Implicit and Non-parametric Shape Reconstruction from Unorganized Data Using a Variational Level Set Method

Hong-Kai Zhao*‡ Stanley Osher†§ Barry Merriman†§ Myungjoo Kang†§

Abstract

In this paper we consider a fundamental visualization problem: shape reconstruction from an unorganized data set. A new minimal surface like model and its variational and partial differential equation (PDE) formulation is introduced. In our formulation only distance to the data set is used as our input. Moreover, the distance is computed with optimal speed using a new numerical PDE algorithm. The data set can include points, curves and surface patches. Our model has a natural scaling in the nonlinear regularization that allows flexibility close to the data set while it also minimizes oscillations between data points.

To find the final shape, we continuously deform an initial surface following the gradient flow of our energy functional. An offset (an exterior contour) of the distance function to the data set is used as our initial surface. We have developed a new and efficient algorithm to find this initial surface. We use the level set method in our numerical computation in order to capture the deformation of the initial surface and to find an implicit representation (using the signed distance function) of the final shape on a fixed rectangular grid. Our variational/PDE approach using the level set method allows us to handle complicated topologies and noisy or highly non-uniform data sets quite easily. The constructed shape is smoother than any piecewise linear reconstruction. Moreover our approach is easily scalable for different resolutions and works in any number of space dimensions.

1 Introduction

The problem of shape reconstruction from an unorganized data set has many important applications in medical imaging, scientific computing, computer vision/graphics and range images. The main challenge, especially in three and higher dimensions, is that neither the connections between the data points nor the topology of the final shape is known a priori. Most previous approaches can be classified as parametric or non-parametric (implicit surfaces). Parametric approaches in computational geometry usually use Voronoi diagrams and Delaunay triangulations to find connections between the data points and then construct a triangulated surface by connecting adjacent vertices. There has been much work in this area, for example [7, 2, 3, 4, 17, 16]. Finding the right connections among all data points in three and higher dimensions is generally a challenging problem. Noise in the data and highly non-uniformly sampled data are also difficult to handle. There are also parametric approaches using variational or PDE methods ([6, 12, 21]) where a nice global parametrization is needed. Such a parametrization is almost impossible to obtain for

*Department of Mathematics, University of California, Irvine, CA 92697-3875. e-mail: zhao@math.uci.edu

†Department of Mathematics, University of California, Los Angeles, CA 90095-1555

‡Research supported by NSF DMS 9706566

§Research supported by NSF and DARPA under grants NSF DMS-9706827 and NSF DMS-9615854

complicated topologies. In general, the parametric form lacks depth information and requires a global parametrization which makes shape deformations and topological changes difficult to handle. In recent years the use of implicit surfaces and volumetric methods has attracted a lot of attention [19, 5, 15, 18, 8]. In these approaches a scalar field function is constructed on a fixed grid and the final shape is a particular level set of the field function. Usually the scalar function is the signed distance function to the final shape. Implicit representations usually have topological flexibility, a simple data structure and depth/volumetric information. Such representations can also have memory space efficiency if properly managed [15]. Moreover with the use of the signed distance representation many surface tasks such as Boolean operations, ray tracing for illuminations, computing offsets, etc become quite simple.

In this paper we will introduce a non-parametric variational and PDE formulation. We use a continuous deformation to find the solution. In our numerical computations the level set method [23] is used to capture the deformation of implicit surfaces and to find the signed distance function to the reconstructed shape. In section 2 we first introduce the variational formulation which leads to a minimal surface like model. (We drop the word "like" in what follows, for simplicity of exposition.) The energy functional in our model involves a weighted surface area, i.e. it consists a balance of elastic energy (surface area) and potential energy (distance to the data set). In section 3 we calculate the gradient flow of our energy functional. We start with an initial surface that encloses the data set and uses a continuous deformation which follows the gradient flow to find the solution. In section 4 we briefly review the level set method and derive its variational formulation for our problem. We show that the level set method provides a natural tool for capturing dynamic deformations of implicit surfaces. In section 5 details of the numerical implementation are presented. A very efficient and robust PDE based numerical algorithm is introduced to find the function which is the absolute distance to the data set on a rectangular grid. A good initial surface for our dynamic procedure is obtained using a level contour of this distance function. Numerical techniques for the level set method and for a multiresolution framework are also discussed. Finally in section 6 numerical examples are presented. In the appendix we present some rigorous analysis of our model. Important features of our approach are as follows:

1. We use a variational and PDE based formulation that only depends on the distance to a general data set that can include points, curves and surface patches. Moreover we have a very efficient PDE based numerical algorithm to find this distance function.

2. The nonlinear regularization, which involves curvature, in our minimal surface model has an adaptive scaling property that allows for varying flexibility. The constructed shape is smoother than any piecewise linear approximation in three and higher dimensions.

3. Numerically, we use the level set method on rectangular grids to find an implicit representation using the signed distance function to the reconstructed shape.

4. Complicated topologies and noisy or non-uniform data sets can be handled easily.

5. Our method is easily scalable for multiple resolutions and works in any number of space dimensions. It can also be used as a smoothing procedure to postprocess other piecewise linear approaches.

2 The Minimal Surface Model

Let $\mathcal{S}$ denote the data set. In our formulation $\mathcal{S}$ can include points, pieces of curves and surfaces. The technique might also be generalized to reconstruct curves in R^3 using a new vector valued level set method designed in [10]. Define $d(\boldsymbol{x}) = dist(\boldsymbol{x}, \mathcal{S})$ to be the distance function to $\mathcal{S}$. From now on we use bold faced characters for vectors.

We define the following surface energy for our variational formulation

$$E(\Gamma) = [\int_\Gamma d^p(\boldsymbol{x}) ds]^{1/p} \qquad (1)$$

Here Γ is an arbitrary surface and ds is the surface area. The energy functional is independent of parametrization and invariant under rigid transformation. If we think of the distance function as a potential function for $\mathcal{S}$, then our energy is the L^p norm of the potential on Γ. For a general data set the distance function is the only intrinsic information we have. When $p = \infty$, $E(\Gamma)$ is the value of the distance of the most remote point on Γ. We try to find a local minimizer of our energy functional that behaves like a minimal surface or an elastic membrane attached to the data set.

We can easily calculate the first variation of the surface energy with respect to a small perturbation of Γ.

$$\frac{\delta E(\Gamma)}{\delta \Gamma} = \frac{1}{p}[\int_\Gamma d^p(\boldsymbol{x}) ds]^{1/p-1}[p d^{p-1} \nabla d \cdot \boldsymbol{n} + d^p \kappa] \quad (2)$$

Here $\boldsymbol{n}$ is the unit normal and κ is the mean curvature of Γ. The first term in the last square bracket above corresponds to a variation in the potential and the second term corresponds to a variation in the surface area. Hence the Euler-Lagrange equation is

$$d^{p-1}(\boldsymbol{x})[\nabla d(\boldsymbol{x}) \cdot \boldsymbol{n} + \frac{1}{p} d(\boldsymbol{x}) \kappa] = 0 \qquad (3)$$

This PDE is well defined except at kinks of $d(\boldsymbol{x})$ and the solution is interpreted as the viscosity solution [14] of the level set PDE defined in (9) below. We can see a balance between the potential force $\nabla d(\boldsymbol{x}) \cdot \boldsymbol{n}$ (to minimize the potential energy) and the surface tension $d(\boldsymbol{x})\kappa$ (to minimize the surface energy). Moreover the nonlinear regularization due to surface tension has a desirable scaling $d(\boldsymbol{x})$. Thus our membrane is more flexible close to the data and is more rigid away from the data. Equation(3) above enforces a local condition for the sampling density, i.e higher sampling density is required to resolve a fine feature. Another interesting parameter is p which affects the flexibility of the membrane.

In the continuous limit, when $d(\boldsymbol{x})$ is the distance function to a smooth surface Γ_0, we have two global minima, i.e. $\Gamma = \Gamma_0$ and $\Gamma = \emptyset$. In general, $\mathcal{S}$ is a discrete set and things become more interesting and complicated. In particular, there is only one trivial global minimum $\Gamma = \emptyset$ (the empty set). We are only interested in finding the local minimum that corresponds to our minimal surface attached to the data set. The local minimum solves the Euler-Lagrange equation (3). The existence of such a minimal surface approximation to the real shape depends on whether the sampling density of the data can resolve the real shape.

In two dimensions we show in the appendix that a polygon which connects adjacent points by straight lines is a local minimum. This fact is based on two simple results:

1. The polygon is a equilibrium solution to (3).

2. The polygon is a local minimum of the energy.

This result has a dual meaning. On one hand it shows a connection between our variational formulation and previous approaches. On the other hand it shows that our formulation does not provide any new insights in two dimensions since a minimal surface passing through two points is a straight line. However in three dimensions the picture is totally different and is much more interesting. Our minimal surface approximation will be smoother than a polyhedron.

