A NEW MULTISCALE REPRESENTATION FOR SHAPES AND ITS APPLICATION TO BLOOD VESSEL RECOVERY

BIN DONG *, AICHI CHIEN †, ZUOWEI SHEN ‡, AND STANLEY OSHER §

Abstract. In this paper, we will first introduce a novel multiscale representation (MSR) for shapes via level set motions and partial differential equations (PDEs). Based on the MSR, we will then design a surface inpainting algorithm to recover 3D geometry of blood vessels. Because of the nature of irregular morphology in vessels and organs, both phantom and real inpainting scenarios were tested using our new algorithm. Successful vessel recoveries are demonstrated with numerical estimation of the degree of arteriosclerosis and vessel occlusion.

Key words. Level set motion, mean curvature, multiscale representation, partial differential equations, surface inpainting.

AMS subject classifications. 92C55, 68U10, 65D18, 53C44.

1. Introduction.

1.1. Literature Reviews and Motivations of Multiscale Representations.

Multiscale representation (MSR) of functions, e.g. wavelets, has been extensively studied in the past twenty years [26, 55]. However, when one deals with shapes, e.g. biological shapes in $\mathbb{R}^3$, most of the classical theories and algorithms cannot be directly extended. In this paper, we will propose a new MSR for shapes based on PDEs and level set method. Although we shall focus on studying 3D biological shapes/surfaces, the MSR that we introduce here applies to general shapes/surfaces in both 2D and 3D.

Many attempts have been made in the past on designing wavelet-typed MSR for 3D shapes [59, 58, 41, 29]. Among them, the method proposed by Nain et. al. [59, 58] is especially effective to study biological shapes. They first map the shape (triangulated) onto the unit sphere so that one obtains a vector-valued function $f : \mathbb{S}^2 \mapsto \mathbb{R}^3$; then apply spherical wavelet decomposition [72] to each component of f. However, the wavelet coefficients are not intrinsic to the shape, but dependent on the mapping f. Furthermore, finding a good mapping from a shape to the unit sphere (or to some other canonical domains) is nontrivial and in fact a popular ongoing research area (see e.g. [43, 77, 42, 49, 47, 40, 44, 50, 74, 65]).

Another interesting approach was proposed by Pauly et. al. [64], where they introduced an MSR for point-based surfaces. Their idea was to use Moving Least Square method [52] to define a series of smoother and smoother point-based surfaces, and then define wavelet coefficients as the displacements from two successive levels. Their method only requires a local parametrization of the point-based surface which is easy to calculate. However, the application of their method is rather limited in medical image analysis, because most of the biological shapes are not point-based.

[*] Department of Mathematics, University of California, Los Angeles, CA (bdong@math.ucla.edu). This author's research is supported by NIH GRANT, P20 MH65166; NSF GRANT, DMS-0714807.

[†] Division of Interventional Neuroradiology, David Geffen School of Medicine at UCLA, 10833 LeConte Ave, Los Angeles, CA (aichi.seas@gmail.com). This author's research is supported by NSF GRANT, CCF-0830554.

[‡] Department of Mathematics, National University of Singapore, 2 Science Drive 2, Singapore, 117543 (matzuows@nus.edu.sg).

[§] Department of Mathematics, University of California, Los Angeles, CA (sjo@math.ucla.edu). This author's research is supported by NIH GRANT, P20 MH65166; NSF GRANT, DMS-0714807.

Motivated by Pauly et. al.'s work, we will propose a new MSR for shapes in Section 2. The basic idea is using level set motions via solving some properly chosen Hamilton-Jacobi (HJ) like equation to obtain a sequence of shapes that become smoother and smoother as time evolves (analogous to coarse level approximation in wavelet decomposition). Then we carefully define the so-called "details" (analogous to wavelet coefficients) of the MSR which carry important geometric information and facilitate a perfect reconstruction. While the wavelet based multiscale decomposition and reconstruction use filters, which are linear processes, the proposed new MSR for shapes uses (nonlinear) PDEs for both decomposition and reconstruction. However, the spirit is the same, i.e. separate features from smooth components of the surface and the underlying surface has a sparse approximation in feature domain together with the smooth components. Due to the level set formulation, parametrization is no longer needed.

