information propagates from the smaller values, near the boundary Γ, to the larger ones as we move far from it. In other words, since the characteristics of the eikonal equation are straight lines (see [20]) emanating from the boundary Γ, *the numerical solution can be built "outwards" from the smallest values*, as Sethian pointed out. The idea is to sweep the front ahead, by considering a set of points in a *narrow band* around the existing front, and to march this narrow band forward, freezing the values of existing points and bringing new ones into the narrow band structure. The key is in the selection of which grid point in the narrow band to update. The answer is that the point having the smallest value (i.e. the closest to the already calculated points) in this narrow band around the front is the one that cannot be affected by the other points next to it, thus its value must be correct.

Returing back to the discrete transport equation (15) an extremely fast convergence can be achieved by visiting the points in the order they are reached by the *characteristic curves*, in an analogous way to that of the fast marching method for the eikonal equation (see [25, 2]). Considering the characteristics of the equation (9) we obtain that the absolute values of $\Psi^{i,j,k}$ decrease, as we move from the boundary to the zero-level set of Ψ, provided that the coefficient α is greater than zero, thus in each step we update the values of $\Psi^{i,j,k}$ on a *narrow band* of nodes, using the values of $\Psi^{i,j,k}$ that have already been calculated (*solved*), starting from the boundary of the domain, via equation (15). Then, we consider as solved the point, whose value is *closest to solved points*, i.e., the one with the *maximum absolute value* in the narrow band.

Regarding the *boundary conditions*, since we concern only for the zero level set of Ψ, and the condition $\Psi = 0$ on Γ, following [2] we define the closed set:

$$\mathcal{V}_\eta = \{\mathbf{P} \in \Omega'_{pqr_\Gamma} : \ D(\mathbf{P}, \Gamma) \leq \eta\},$$

where η is a real positive value. We impose Ψ to be equal to the signed distance between $\mathbf{P}$ and Γ on the nodes of $\mathcal{V} \cap \Omega'_{pqr_\Gamma}$ and equal to $\pm\min(D(\mathbf{P}, \Gamma))$, $\mathbf{P} \in \Omega'_{pqr_\Gamma}$ on the rest of the boundary nodes of Ω'_{pqr}, by choosing the negative sign for the nodes exterior to Γ and the positive sign for those interior to Γ. Note also that Γ can be on the boundary of Ω, while Δ must be entirely inside Ω. Numerical experimentation has shown that visually acceptable results can be achieved if we extend the grid Ω'_{pqr} in the z direction, so as Δ lies in the middle z-plane. Finally, we should remark that the algorithm yields a different solution, if we compute the surface containing the minimal paths $\mathcal{S}_\Delta^\Gamma$, instead of the one containing $\mathcal{S}_\Gamma^\Delta$. The authors of [3] raise this asymmetry by exploiting both the minimal action maps U_Γ and U_Δ, which is defined analously to U_Γ.

5 Examples

In what follows we have taken the digital contours Γ and Δ to lie on the planes $z = r_\Gamma$ and $z = r_\Delta$, with $r_\Gamma < r_\Delta$ and the coefficient α in equation (15) to be equal to 0.1. The grids are relatively coarse, ranging from $50 \div 70$ nodes in the x and y directions and 20 nodes in the z direction.

The first example (see Fig.5) can be characterized as a "simple case" where the triangular facet surface has no self-intersections. In the example shown in Figs.5-5 the triangular surface has a widely spread self-intersection region, due to the interpolated contours, which are far from being convex. The method yields a surface with no self-intersections. The third example (see Fig.5) is an interpolation of two contours of U as S like shapes. It shows that "morphing" cannot be achieved always due to the fact that in some cases the resulting surface, although it has no self-intersections, appears to have holes, i.e., disconnected cross sections in the area of self-intersection of the triangular surface. This means that the particular image potential function dictates the minimal paths to go around the self-intersection area, thus generating a hole in the surface.

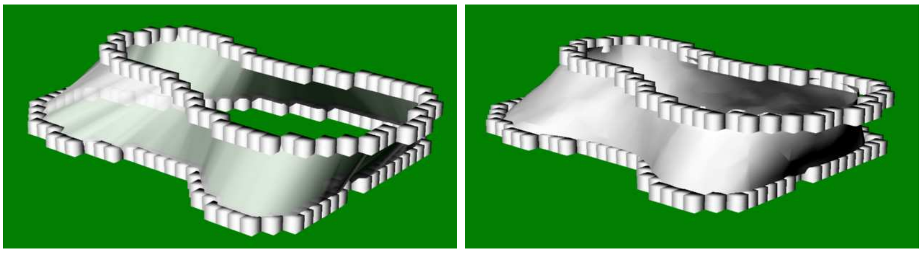

Fig. 2. Ex. 1: The C^0 triangular facet surface and the implicit surface $\Psi = 0$.