3 Continuous Deformation Under the Gradient Flow

To construct the minimal surface we will use a continuous deformation which follows the gradient descent of our energy functional. We start with a initial surface that encloses all data and follow the gradient flow of (1)

$$\frac{d\Gamma}{dt} = -\frac{1}{p}\left[\int_\Gamma d^p(\boldsymbol{x})ds\right]^{1/p-1}\nabla\cdot(d^p(\boldsymbol{x})\boldsymbol{n})\boldsymbol{n}$$
$$= -\left[\int_\Gamma d^p(\boldsymbol{x})ds\right]^{1/p-1}d^{p-1}(\boldsymbol{x})\left[\nabla d(\boldsymbol{x})\cdot\boldsymbol{n} + \frac{1}{p}d(\boldsymbol{x})\kappa\right]\boldsymbol{n} \tag{4}$$

If, initially the membrane is far away from the data set, both the potential force $\nabla d(\boldsymbol{x})\cdot\boldsymbol{n}$ and the surface tension $d(\boldsymbol{x})\kappa$ make the surface shrink. When the surface is close to the data there is a balance between those two forces. The reason we do not initialize with a surface lying inside the data set is that the surface may shrink and disappear by converging to the global minimum, i.e. to the empty set, unless it starts out very close to the data set. If $p \gg 1$,

$$\frac{d\Gamma}{dt} \approx -\left[\frac{d(\boldsymbol{x})}{d_{max}}\right]^{p-1}\left[\nabla d(\boldsymbol{x})\cdot\boldsymbol{n} + \frac{1}{p}d(\boldsymbol{x})\kappa\right]\boldsymbol{n}$$

where

$$d_{max} = \max_{\boldsymbol{x}\in\Gamma} d(\boldsymbol{x}) \approx \left[\int_\Gamma d^p(\boldsymbol{x})ds\right]^{1/p}$$

The scaling factor $[\frac{d(\boldsymbol{x})}{d_{max}}]^{p-1}$ helps make the most remote points on the surface move most quickly. As the most remote points close in, other points take their place and close in. In practice we would like to have the whole surface close in more or less simultaneously, so we find that $p = 2$ (similar to a least squares method) is a good choice, in general.

There are two immediate questions we need to consider:

1. Will the initial membrane get stuck at some other local minimum before it is close to all the data points?
2. Will the membrane collapse through the data set?

In general, for shapes having complicated geometry and/or topology, if the initial guess is too far from the data the answer to the first question might well be yes. In practice, a good initial surface is very important for obtaining the right shape as well as for fast convergence. In our numerical computations we use an offset of the distance function of the data set, i.e. a contour for which $d(\boldsymbol{x}) = \epsilon$ is our initial surface. We present evidence in the appendix that, in the continuous limit, if the initial surface is a level contour close enough to the real surface, then it has a larger energy than the real surface. Moreover if the sampling density can resolve the geometry of the real surface, we can find an $\epsilon > 0$ so that the exterior contour for which $d(\boldsymbol{x}) = \epsilon$ is homeomorphic to the real surface and is a good initial guess. We do not have a rigorous answer to the second question in three dimensions. The result depends on the grid resolution and the sampling density. From (3) and as is seen in our numerical experiments, it appears that if both grid size and data sample density can resolve the geometry of the real surface then we do obtain a weighted minimal surface approximation to the real surface.

At equilibrium we have

$$\nabla d(\boldsymbol{x})\cdot\boldsymbol{n} = -\frac{1}{p}d(\boldsymbol{x})\kappa, \quad a.e.$$

Since the maximum of $d(\boldsymbol{x})$ on our approximating surface is inversely proportional to the density of the data set, we have local conditions on the sampling density that are similar to those prescribed in [4].

4 Variational Level Set Method

In general we do not have any a priori knowledge about the topology of the surface to be approximated. Moreover topological changes may occur during our continuous deformation process. This makes a consistent parametrization almost impossible. We shall use the level set method to handle topological changes and to obtain an implicit surface. We give a brief review of the level set method, introduced by Osher and Sethian [23], for capturing moving interfaces. Consider a closed surface $\Gamma(t)$ of co-dimension one in R^n moving with time. Let $\Omega(t)$ be the (generally multiply connected) region enclosed by $\Gamma(t)$. Let $\phi(\boldsymbol{x},t)$, be the level set function associated with $\Omega(t)$, i.e.

$$\begin{array}{lclcl} \phi(\boldsymbol{x},t) & > & 0 & \text{in} & \Omega(t) \\ \phi(\boldsymbol{x},t) & = & 0 & \text{on} & \Gamma(t) \\ \phi(\boldsymbol{x},t) & < & 0 & \text{in} & \bar{\Omega}^c(t) \end{array} \qquad (5)$$

We have thus identified $\Gamma(t)$ as the zero level set of $\phi(\boldsymbol{x},t)$. Differentiating the equation:

$$\phi(\Gamma(t),t) \equiv 0,$$

with respect to t, leads to a time dependent PDE for ϕ which holds on $\Gamma(t)$

$$\phi_t + \frac{d\Gamma(t)}{dt}\cdot\nabla\phi = 0 \Leftrightarrow \phi_t + v_n|\nabla\phi| = 0. \qquad (6)$$

Here, v_n is the normal velocity of $\Gamma(t)$ which may depend on external physics or global, local or geometric quantities. Instead of tracking the moving surface $\Gamma(t)$ we only have to solve this PDE for ϕ on a rectangular grid near the set for which $\phi = 0$. The data structure becomes extremely simple and topological changes can be handled easily. The apparent extra dimension worth of calculation is easily removed by a simple "local level set method" [24], i.e. the computation is restricted to a narrow band close to the zero level set. The level set method has been used quite successfully in moving interface/boundary problems as well as in image processing, image segmentation etc. [11, 20, 13]. See [22] for a recent comprehensive review article.

Now we will follow the procedure introduced in [28, 29] to obtain the variational level set formulation for our problem. If Γ is the zero level set of $\phi(\boldsymbol{x})$ then

$$\begin{aligned} E(\Gamma) &= [\textstyle\int_\Gamma d^p(\boldsymbol{x})ds]^{1/p} \\ &= E(\phi) = [\textstyle\int d^p(\boldsymbol{x})\delta(\phi(\boldsymbol{x}))|\nabla\phi(\boldsymbol{x})|d\boldsymbol{x}]^{1/p} \end{aligned} \qquad (7)$$

where $\delta(x)$ is the one dimensional Delta function and $\delta(\phi(\boldsymbol{x}))|\nabla\phi(\boldsymbol{x})|dx$ is the surface area element at the zero level set of ϕ. The corresponding gradient flow for ϕ (See [28, 29] for details.) is

$$\frac{\partial\phi}{\partial t} = \frac{1}{p}\delta(\phi)\left[\int d^p(\boldsymbol{x})\delta(\phi)|\nabla\phi|d\boldsymbol{x}\right]^{1/p-1}\nabla\cdot\left[d^p(\boldsymbol{x})\frac{\nabla\phi}{|\nabla\phi|}\right] \qquad (8)$$

As usual we extend this geometric motion at $\phi = 0$ to other level sets by replacing $\delta(\phi)$ with $|\nabla\phi|$. This makes the flow morphological, [1], i.e. it is independent of the scaling of ϕ. This means that if we replace ϕ by $h(\phi)$ with $h(0) = 0$ and $h'(\phi) > 0$, the equation is invariant. Equation (8) can be expanded as

$$\begin{aligned} &\tfrac{\partial\phi}{\partial t} = \tfrac{1}{p}|\nabla\phi|\left[\textstyle\int d^p(\boldsymbol{x})\delta(\phi)|\nabla\phi|d\boldsymbol{x}\right]^{1/p-1}\nabla\cdot\left[d^p(\boldsymbol{x})\tfrac{\nabla\phi}{|\nabla\phi|}\right] \\ &= |\nabla\phi|\left[\textstyle\int d^p(\boldsymbol{x})\delta(\phi)|\nabla\phi|d\boldsymbol{x}\right]^{1/p-1} \\ &\times d^{p-1}(\boldsymbol{x})\left[\nabla d(\boldsymbol{x})\cdot\tfrac{\nabla\phi}{|\nabla\phi|} + \tfrac{1}{p}d(\boldsymbol{x})\nabla\cdot\tfrac{\nabla\phi}{|\nabla\phi|}\right] \end{aligned} \qquad (9)$$

As usual $\frac{\nabla\phi}{|\nabla\phi|}$ and $\nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}$ are the level set representations of the unit normal and the mean curvature respectively. We can show that under the flow defined by (9), $E(\phi)$ always decreases until it reaches equilibrium.

$$\begin{aligned} &\frac{d}{dt}E_\Gamma^{(p)}(\phi) = -\frac{1}{p}\left(\bar{d}^{(p)}\right)^{1-p}\int_\Omega \delta(\phi)\nabla\cdot\left(d^p(x)\frac{\nabla\phi}{|\nabla\phi|}\right)\phi_t dx \\ &= -\frac{1}{p^2}\left(\bar{d}^{(p)}\right)^{2(1-p)}\int_\Omega \delta(\phi)|\nabla\phi|\left[\nabla\cdot\left(d^p(x)\frac{\nabla\phi}{|\nabla\phi|}\right)\right]^2 dx \\ &\leq 0 \end{aligned}$$

Here:

$$\bar{d}^{(p)} \overset{define}{=} \left(\int_\Omega d^p(x)\delta(\phi)|\nabla\phi|dx\right)^{\frac{1}{p}} = E(\Gamma)$$

However the scaling factor $\left[\frac{d(x)}{\bar{d}^{(p)}}\right]^{p-1}$ in the equation (9), which makes the most remote points move most quickly, could cause a problem in the implementation. The quantity $\left[\frac{d(x)}{\bar{d}^{(p)}}\right]^{p-1}$ can be huge for points x belonging to level sets of ϕ which are far from the data set $\mathcal{S}$ when the zero level set is close to the data set, i.e. when $\bar{d}^{(p)} \approx 0$. This difficulty is removed because our *local* level set method uses only points in a narrow band of the zero level set and our grid resolution is taken to be comparable to the density of the data points.