1.2. Shape Modelling and Evolution PDEs. Throughout this paper, shapes are defined to be smooth boundaries of domains $\Omega \in \mathbb{R}^3$ and are represented by level set functions, typically signed distance functions. We note, however, that point-based and triangulated surfaces can also be handled in a similar way by associating the point-based/triangulated surface with a level set function (see item 4 of Remark 2.2 for details).

A level set function ϕ that represents the shape $\partial\Omega$ is defined as follows

$$\phi(x)\begin{cases} <0 \; x \in \Omega; \\ >0 \; x \in \Omega^c. \end{cases}$$

We always assume that the function ϕ is at least Lipschitz continuous.

Level set motions can be achieved by solving the following HJ like equation [63],

$$\phi_t + v_n(\nabla\phi)|\nabla\phi| = 0, \quad \phi(x,0) = \phi_0(x), \tag{1.1}$$

where we take $(x,t) \in \mathcal{D} \times [0,T]$ with $\mathcal{D}$ some bounded domain in $\mathbb{R}^3$ and $T>0$. Here $v_n(\nabla\phi)$ is the normal velocity, which essentially depends on $\nabla\phi$ while second order derivatives of ϕ may be involved (e.g. mean curvature). If v_n only depends on first order derivatives of ϕ, then (1.1) is a standard HJ equation. We also assume that the PDE (1.1) is geometric [20, 39], which guarantees contrasts invariance. Comprehensive theoretical analysis of PDE (1.1) and surface evolution equations can be found in [20, 39, 33, 34, 35, 36, 23, 24, 22].

The choice of velocity fields is very important and yet very non-unique. We need to choose one that generates a "meaningful" MSR for a given piecewise smooth shape. The bottom line is that we want the zero level set of $u(x,t)$ to become smoother and smoother as t increases. This is in fact a typical scale space behavior that has been studied for decades (see e.g. [1, 46]). It is known [1, 46] that under some general axiomatic hypothesis and some invariance (i.e. rotation and contrast invariance) assumptions on $\{u(x,t)\}_{t\geq 0}$, $u(x,t)$ must be a viscosity solution to a PDE of the form (1.1), with the velocity field v_n only depending on the principle curvatures of level sets of u at time t. In other words, a "meaningful" velocity field must be curvature dependent.

The type of velocity fields that we shall focus in this paper is

$$v_n = c + \alpha\kappa_a - \beta\kappa, \quad c,\alpha \in \mathbb{R}, \quad \beta > 0, \tag{1.2}$$

where κ is the mean curvature defined as $\kappa := \nabla\cdot\frac{\nabla\phi}{|\nabla\phi|}$, and κ_a is the average mean curvature [31]. Note that when $c=0$ and $\alpha=\beta=1$, i.e. $v_n=\kappa_a-\kappa$, the PDE (1.1) generates a volume preserving mean curvature motion [31, 12, 71, 67].

2. Level Set Based MSR of Shapes: Continuous Transforms and Discrete Algorithms.

Let $\Omega_t\in\mathbb{R}^3$ be some domain with scale t, and $S_t := \partial\Omega_t$ be the shape at scale t represented by some time-dependent level set function $\phi(x,t)$, i.e. $\phi(x,t)<0$ for $x\in\Omega_t$, $\phi(x,t)>0$ for $x\in\Omega_t^c$, and

$$S_t=\{x\in\mathbb{R}^3 \mid \phi(x,t)=0\}_{t\geq 0}. \tag{2.1}$$

Here S_0 denotes the original shape with the corresponding level set function $\phi_0(x)=\phi(x,0)$. Throughout the rest of the paper, the function $\phi(x,t)$ is always taken to be the solution of (1.1). For some properly chosen v_n in (1.1), e.g. with $v_n=-\kappa$ or $\kappa_a-\kappa$, we can obtain a continuous series of shapes $\{S_t\}_{t\in[0,T]}$, which tends to become smoother when t increases. Based on this, we define our continuous level set based MSR of S_0 as follows.