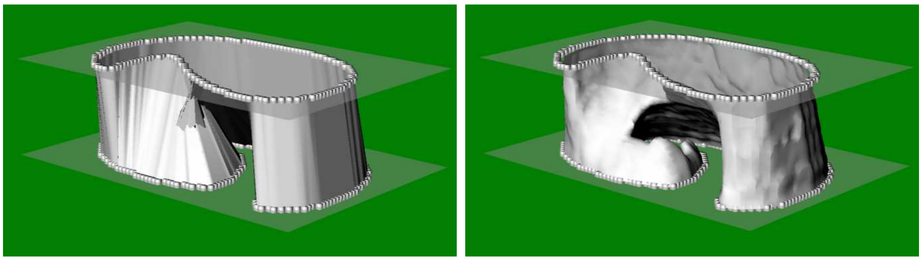

Fig. 3. Ex. 2: The C^0 triangular facet surface and the implicit surface $\Psi = 0$.

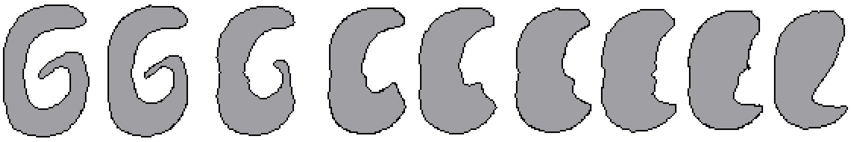

Fig. 4. Ex. 2: Intermediate slices of the implicit surface, from the contour Γ to Δ.

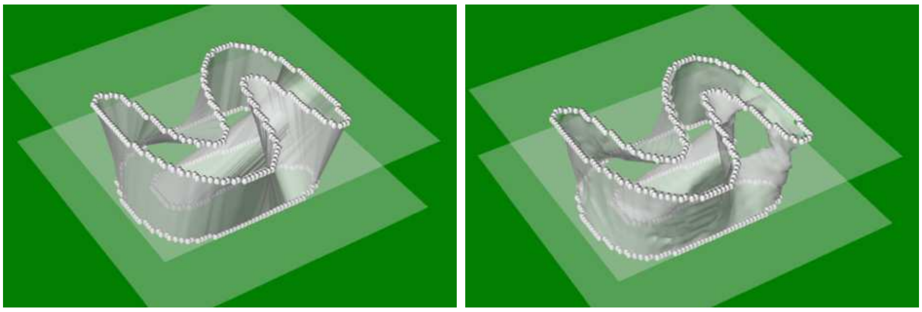

Fig. 5. Ex. 3: The C^0 triangular facet surface and the implicit surface $\Psi = 0$.

6 Conclusions

We presented an implicit method to interpolate two digital contours on parallel planes, employing the 3D image segmentation technique of [2]. In order to guarantee that the voxels of both contours will always be interpolated, we introduced an artificial potential function. Towards this, we developed an interpolation method, that matches all the pixel centers through a C^0 triangular facet surface, and set the potential function to be the eucledean distance to this surface. The method results to non self-intersecting surfaces. However, when the polygons, connecting the contour voxel centers, are far from convex, it cannot always produce morphs that preserve the connectedness of the given contours along each intermediate slice, which in turns arises a question on how can the potetial function be improved, so as to stably accomplish an acceptable morphing between Γ and Δ. This remains an open question.

The idea behind this work was to set up processes for interpolating sets of pixels/voxels following minimal paths on some appropriately defined manifolds. The idea seems to be fruitful and it might pave the way to solve even more difficult interpolation problems in the future.

References

1. R. Ardon & L.D. Cohen, Fast constrained surface extraction by minimal paths, *Intern. J. of Computer Vision*, Vol.69 (2006), pp.127-136.
2. R. Ardon, L.D. Cohen and A. Yezzi, A new implicit method for surface segmentation by minimal paths in 3D images, *Appl. Math. Optim*, Vol.55 (2007), pp.127-144.