The initial data $\phi(\boldsymbol{x},0)$ for our time dependent PDE is the level set function (which we take to be the signed distance function) for our initial surface. The evolution PDE defined by (9) for ϕ captures the continuous deformation of the surface. Numerically, the computation is done on a simple rectangular grid. As usual, although we start with a level set function that is a signed distance function, we have to use a redistancing algorithm from [26, 24] to maintain $|\nabla\phi(\boldsymbol{x})| = 1$ near the zero level set during the evolution. At convergence we find the signed distance function to the minimal surface that solves (3).

All geometric functions of the final surface such as mean curvature, Gaussian curvature, surface area, volume, etc. can be easily computed in terms of the level set function ϕ. This information can be very useful in applications. For example we can calculate the volume and surface area of the reconstructed shape using the following formula.

$$Volume = \int H(-\phi)d\boldsymbol{x}, \quad Area = \int \delta(\phi)|\nabla\phi|d\boldsymbol{x}$$

Here $H(x)$ is the Heaviside function which is defined as

$$H(x) = \begin{cases} 1 & x > 0 \\ 0 & x < 0 \end{cases}$$

We use numerical approximations to $H(x)$ and $\delta(x)$ as defined in [28] in our computations.

5 Numerical Implementation

5.1 Finding Distance to the Data

The distance function to the data set is used in our formulation. There are many algorithms which can be used to compute this function. Here we use a novel PDE based approach which requires $O(N)$ operations in order to find the distance function at N grid points.

The distance function $d(\boldsymbol{x})$ to a data set $\mathcal{S}$ satisfies the eikonal equation (in the sense of viscosity solutions [14]) with boundary conditions:

$$|\nabla d(\boldsymbol{x})| = 1 \qquad d(\boldsymbol{x} \in \mathcal{S}) = 0$$

$\mathcal{S}$ can include points, curves or surfaces. In practice we only need boundary data for $d(\boldsymbol{x})$ at neighboring grid points to $\mathcal{S}$. We solve the eikonal equation using upwind schemes and Gauss-Seidel iterations with different ordering of the sweeps. The algorithm is as follows: We fix the distance values at given grid points and use the following first order monotone upwind discretization at grid points (i,j) for which $d(x_i, y_j)$ is not known (for simplicity of exposition, we present the two dimensional algorithm)

$$[(u_{i,j} - x_{min})^+]^2 + [(u_{i,j} - y_{min})^+]^2 = h^2 \tag{10}$$

where h is the grid size, $(x)^+ = \begin{cases} x & x > 0 \\ 0 & x \leq 0 \end{cases}$ and

$$x_{min} = min(u_{i-1,j}, u_{i+1,j}) \quad y_{min} = min(u_{i,j-1}, u_{i,j+1})$$

This discretization was first used in [25]. We see that only smaller values of the neighbors, i.e. the values of the neighbors that are closer to the data set, (upwind along the characteristics of the eikonal equation) are used to approximate the first derivative u_x, u_y. We use Gauss-Seidel iteration to solve this system of non-linear equations for all the unknown values in some prescribed order. We solve (10) at each grid point for $u_{i,j}$ given its most recently updated four neighbors. Since the left hand side of (10) is continuous it is not hard to see that there is exactly one solution $u_{i,j}$ of (10) and that is monotone nondecreasing as a function of its neighbors and that it satisfies

$$min(x_{min}, y_{min})) < u_{i,j} \leq min(x_{min}, y_{min})) + h$$

Actually there is an exact formula for $u_{i,j}$:

$$u_{i,j} = \begin{cases} min(x_{min}, y_{min}) + h, & if\ |x_{min} - y_{min}| \geq h \\ \frac{x_{min}+y_{min}+\sqrt{2h^2-(x_{min}-y_{min})^2}}{2} & if\ |x_{min} - y_{min}| < h \end{cases}$$

The crucial sweeping idea is to alternate the ordering in the Gauss-Seidel iterations. For example in two dimensions the system of nonlinear equations (10) for $u_{i,j}$ can be solved by alternating the following four orderings as was used in [9].

$$\begin{array}{ll} (1)\ i=1:n, j=1:n & (2)\ i=1:n, j=n:1 \\ (3)\ i=n:1, j=1:n & (4)\ i=n:1, j=n:1 \end{array}$$

Usually, convergence takes no more than one cycle of four sweeps plus one more arbitrary sweep, i.e. the operation count is $5N$.

In three dimensions a straightforward extension works and $8+1=9$ sweeps of this iterative procedure are needed. Examples are shown in the numerical section. More details and proof will be discussed in a future paper.

Given M data points, we first locate each data point within a grid cell and find the exact distance to the data point at the vertices of the grid cell. This requires $O(M)$ operations. We use these exact distance values as initial data and solve the eikonal equation to find $d(\boldsymbol{x})$ at all other grid points. It thus takes $O(M+N)$ operations to find the distance function to M data points at N grid points. This procedure can also be applied to data sets that contain curves or surfaces.

5.2 Finding a Good Initial Guess

In our approach we continuously deform an initial surface to the final surface by following the gradient flow of our energy functional. A nonlinear parabolic PDE, equation (8) is solved for the level set function to capture the deformation. We can start with a simple surface such as a plane, a sphere or a rectangular box as we show below in our the numerical examples. According to the well known CFL condition for a parabolic PDE, the time step δt is restricted to the order of $O(h^2)$. If we start with an initial surface that is too far from the real shape we have to evolve the PDE for a long time and the computation cost is therefore quite expensive. A good initial surface helps to speed up convergence to the equilibrium surface. In many situations a good initial surface is also needed to avoid spurious local minimum of our energy. We have found the outer level contour of the distance function $d(\boldsymbol{x})=\epsilon$ that encloses the data set is a very good initial guess. Since the contour $d(\boldsymbol{x})=\epsilon$ most likely has a shell like structure we use the following simple method to find the exterior contour surface and its associated signed distance (level set) function.

First we use a simple tagging algorithm to find those grid points (i,j,k) that are connected to any fixed grid point, $(1,1,1)$, for example, that is outside of our approximating surface and for which $d_{i,j,k} \geq \epsilon$. We label these grid points as exterior points. The simple tagging algorithm goes like this: if (i,j,k) is an exterior point we find those neighbors $(i\pm 1, j\pm 1, k\pm 1)$ which satisfy $d_{i\pm 1, j\pm 1, k\pm 1} \geq \epsilon$ and which have not been tagged yet. At each step we store these newly tagged points and use them as an expanding boundary to tag more points in the next step. Simultaneously we also remember those grid points that have a distance value less than ϵ and are adjacent to those tagged exterior points. The whole tagging process will be applied to every exterior point only once.

Next we find the signed distance function $d_0(\boldsymbol{x})$ to this exterior contour $d(\boldsymbol{x})=\epsilon$ using the fast PDE algorithm we just discussed above. First we notice that the exterior contour of $d(\boldsymbol{x})=\epsilon$ is enclosed between those tagged outside points and their adjacent inside neighbors. The correct distance values at all outside points and their adjacent inside points are $|d_0(\boldsymbol{x})| = |d(\boldsymbol{x})-\epsilon|$. We fix these correct values for $|d_0(\boldsymbol{x})|$ and use them as the initial values to find the distance values $|d_0(\boldsymbol{x})|$ at those non-tagged interior points. Then we negate the distance values at all interior points. We thus construct a signed distance function $d_0(\boldsymbol{x})$ of the exterior contour.