Definition 2.1. *Let $\phi(x,t)$ be the solution of the PDE (1.1) and $(x,t)\in\mathcal{D}\times[0,T]$. We now understand $x_l(t)$ as a path on the propagating l-th level set of ϕ, i.e. $\phi(x_l(t),t)=l$. For simplicity, we shall omit the subscript "l" unless a particular level set is considered.*

1. *We now define the* ***multiscale transformation (MST)*** *of $\phi_0(x)$ as*

$$\vec{W}(x,t) := W(\phi_0) := -v_n\frac{\nabla\phi}{|\nabla\phi|} = -x'(t). \tag{2.2}$$

   Vector $-x'(t)$ is the ***displacement vector*** *and $w(x,t) := -v_n(x,t)$ is the* ***detail*** *of the MST.*

2. *We shall call $\vec{W}(x,t)$ the* ***displacement vector field*** *at scale t, and denote $\vec{W}_|(x,t)$ $(w_|(x,t))$ as the restriction of $\vec{W}(x,t)$ $(w(x,t))$ on S_t.*
3. *The MSR for the original shape S_0 in terms of $\phi_0(x)$ is denoted as*

$$MSR(\phi_0,T)=\left\{\left\{\vec{W}(x,t)\right\}_{t\in(0,T)},\phi(x,T)\right\}.$$

4. *We define the* ***inverse multiscale transformation (IMST)*** *via solving the following PDE*

$$\psi_\tau+\vec{W}(x,T-\tau)\cdot\nabla\psi=0, \quad \psi(x,0)=\phi(x,T). \tag{2.3}$$

   for given $T>0$ and $0\leq\tau\leq T$.

Remark 2.2.

1. *The technique of generating a sequence of the spaces $\{S_t\}$ via solving PDEs is known as scale space decomposition (see e.g. [1, 46]). However, a classical scale space analysis does not study the details as defined in item 2 above, and does not have a reconstruction as in (2.3).*
2. *The last identity in (2.2) can be easily shown by using PDE (1.1) and the assumption that $x'(t)$ is aligned with normal directions of level sets of ϕ.*
3. *The detail $w_|(x,t)$ is a function on S_t that characterizes intrinsic geometric information of the shape at scale t. Here by intrinsic we mean that $w_|(x,t)$, as well as $\{S_t\}_{t>0}$, does not depend on the initial embedding ϕ_0 for a large*

class of functions [39], but only depends on S_0. Therefore, we now have an intrinsic MSR for S_0:

$$MSR(S_0)=\Big\{\{\vec{W}_|(x,t)\}_{t\in(0,T)},S_T\Big\}. \tag{2.4}$$

Furthermore, the above MSR is invariant under translation and rotation of S_0.

4. *The MSR defined above can be easily adapted to a point-based or triangulated surface. One simply need to first associate the surface with a level set function and then perform the MST. For point-based surfaces, the IMST from its MSR* (2.4) *can be point-wise defined as $S_0=S_T+\int_0^T\vec{W}_|(x,t)dt$ or equivalently $x_0(0)=x_0(T)+\int_0^T -x_0'(t)dt$, which is obviously true.*

Now the question is that if we have perfect reconstructions via (2.3). The answer is given in the following proposition, which directly follows from theories of ODEs.

Proposition 2.3. *Assume that $\vec{W}(x,t)$ stays Lipschitz continuous for $(x,t)\in\mathcal{D}\times[0,T]$. Then the equation* (2.3) ***inverts*** *the MST defined by* (2.2) *in the sense that $\psi(x,\tau):=\phi(x,T-\tau)$ is the unique solution of* (2.3).

Remark 2.4.