3. R. Ardon, L.D. Cohen and A. Yezzi, Fast surface segmentation guided by user input using implicit extension of minimal paths, *J. of Math. Imaging & Vision*, Vol.25 (2006), pp.289-305.
4. S. Batnitzki et al., Three-dimensional computer reconstruction from surface contours for head CT examinations, *J. of Comp. Assist. Tomogr.*,Vol.5(1981),pp.60-67.
5. Y.-K. Choi & K.-H. Park, A heuristic triangulation algorithm for multiple planar contours using an extended double branching procedure, *Visual Computer*, Vol.10 (1994), pp.372-387.
6. H.N. Christiansen & T.W. Sederberg, Conversion of complex contour lines into polygonal element mosaics, In *Computer Graphics (SIGGRAPH '78 Proceedings)*, R.L. Phillips (ed), Vol.12, 1978, pp.187-192.
7. L.D. Cohen, Minimal paths and fast marching methods for Image Analysis, In *Mathematical Models in Computer Vision: The Handbook*, N. Paragios, Y. Chen and O. Faugeras (eds). Springer 2005, pp.97-111.
8. L.D. Cohen and R. Kimmel, Global minimum for active contour models: A minimal path approach, *Intern. J. of Computer Vision*, Vol.24 (1997), pp.57-78.
9. M. Crandall & P.L. Lions, Viscosity solutions of Hamilton-Jacobi equations, *Trans. Amer. Math. Soc.*, Vol.277 (1983), pp.1-42.
10. M. Crandall & Lions, Two approximations of solutions of Hamilton-Jacobi equations, *Math. of Comp.*, Vol.43, (1984), pp.1-19.
11. A. Efrat, S. Har-Peled, L. Guibas and T. Murali, Morphing between Polylines, In *Proc. of* 12^{th} *Ann. ACM-SIAM Symp. on Discr. Alg. (SODA01)*, 2001, pp.680-689.
12. A.B. Ekoule, F.C. Peyrin and C.L. Odet, A triangulation algorithm from arbitrary shaped multiple planar contours, *ACM Trans. on Graph.*,Vol.10(1991),pp.182- 199.
13. H. Fuchs, Z.M. Kedem and S.P. Uselton, Optimal surface reconstruction from planar contours, *Commun. ACM*, Vol.20 (1977), pp.693-702.
14. G. Gitlin, J. O'Rourke and V. Sabramanian, On reconstructing polyhedra from parallel slices, *Intern. J. of Comp. Geom. & Appl.*, Vol.6 (1996), pp.103-122.
15. S. Hahmann, G.-P. Bonneau, B. Caramiaux and M. Cornillac, Multiresolution morphing of planar curves, *Computing*, Vol.79 (2007), pp.197-209.
16. M. Kass, A. Witkin and D. Terzopoulos, Snakes: Active Contour Models, *Intern. J. of Computer Vision*, Vol.1 (1988), pp.321-331.
17. E. Keppel, Approximating complex surfaces by triangulation of contour lines, *IBM J. Res. Devel.*, Vol.19 (1975), pp.2-11.
18. R. Kimmel, A. Amir and A. Bruckstein, Finding shortest paths on surfaces using level sets propagation, *IEEE Trans.Pat.Anal.Mach.Int.*,Vol.17(1995),pp.635-640.
19. R. Kimmel, N. Kiryati and A. Bruckstein, Sub-pixel distance map and weighted distance transforms, *J. of Math. Imaging & Vision*, Vol.6 (1996), pp.223-233.
20. S. Mauch, *Efficient Algorithms for Solving Static Hamilton-Jacobi Equations*, Doctoral Thesis, California Institute of Technology, Pasadena, California, 2003.
21. A. Melkman, On-line construction of the convex hull of a simple polygon, *Inform. Proc. Letters*, Vol.25 (1987), pp.11-12.
22. E. Rouy & A. Tourin, A viscosity solutions approach to shape-from-shading, *SIAM J. Numer. Anal.*, Vol.29, (1992), pp.867-884.
23. J. Sethian, A fast marching level set method for monotonically advancing fronts, *Proc. Natl. Acad. Sci. USA*, Vol.93, (1996), pp.1591-1595.
24. E. Welzl & B. Wolfers, Surface reconstruction between simple polygons, In *Proceedings, 1st Annual European Symposium on Algorithms (ESA '93), Lecture Notes in Computer Science*, Vol. 726, Springer-Verlag, Berlin/New York, 1993, pp.397-408.
25. A. Yezzi & J. Prince, An Eulerian PDE approach for computing tissue thickness, *IEEE Trans. on Medical Imaging*, Vol.22 (2003), pp.1332-1339.