In the continuous limit, if $d(\boldsymbol{x})$ is the distance function to a smooth surface, the zero level set $\{\boldsymbol{x} : d(\boldsymbol{x}) = \epsilon = 0\}$ is the true surface. From this point of view we would like to choose ϵ as small as possible. However in the discrete case if we choose ϵ too small the contour $d(\boldsymbol{x}) = \epsilon$ consists of separated small spheres around data points. To get a good initial surface the choice of the parameter ϵ depends on the sampling density and the local feature size, a concept introduced in [3], which is related to local curvature and to the distance to the medial axis. Theoretically a good initial surface should "see" all data points, i.e. the initial surface should intersect all the Voronoi cells attached to the data points. If our sampling density satisfies $\frac{1}{\alpha} > \frac{r}{2}$, where r is the maximum distance between two connected data points in $\mathcal{S}$ and α is the minimum local feature size, then we can choose an exterior contour $d(\boldsymbol{x}) = \epsilon$, $\frac{1}{\alpha} > \epsilon > \frac{r}{2}$ such that it is homeomorphic to the true shape. If the sample density is fairly uniform and if it resolves all local features of the real shape, then we can find an appropriate ϵ. In practice, it is very easy and fast to find an outer contour of the distance function, and this is usually a remarkably good approximation of the real shape. All it takes is $O(M+N)$ operations to find such a contour.

If the sampling density is not uniform but if it can resolve all local features then we have to use the outer contour of $d(\boldsymbol{x})$, i.e. $d(\boldsymbol{x}) = \epsilon(\boldsymbol{x})$, where $\epsilon(\boldsymbol{x})$ is proportional to local feature size and/or is inversely proportional to sampling density.

We also note that we can combine our variational level set based algorithm with any other algorithm that provides a good initial approximation. Moreover our algorithm can also be used to post process any other approximations in order to obtain a smoother surface reconstruction.

5.3 Solving the PDE for the Level Set Function

Numerically, we use the same local level set method devised and implemented in [28, 24] to solve equation (9). At each time step the computation is restricted to a narrow band around the zero level set. We also need to use a redistancing algorithm to keep the level set function close to signed distance function near the zero level set, as is described in [24]. If the total number of grid points is N then at each time step the operation count is roughly $O(N^{2/3})$ in three dimensions. Since we use an explicit scheme to solve the parabolic equation (9), the CFL condition on the time step is $\delta t \approx h^2$. If we start with an outer contour of the distance function our initial surface is located a few grid cells away from the final surface. We need roughly $O(1/h) \approx O(N^{1/3})$ time steps for convergence. So the computational complexity to solve the PDE is $O(N)$. Our stopping criteria are the following: Either all data points are close enough to the zero level set, i.e. $|\phi(\boldsymbol{x})| < tol$ for any $\boldsymbol{x} \in \mathcal{S}$ (ϕ is a signed distance function), or $|\phi_t|$ is small enough, which means that we are close to an equilibrium state.

The total complexity of our method is $O(M+N)$. If the grid resolution is comparable to the sampling density then $M \approx N^{2/3}$. Thus, the total complexity of our method is $O(M+N) \approx O(M^{3/2})$. Since our reconstruction is smoother than piecewise linear, we need a lower sampling density to get the similar result.

In future work we shall introduce a simple convection equation to find an piecewise linear approximation of the final shape, and then we shall postprocess with a few iterations of our minimal surface algorithm to reduce the total complexity to $O(M)$.

5.4 Multiresolution Framework

We have two resolutions in our formulation. One is the resolution of the data set, i.e. the sampling density. The other is the resolution of our rectangular grid on which we solve the equation (9) to find the signed distance function to the final shape. We can use different grid sizes for different resolutions. Moreover we can use a coarse resolution solution as an initial guess for a fine resolution result in a multiresolution framework. An adaptive grid can also be used to optimize efficiency and resolution.

5.5 Manipulation of Implicit Surfaces

After we have constructed a level set (signed distance) representation of a shape, then many surface/shape manipulations become very simple. (1) Geometric properties of the surface can be recovered from its level set representation quite easily. (2) The level set method can be used to easily handle motion, deformation and interaction of surfaces in animation, simulation and morphing. (3) Boolean operations (shown in the next section), finding shape offsets (a contour of the signed distance function) and ray tracing (using the fact that closest point on the surface to a point $\boldsymbol{x}$ is $\boldsymbol{x} - d(\boldsymbol{x})\nabla d(\boldsymbol{x})$), become almost trivial. To find the number of connected components we can use the following algorithm. We start with an arbitrary grid point whose level set function value is negative (an interior point) and tag all its neighbor grid points whose level set function value are also negative. We use the newly tagged interior points to tag their neighboring interior points. When no more interior points can be tagged in this way, if all remaining grid points have positive values (exterior points) then there is only one connected component. Otherwise we label those tagged interior points as component one and start from a untagged interior point and tag the second connected component. After all interior points are tagged, we have found and labeled all connected components. (4) The memory needed to store an implicit surface is almost optimal. We only need to store the indices and signed distance values of those grid points which are next to the surface, i.e. grid points such that one of its neighbors has a different sign of the level set function. Since the data structure is so simple there is no overhead at all. For most rendering algorithms, such as the well known marching cube method, the information we store is enough to render or triangulate the surface. If the signed or unsigned distance function is needed in a larger domain for rendering or simulation, we can find it easily using the fast PDE algorithm mentioned above, or any other fast algorithm to find the distance function, e.g. in [27].

6 Numerical Examples

In this section we present some numerical examples. We use vis5d packages available at http://www.ssec.wisc.edu/ billh/vis5d.html to visualize the implicit surfaces. The calculations are done on a Pentium III with 600 Mhz speed and 1GB memory. The running time depends primarily on the total number of grid points and slightly on the number of data points. (The number of data points is important only when we calculate the distance function). Using the outer contour of $d(\boldsymbol{x}) = 3h$, h is the grid size, as our initial shape it typically takes about 10-15 minutes for our algorithm to converge, e.g., for the last four realistic examples shown in figures 10-13. In future work we shall use a simplified convection model together with our present formulation here to accelerate the computation by at least a factor of five. Further acceleration will also be obtained by using an adaptive multiresolution approach to the gridding.

In figure 1 we show the computation of distance function in two dimensions of four linked circles using the fast PDE algorithm. This figure also shows that the contours of the distance function can be a good approximation of the real shape. In figure 1(a) only isolated points are given. In figure 1(b) the distance values at all the grid points next to the four linked circles are given.

Figure 2 shows Boolean operations of two shapes, a ball and a cube. If the two shapes are represented by signed distance functions $\phi_1(\boldsymbol{x})$ and $\phi_2(\boldsymbol{x})$, then the zero level set of the functions

$$\phi(\boldsymbol{x}) = \min(\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x})),$$
$$\phi(\boldsymbol{x}) = \max(\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x})),$$
$$\phi(\boldsymbol{x}) = \max(\phi_1(\boldsymbol{x}), -\phi_2(\boldsymbol{x})),$$
$$\phi(\boldsymbol{x}) = \max(-\phi_1(\boldsymbol{x}), \phi_2(\boldsymbol{x}))$$

represent the union, intersection and differences, respectively, of the two shapes.

In figure 3 the data set for a two headed cone is composed of two bottom discs, six circles on each cone and two vertex points. The initial shape is a box that encloses all data.

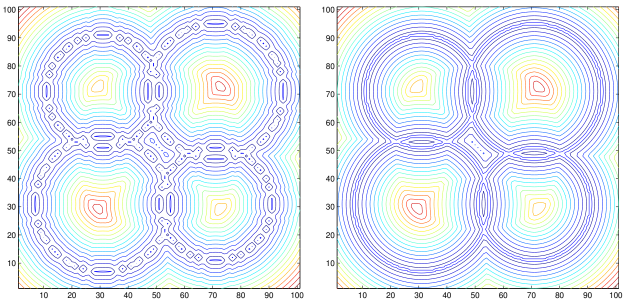
Figure 8 (a)-(g) shows a box being deformed into two linked tori. Although we start with this very rough initial guess, our gradient flow allows us to find the final shape. We also see that there is no problem in capturing topological changes during the deformation. Figure 8(h) shows the approximation on a finer resolution.

Figure 4 shows the reconstruction of a graph starting from a flat sheet.

The data sets in figure 9 are taken from http://ftp.research.microsoft.com/users/hhoppe/data/thesis. Figure 10, 11 and 12 are reconstructed surfaces from the data sets in figure 9. In each figure both the initial surface using an outer contour of the corresponding distance function and the final reconstructed surface are shown. In figure 13 we reconstructed a rat brain using data points from MRI slices. The data points are highly non-uniform.

7 Conclusions and future work

In this paper we present a minimal surface like model for shape reconstruction using variational formulation and PDE based method. We use a continuous deformation following the gradient flow of our energy to find the solution. In our numerical implementation the level set method provides a powerful and efficient tool to capture the shape deformation and to find the signed distance function to the final shape on a simple rectangular grid. We also use an optimal PDE based algorithm to find the value of the distance function at the rectangular grid nodes and at the same time use an outer contour of the distance function as our initial guess. Our formulation and algorithm works for quite general data sets in any number of space dimensions. In three or higher dimensions our constructed surfaces are smoother than piecewise linear reconstructions. In our future work we will introduce a simplified convection model together with our present algorithm to further speed up the computation. We will also incorporate more constraints such as unit normals, fixed volumes and uncertainties in the data in our variational formulation.