1. *The assumption in Proposition 2.3, i.e. $\vec{W}(x,t)$ being Lipschitz in $\mathcal{D}\times[0,T]$ for some $T>0$, is not always valid (e.g. $v_n=c<0$ and $\phi_0(x)$ representing a cube). However, if we choose v_n as in* (1.2) *and choose some appropriate ending time $T>0$ (e.g. before any topological changes occur), the above assumption will be valid and we will have a perfect reconstruction using* (2.3) *[39, 31].*
2. *Generally speaking, the vector field $\vec{W}(x,t)$ does not stay Lipschitz globally in time, and this happens when the corresponding surface evolution starts to develop singularities. It is very difficult to find a mean curvature dependent surface evolution that guarantees to have global smooth solutions for a general initial surface S_0, and the evolution is also invariant under translations and rotations of S_0. For some special class of initial surfaces, however, it is relatively easy to find such motion. Taking $v_n=\kappa_a-\kappa$ for example, it is shown in [31] that if the initial surface S_0 is close enough (but not necessarily convex) to a certain sphere, then S_t stays smooth and converges exponentially fast to the sphere.*

Notice from Definition 2.1 and Proposition 2.3 that to perfectly reconstruct $\phi_0(x)$ from $\phi(x,T)$, we need to store the entire vector field $\vec{W}(x,t)$ for every $x\in\mathcal{D}$ and all scale t. However, in practice, we only want a perfect reconstruction of S_0, and thus we do not need that much information. Therefore, only the displacement vectors within a narrow band of the zero level set of $\phi(x,t)$ need to be stored.

We can be even more "greedy" here by only storing $\vec{W}_|(x,t)$. When performing inverse transform, we will need to extend $\vec{W}_|(x,t)$ to at least a narrow band of the zero level set of $\phi(x,t)$. Note that no extension can guarantee an exact recovery of the vector field $\vec{W}(x,t)$, and hence the reconstruction of S_0 will not be exact. However, if the extension is conducted accurately and the mesh grid is dense enough, i.e. the resolution of the shape is high enough, the reconstruction should be more and more accurate. The extension we shall adopt here is such that the extended vectors are constant in the normal directions of each level set of $\phi(x,t)$ [7]. For simplicity, we will use a local search method to extend $\vec{W}_|(x,t)$ to a narrow band of the zero level set of $\phi(x,t)$. We further note that if we perform the extension of $\vec{W}_|(x,t)$ more carefully, we may achieve smaller reconstruction errors than those shown in Figure 3.5. When more accurate reconstruction is desired and space of storage is not an issue, one can be less "greedy" on saving the storage and store $\vec{W}(x,t)$ within a narrow band of $\phi(x,t)$ for each step. Then we will not have errors due to the vector field extensions. However, we will still have errors introduced by solving (2.3) numerically.

Our proposed discrete version of MSR is given in Algorithm 1.

Algorithm 1 Level Set Based MST and IMST

```
Start from the given level set function $\phi_0(x)$ representing shape $S_0$. Choose time steps $0=t_0<t_1<\ldots<t_N=T$. Denote $\Delta t_i=t_{i+1}-t_i$ and assume that $\max_i(\Delta t_i)$ is small.
Initialize: Sample a point set $X_0$ from $S_0$ (either uniformly or non-uniformly).
                                MST:
while $i\le N$ do
  1. Starting from $\phi(x,t_{i-1})$, solve PDE (1.1) for $t\in[t_{i-1},t_i]$ and obtain $\phi(x,t_i)$.
  2. Orthogonally project $X_{i-1}$ onto the zero level set of $\phi(x,t_i)$ and obtain $X_i$.
  3. Compute the discrete displacement vector by $\vec{W}_{|i}=\frac{X_i-X_{i-1}}{\Delta t_{i-1}}$, and $i\leftarrow i+1$.
end while
We then obtain the discrete MSR of $S_0$: $\mathrm{MSR}(S_0,T):=\{\vec{W}_{|1},\vec{W}_{|2},\ldots,\vec{W}_{|N},\phi(x,T)\}$.
                                IMST:
1. Extend the vector fields $\{\vec{W}_{|i}\}_{i=1}^N$ such that the values are constant along normal directions of the level sets of $\phi(x,t_i)$ (see Figure 2.1).
2. Solve (2.3) using $\vec{W}_{|i}$ within interval $[t_i,t_{i-1}]$ iteratively for each $i$.
```

3. Numerical Experiments on the MSR.

One of the key steps of implementing Algorithm 1 is to solve the evolution PDE (1.1) efficiently. There are many ways of solving equation (1.1). The most straightforward way is to use monotone finite difference schemes [63, 62]. However, it is not very efficient computationally. To overcome this, Merriman, Bence and Osher introduced a diffusion-based level set motion in [57, 56], and it was further studied in [48, 68, 69, 70], where in [48] the correctness of the method is rigorously proven. In [71], Ruuth and Wetton introduced a fast algorithm to calculate volume preserving motion by mean curvatures. All these methods speed up curvature driven motions drastically.