(a) distance to discrete points. (b) distance to continuous curves

Figure 1: computation of distance function in two dimensions

Acknowledgement: We would like to thank Dr. Ronald Fedkiw for many profitable discussions.

References

[1] L. Alvarez, F. Guichard, P.-L. Lions, and J.-M. Morel. Axioms and fundamental equations of image processing. *Arch. Rat. Mechanics*, 123:199–257, 1993.

[2] N. Amenta and M. Bern. Surface reconstruction by Voronoi filtering. *14th ACM Symposium on Computational Geometry*, 1998.

[3] N. Amenta, M. Bern, and D. Eppstein. The crust and the β-skeleton: combinatorial curve recon-

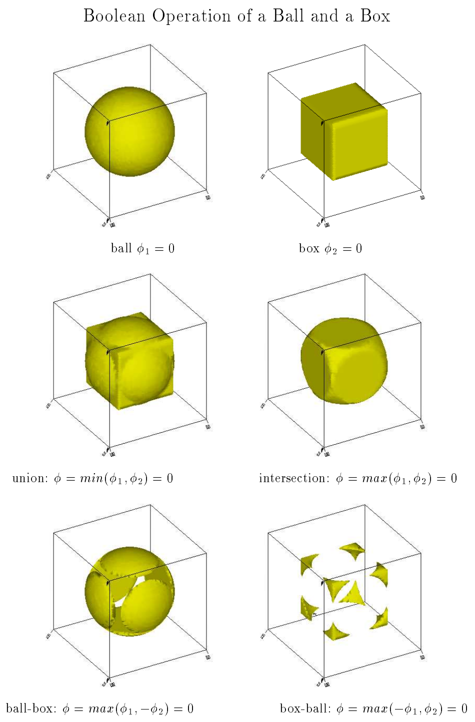


Figure 2:

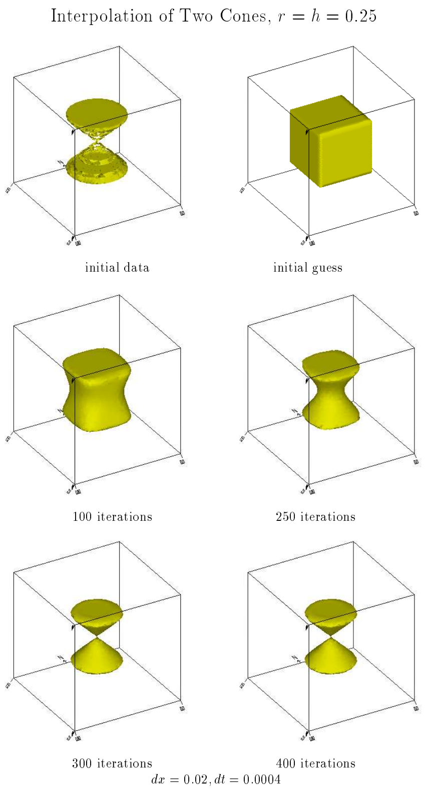


Figure 3:

struction. *Graphical Models and Image Processing*, 60:125–135, 1998.

[4] N. Amenta, M. Bern, and M. Kamvysselis. A new Voronoi-based surface reconstruction algorithm. *Proc. SIGGRAPH'98*, pages 415–421, 1998.

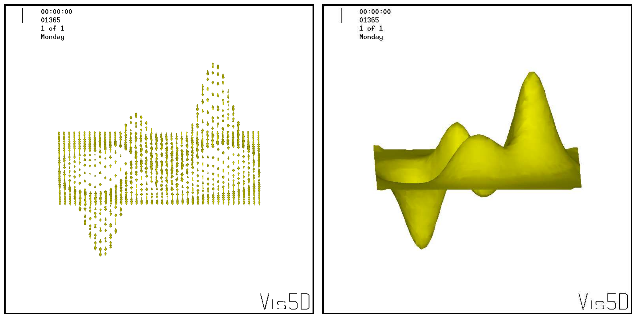
[5] C. Bajaj, F. Bernardini, and G. Xu. Automatic reconstruction of surfaces and scalar fields from 3D scans. *SIGGRAPH'95 Proceedings*, pages 193–198, July 1995.

[6] M. Bloor and M. Wilson. Using partial differential equations to generate free-from surface. *Computer Aided Des.*, (22):257–266, 1990.

[7] J. D. Boissonnat. Geometric structures for three dimensional shape reconstruction. *ACM Trans. Graphics 3*, pages 266–286, 1984.

[8] J.D. Boissonnat and F. Cazals. Smooth shape reconstruction via natural neighbor interpolation of distance functions. *ACM Symposium on Computational Geometry*, 2000.

[9] M. Boué and P. Dupuis. Markov chain approximations for deterministic control problems with



(a) 1600 data points (b) reconstructed shape

Figure 4: reconstruction of a graph

affine dynamics and quadratic cost in the control. *SIAM J. NUMER. ANAL.*, 36(3):667–695, 1999.

[10] P. Burchard, L.-T. Cheng, B. Merriman, and S. Osher. Motion of curves in three spatial dimensions using a level set approach. Technical report, UCLA, CAM Report 00-29, 2000.

[11] V. Caselles, R. Kimmel, G. Sapiro, and C. Sbert. Minimal surfaces based object segmentation. *IEEE Trans. Pattern Analysis Machine Intelligence*, 19:4:394–398, 1997.

[12] G. Celniker and D. Gossard. Deformable curve and surface finite element for free-form shape design. *Computer Graphics*, (25):257–266, 1991.

[13] T. Chan and L.A. Vese. Active contours without edges. *UCLA CAM report, 98-53*, 1998.

[14] M. Crandall, H. Ishii, and P.L. Lions. Users's guide to viscosity solutions. *Amer. Math. Soc. Bull.*, v27:1–67, 1992.

[15] B. Curless and M. Levoy. A volumetric method for building complex models from range images. *SIGGRAPH'96 Proceedings*, pages 303–312, 1996.

[16] H. Edelsbrunner. Shape reconstruction with Delaunay complex. In *Proc. of LATIN'98: Theoretical Informatics*, volume 1380 of *Lecture Notes in Computer Science*, pages 119–132. Springer-Verlag, 1998.

[17] H. Edelsbrunner and E. P. Mücke. Three dimensional α shapes. *ACM Trans. Graphics 13*, pages 43–72, 1994.

[18] A. Hilton, A.J. Stoddart, J. Illingworth, and T. Windeatt. Implicit surface - based geometric fusion. *Comput. Vision and Image Understanding*, 69:273–291, 1998.

[19] H. Hoppe, T. DeRose, T. Duchamp, J. McDonald, and W. Stuetzle. Surface reconstruction from unorganized points. *SIGGRAPH'92 Proceedings*, pages 71–78, 1992.

[20] A. Marquina and S. Osher. Explicit algorithms for a new time dependent model based on level set motion for nonlinear deblurring and noise removal. *UCLA CAM report, 99-5*, 1999.

[21] T. McInerney and D. Terzopoulos. A finite element model for 3D shape reconstruction and nonrigid motion tracking. *Proc. IEEE 4th Int. Conf. on Comp. Vision*, pages 518–523, 1993.

[22] S. Osher and R. Fedkiw. Level set methods. *UCLA CAM Report 00-08*, 2000.

[23] S. Osher and J. Sethian. Fronts propagating with curvature dependent speed, algorithms based on a Hamilton-Jacobi formulation. *J. Comp. Phys.*, 79:12–49, 1988.

[24] D. Peng, S. Osher, B. Merriman, H. Zhao, and M. Kang. A PDE based fast local level set method. *J. Comput. Phys.*, (155):410–438, 1999.

[25] E. Rouy and A. Tourin. A viscosity solutions approach to shape-from-shading. *SIAM J. Num. Anal.*, v29:867–884, 1992.

[26] M. Sussman, P. Smereka, and S. Osher. A level set approach for computing solutions to incompressible two-phase flows. *J. Comp. Phys.*, (119):146–159, 1994.

[27] J.N. Tsitsiklis. Efficient algorithms for globally optimal trajectories. *IEEE Transactions on Automatic Control*, 40(9):1528–1538, 1995.

[28] H. Zhao, T. Chan, B. Merriman, and S. Osher. A variational level set approach to multiphase motion. *J. Comput. Phys.*, (127):179–195, 1996.

[29] H. Zhao, B. Merriman, S. Osher, and L. Wang. Capturing the behavior of bubbles and drops using the variational level set approach. *J. Comput. Phys.*, (143):495–518, 1998.

Appendix: Local Analysis

First we use perturbation analysis to show the validity of the following statement in two dimensions:

Theorem 1 *A polygon that connects adjacent points with a straight line is a local minimum of the energy functional (1).*

Proof: In the neighborhood of a point x_i, $d(x) = |x - x_i|$ and ∇d points along the line from x_i to x. It is easy to see from figure 5 that in this neighborhood

$$\nabla d \cdot \nabla \phi + \frac{1}{p} d \nabla \cdot \frac{\nabla \phi}{|\nabla \phi|} = 0.$$

The angle θ between two line segments is determined by the neighbors of x_i in S to which it is connected.

Now we shall show that this polygon is actually a local minimum of $E_\Gamma^{(p)}$. We have two simple observations (i) Any curve that passes through x_i cannot have lower energy than two straight line segments passing through x_i (see figure 5). (ii) Any two line segments meeting at x_i have the same energy. (iii) An arbitrary curve can be replaced by straight lines with the same or lower energy. We shall show that if the curve is deformed a bit from x_i, then the energy will increase. Without loss of generality we perturb Γ to $\tilde{\Gamma}$ as in figures 5, 6, 7. We denote the weighted length of Γ by I_L and II_L. (By symmetry, we only need to consider the right hand part of each line). We have

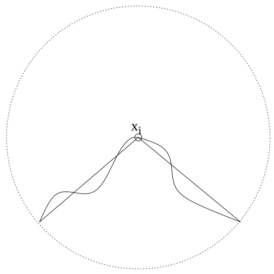


Figure 5:

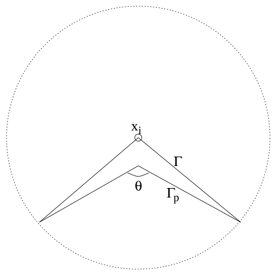


Figure 6:

$$\begin{aligned} I_L &= \int_0^{\epsilon \cot \frac{\theta}{2}} s^p ds = \epsilon^{p+1} \frac{(\cot \theta)^{p+1}}{p+1} \\ II_L &= \int_{\epsilon \cot \frac{\theta}{2}}^{\ell} s^p ds. \end{aligned}$$

The weighted length of the shifted line segment is

$$\begin{aligned} III_L &= \int_{\epsilon \cot \frac{\theta}{2}}^{\ell} d^p ds = \int_{\epsilon \cot \frac{\theta}{2}}^{\ell} (s^2 + \epsilon^2)^{\frac{p}{2}} ds \\ &= I_L + \int_{\epsilon \cot \frac{\theta}{2}}^{\ell} s^p ((1 + \frac{\epsilon^2}{s^2})^{\frac{p}{2}} - 1) ds \\ &\geq II_L + c(\ell, \theta)\epsilon^2 \text{ if } p > 1 \\ &\geq II_L + c(\ell, \theta)\epsilon^2 \ell_n \epsilon \quad \text{if } \ p = 1. \end{aligned}$$

Here the positive quantity c depends only on ℓ and θ.

We see that if $p \geq 1$, the slightly perturbed line segment has larger energy than the unperturbed segment. So this perturbation analysis proves Theorem 1. It also shows that, during the continuous deformation defined by equation (4), if a curve passes through a data point it will not leave that data points.

Next we obtain certain results about the contours of the distance function $d(\boldsymbol{x})$ in the continuous limit, i.e. when $d(\boldsymbol{x})$ is the distance function to a smooth surface. The zero level contour is the true surface. For simplicity of analysis, we modify $d(\boldsymbol{x})$ to be the signed distance function such that contours inside the region

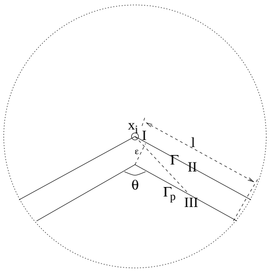


Figure 7:

have negative values and contours outside the region have positive values.

Let $\Gamma(a)$ correspond to the level contour $d(x) = a$ and let $L(a)$ be its length or surface area. Then

$$E_\Gamma^{(p)}(\Gamma(a)) = (\int_{\Gamma_a} d^p ds)^{\frac{1}{p}} = |a| L^{\frac{1}{p}}(a).$$

Let $F(a) = |a| L^{\frac{1}{p}}(a)$. Then $F(0) = 0$ is the minimum for $F(a)$. For $a \neq 0$,

$$sign(a) F'(a) = L^{\frac{1}{p}}(a) + \frac{a}{p} L^{\frac{1}{p}-1}(a) L'(a).$$

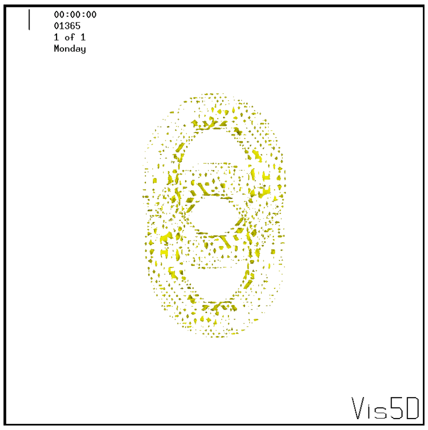
Suppose $\Gamma(d)$ is C^1. Then for small δa:

$$L(a + \delta a) = L(a) + \delta a \int_{\Gamma(a)} \kappa ds + O(\delta a^2)$$

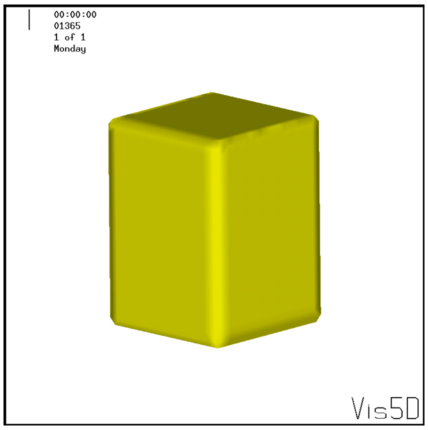
where κ is the mean curvature. Let $\chi = \int_{\Gamma(a)} \kappa ds$. (Note in 2D χ is 2π times the winding number of $\Gamma(a)$.) If $\Gamma(a)$ is diffeomorphic to Γ_0, then:

$$\begin{aligned} sign(a) F'(a) &= L^{\frac{1}{p}-1}(a)[L(a) + \tfrac{a}{p} L'(a)] \\ &= L^{\frac{1}{p}-1}(a)[L(0) + \chi a + \tfrac{a}{p}\chi] \\ &= L^{\frac{1}{p}-1}(a)[L(0) + \tfrac{p+1}{p}\chi a] \end{aligned}$$

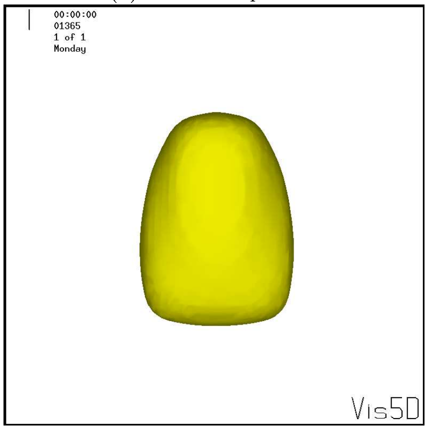
So we have shown that for a neighborhood proportional to $\frac{pL(0)}{(p+1)\chi}$ in which all contours $\Gamma(a)$ are diffeomorphic to Γ_0, $E_\Gamma^{(p)}(\Gamma(a))$ decreases as $|a| \to 0$. This strongly suggests Γ_0 is a local minimum of our energy for curves or surfaces which lie within substantial neighborhood of $\Gamma(0)$. The chief restriction seems to be that the neighboring contours should not change topology (although our method does handle many of these cases). We shall analyze the domain of attraction in future work and relate it to the local feature size described in [4].