In this section, we will recall the fast algorithms of level set motion for the cases $v_n=c$ and $v_n=\kappa_a-\kappa$ given by [71, 57, 56, 69]. These algorithms will be used later to generate fast multiscle decompositions of shapes.

We first recall the fast method of solving (1.1) with $v_n=c$ (see [57, 56, 69]) in Algorithm 2.

We now recall the fast implementation of (1.1) with $v_n=\kappa_a-\kappa$ proposed by Ruuth and Wetton [71] in Algorithm 3, which is based on the diffusion-based mean curvature motion proposed by [57, 56]. Note that if we remove step 3 in Algorithm 3 and choose $\lambda=0.5$ in step 4, it is exact the fast mean curvature motion proposed in [57, 56].

Some numerical results of the MST and IMST in Algorithm 1 are presented in Figure 3.1 and 3.3 using two biological shapes (right hemisphere of a cortex and part of a blood vessel). The velocity field in (1.1) is chosen to be $v_n=\kappa_a-\kappa$ and 5 levels of decomposition are conducted (first and second row of Figure 3.1 and 3.3). The mesh size is taken to be 1 throughout the computations. Details $\vec{W}_{|i}$ are drawn on

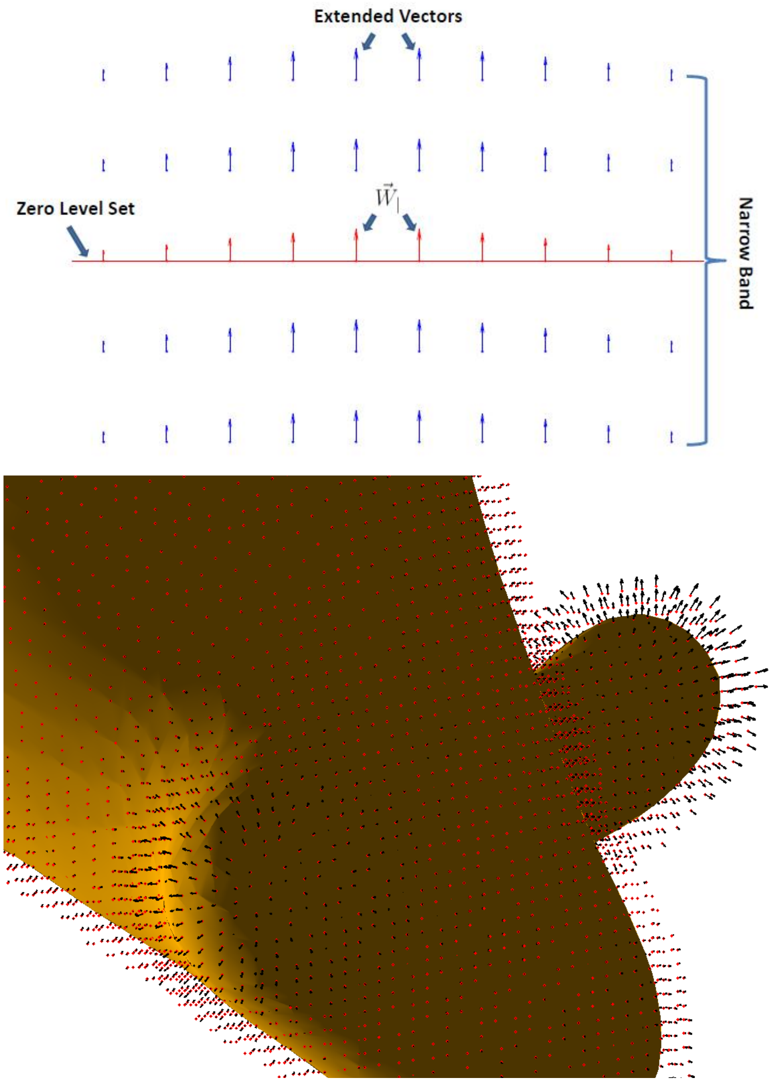

FIG. 2.1. *Top figure illustrates the vector field extension, where red line is the zero level set; red vectors are $\vec{W}_|$; and blue vectors are the extended vectors from $\vec{W}_|$. The bottom figure shows the actual extension of $\vec{W}_|(x,t)$.*

the surface S_i (second row of Figure 3.1 and 3.3), where the value is positive, when $\vec{W}_{|i}$ is pointing outwards and negative when it is pointing inwards. The IMSTs are also presented where $\tilde{S}_i$ denotes the reconstruction of level i from level $i+1$. As we can see, although the reconstructions are not exact for each level, they are quite accurate in the sense that most of the features are well reconstructed. We quantified the reconstruction errors in terms of Hausdorff distance in Figure 3.1 and 3.3. Note that the reconstruction errors are dominant by some of the regions with relatively high curvatures as shown in Figure 3.5.

The data of cortical surface in Figure 3.1 is of size $45 \times 95 \times 77$. The computation time for each level of decomposition and reconstruction are 35 seconds and 180 seconds

Algorithm 2 Level Set Motion with Constant Normal Velocity

```
Start from a given shape represented by $\phi$.
while $t<T$ do
    1. Define the corresponding characteristic function by $\chi=\mathbf{1}_{\{\phi<0\}}$. Set $V_0$ equal to the volume of $\{\phi<0\}$.
    2. Starting from $\chi$, evolve $\bar{\chi}$ for a time $\Delta t$ by $\bar{\chi}_t=\nabla^2\bar{\chi}$.
    3. Sharpen:
       $$\chi=\begin{cases}1 \text{ if } \bar{\chi}>0\\ 0 \text{ otherwise}\end{cases}$$
    4. Let $t\leftarrow t+\Delta t$. Compute $\phi(x,t)$ from $\chi$ via fast sweeping method [75].
end while
```

Algorithm 3 Volume Preserving Mean Curvature Motion: $v_n=\kappa_a-\kappa$.

```
Start from a given shape represented by $\phi$.
while $t<T$ do
    1. Define the corresponding characteristic function by $\chi=\mathbf{1}_{\{\phi<0\}}$. Set $V_0$ equal to the volume of $\{\phi<0\}$.
    2. Starting from $\chi$, evolve $\bar{\chi}$ for a time $\Delta t$ by $\bar{\chi}_t=\nabla^2\bar{\chi}$.
    3. Determine the threshold value that preserves the volume of the set: i.e. find a $0<\lambda<1$ s.t.
       $$\Big||\{x:\bar{\chi}<\lambda\}|-V_0\Big|<\varepsilon.$$
    4. Sharpen:
       $$\chi=\begin{cases}1 \text{ if } \bar{\chi}>\lambda\\ 0 \text{ otherwise}\end{cases}$$
    5. Let $t\leftarrow t+\Delta t$. Compute $\phi(x,t)$ from $\chi$ via fast sweeping method [75].
end while
```

respectively. The data of blood vessel in Figure 3.3 is of size $58\times50\times50$. The computation time for each level of decomposition and reconstruction are 15 seconds and 40 seconds respectively. The major computation cost for reconstruction is in vector field extension which can be improved by, e.g. solving an extension PDE [7]. All calculations are performed on a Windows laptop with Intel Core 2 Duo T9400 and 3G memory.

We also illustrate sparseness of the coefficients $\{\vec{W}_{|i}\}_{i=1}^5$ in Figure 3.2 for the cortical surface and in Figure 3.4 for the blood vessel. As one can see that the energy of $\vec{W}_{|i}$ are relatively concentrated around 0, especially for the later levels. This sparseness reduces the computational costs.