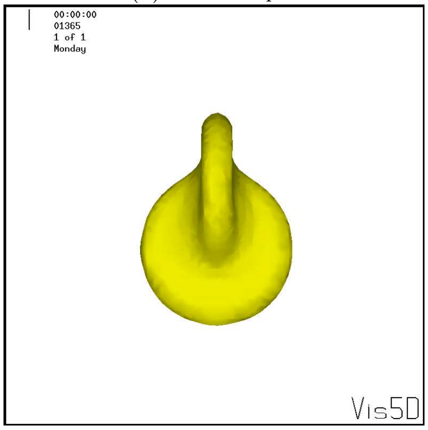
(a) 3200 data points



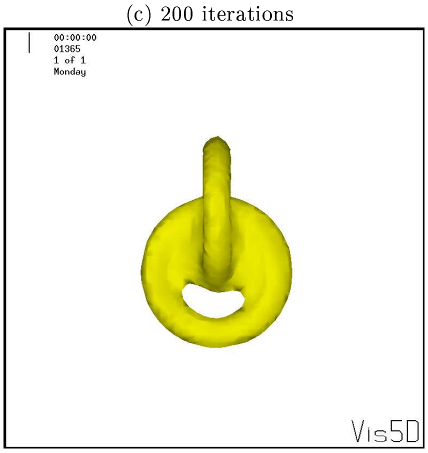
(b) initial shape



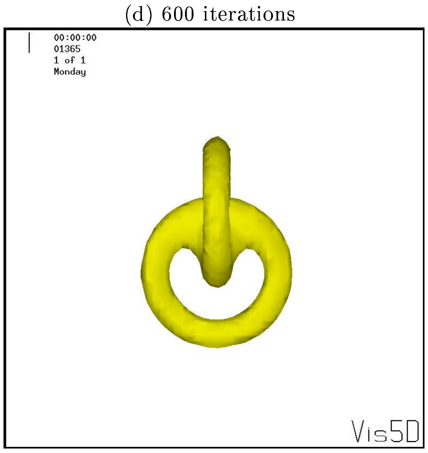
(c) 200 iterations



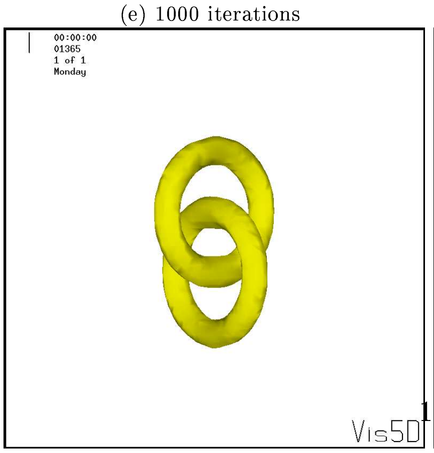
(d) 600 iterations



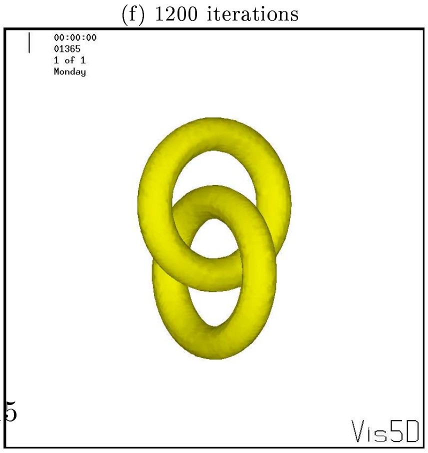
(e) 1000 iterations



(f) 1200 iterations

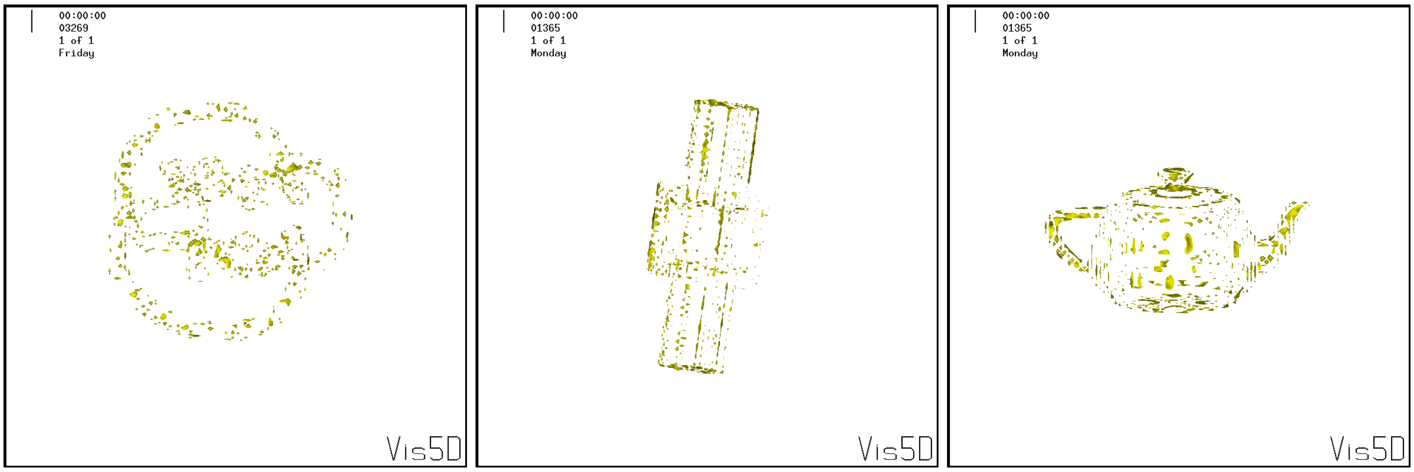


(g) reconstruction on a 39×31×31 grid



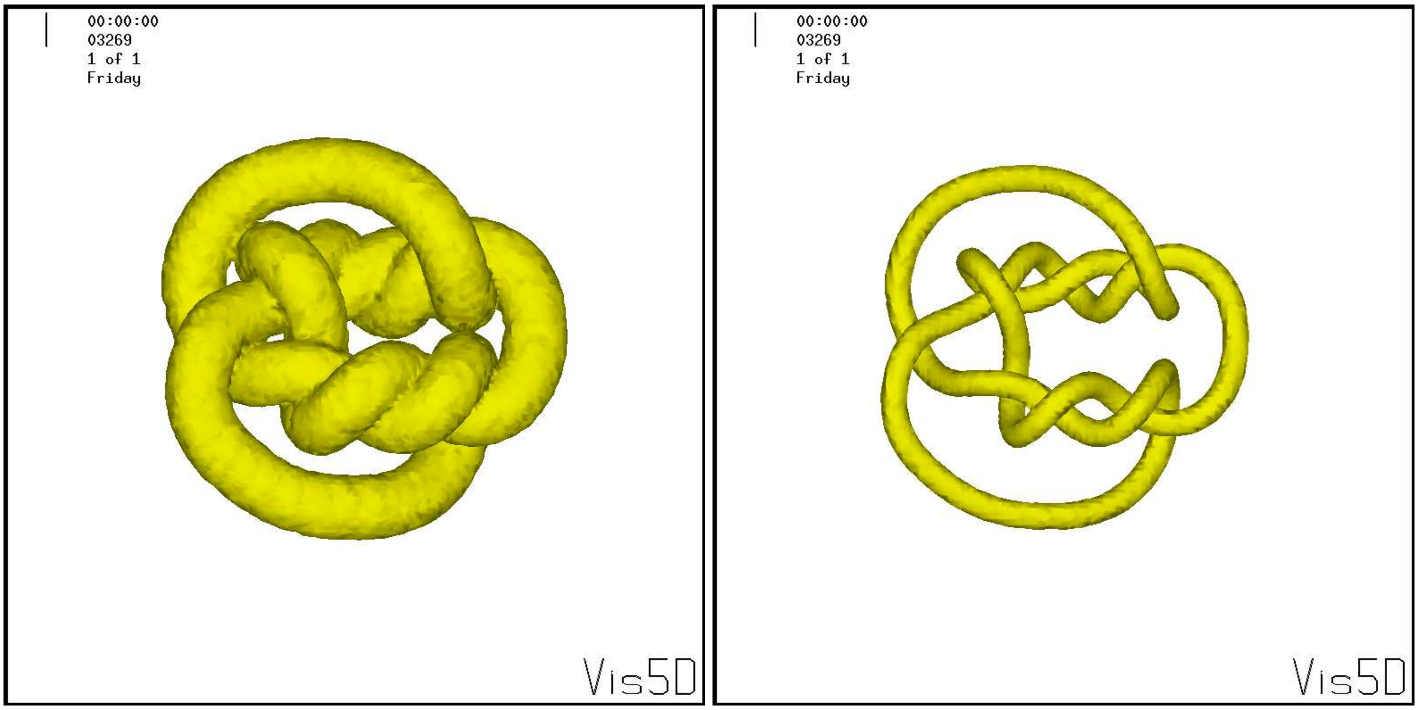
(b) reconstruction on a 80×60×60 grid

Figure 8:



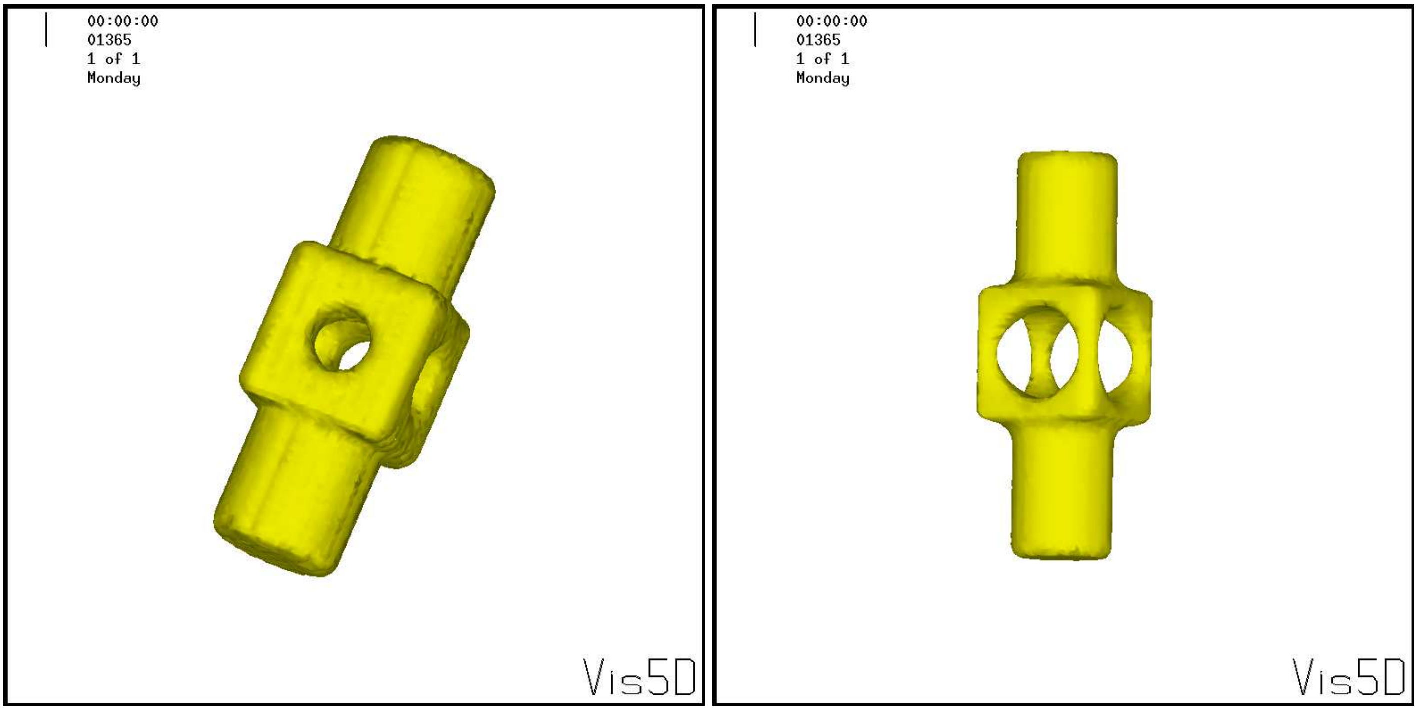
(a) 10000 data points for a knot (b) 4102 data points for a mechanical part (c) 26103 data points for a tea pot

Figure 9: initial data sets



(a) initial shape using an outer distance contour (b) reconstructed shape

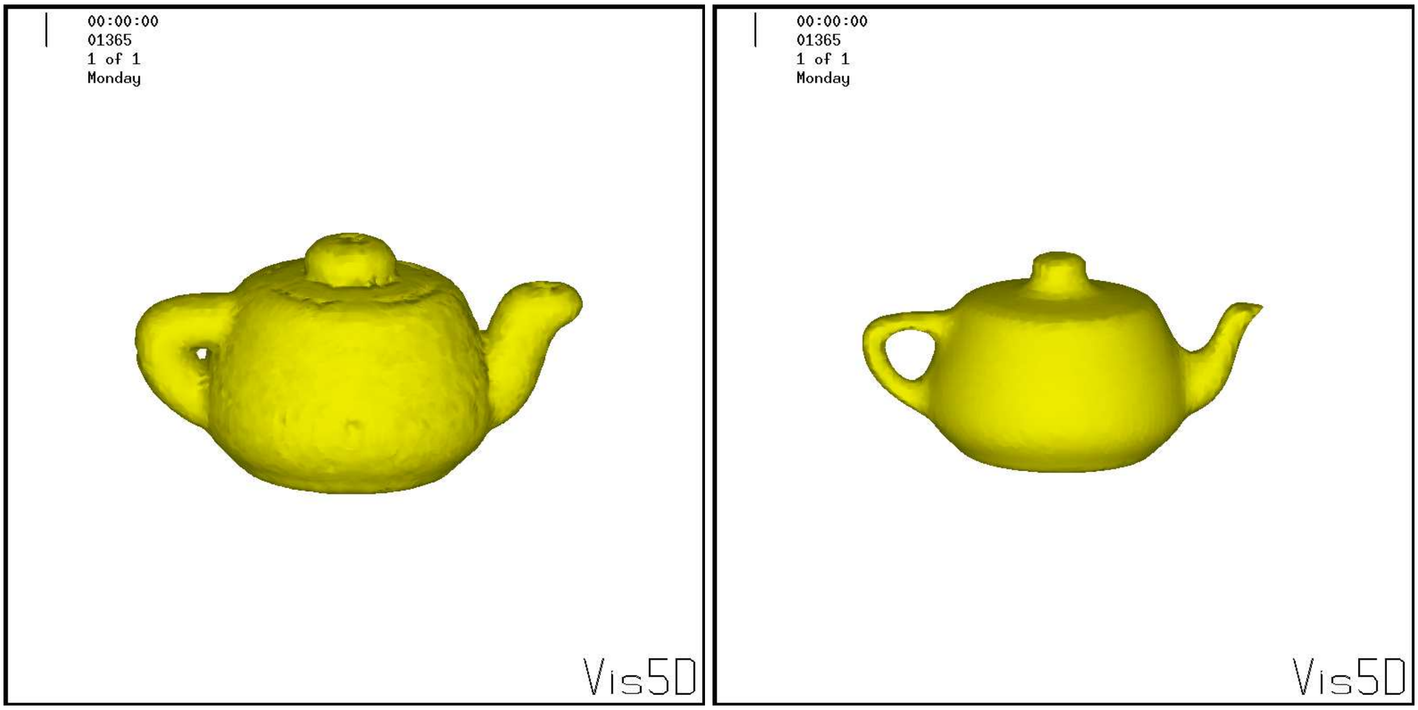
Figure 10: reconstruction of knot on a $80 \times 80 \times 80$ grid



(a) initial shape using an outer distance contour

(b) reconstructed shape

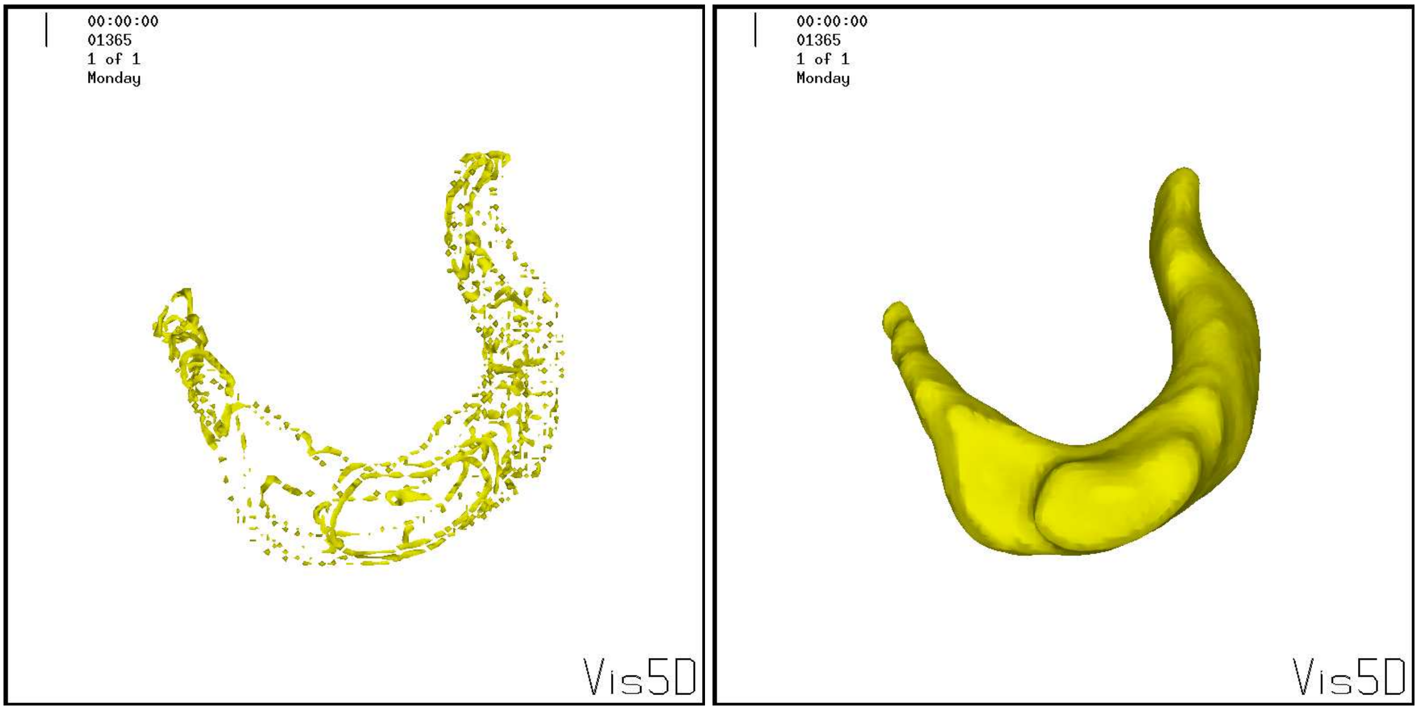
Figure 11: reconstruction of a mechanical part on a $33 \times 33 \times 80$ grid



(a) initial shape using an outer distance contour

(b) reconstructed shape

Figure 12: reconstruction of a teapot on a $79 \times 54 \times 45$ grid



(a) MRI slices of a rat brain with 1500 points (b) reconstructed shape

Figure 13: reconstruction of a rat brain on a $63 \times 62 \times 63$ grid