4. Application in Blood Vessel Recovery. Evaluating missing parts in medical images provides important information as signs of diseases. One of the most common situation is the phenomenon of vessel narrowing or occlusion in angiographic images. Estimating and quantifying these abnormalities can help document disease progression.

The recovery of blood vessels can be regarded as a surface inpainting problem

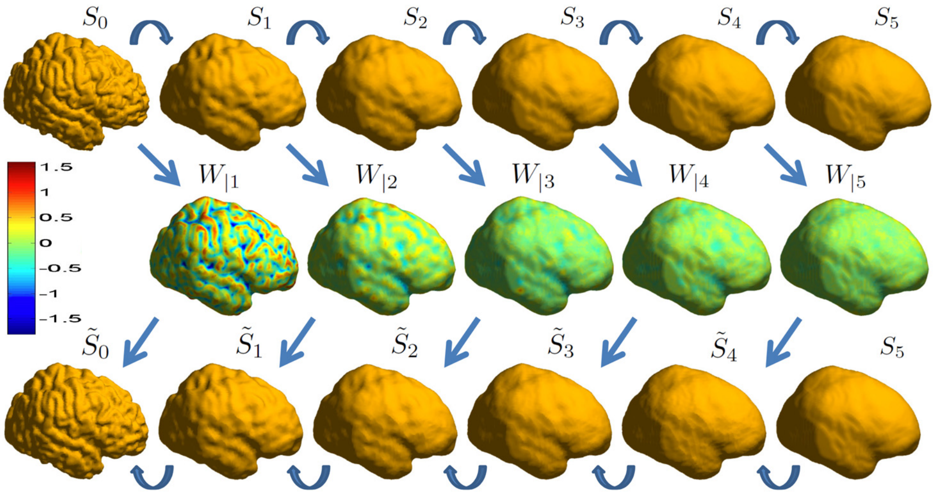

FIG. 3.1. *First row (left to right): MST $S_0, S_1, \ldots, S_5$. Second row shows the details of MSR on $S_1, \ldots, S_5$. Third row shows IMST $\tilde{S}_i$, $i = 0, 1, \ldots, 4$, where the Hausdorff distance between S_i and $\tilde{S}_i$ are:* 1.12, 0.74, 0.74, 0.69, *and* 0.63 *respectively.*

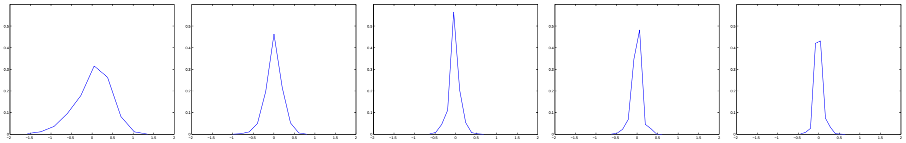

FIG. 3.2. *Histograms of $\vec{W}_{|i}$ for $i = 1, \ldots, 5$ (left to right). The supports of them are (from left to right): [-1.57, 1.35], [-1.00, 0.81], [-0.63, 0.68], [-0.63, 0.61] and [-0.44, 0.65].*

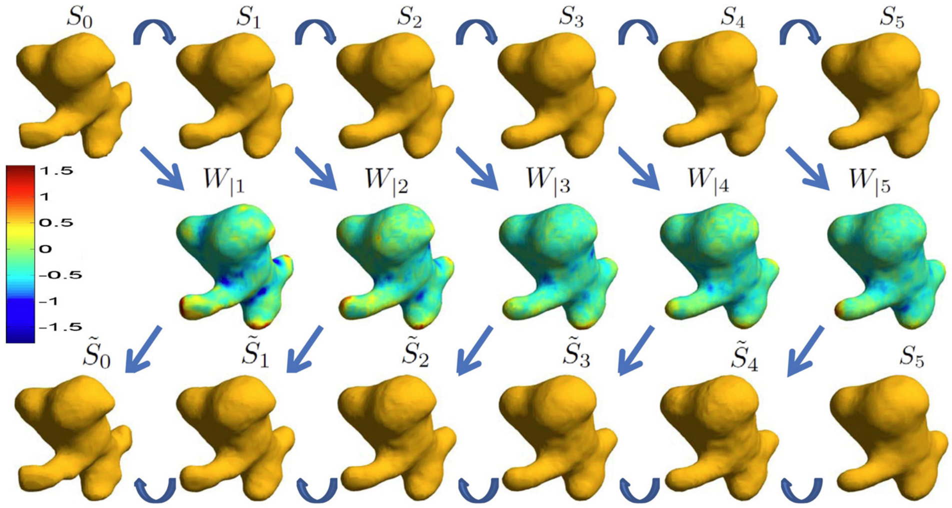

FIG. 3.3. *First row (left to right): MST $S_0, S_1, \ldots, S_5$. Second row shows the details of MSR on $S_1, \ldots, S_5$. Third row shows IMST $\tilde{S}_i$, $i = 0, 1, \ldots, 4$, where the Hausdorff distance between S_i and $\tilde{S}_i$ are:* 0.81, 0.71, 0.77, 0.71, *and* 0.62 *respectively.*

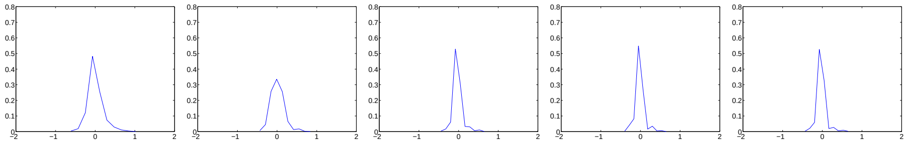

Fig. 3.4. *Histograms of $\vec{W}_{|i}$ for $i=1,\ldots,5$ (left to right). The supports of them are (from left to right): [-0.61, 1.02], [-0.43, 0.84], [-0.44, 0.65], [-0.40, 0.65] and [-0.44, 0.65].*

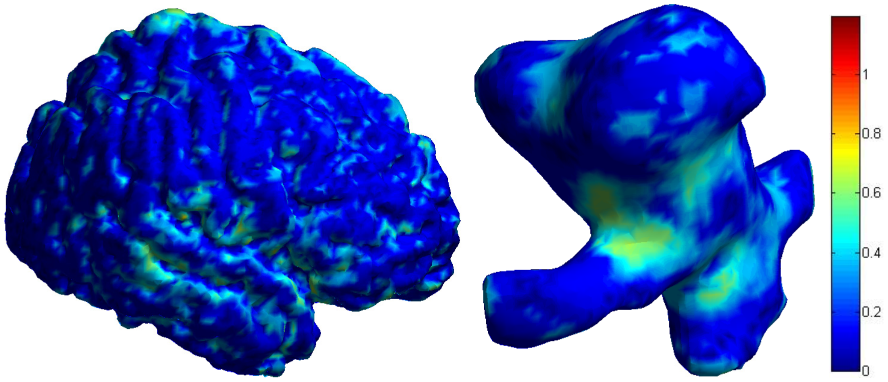

Fig. 3.5. *Reconstruction errors visualized on the original cortical surface and blood vessel.*

[27, 76, 3]. Inpainting problems, for both images and surfaces, have been extensively studied in the literature [60, 17, 8, 6, 18, 11, 10, 32, 28, 9, 14, 15, 30, 19, 27, 76, 3, 2, 5, 79, 78, 81]. They occur when part of the data in an image or regions of a surface is missing or corrupted. The major task of inpainting is to fill in the missing information based on the geometry of the image/surface. In this section, we will propose a new surface inpainting algorithm for blood vessel reconstruction that arises in medical image analysis.

Our surface inpainting algorithm (Algorithm 4 below) inherits the structure of the following framelet-based image inpainting algorithm proposed by Cai et. al. [15]:

1. Take framelet transform of the given image;
2. Truncate the framelet coefficients via soft-thresholding and reconstruct;
3. Apply the exact data outside the inpainting domains, and repeat.

Since we already have an MSR for surfaces, the first step above can be replace by our MST. For the second step, we shall solve the following PDE for IMST instead of the PDE (2.3) that was originally proposed in Definition 2.1:

$$\psi_\tau + \vec{W}(x,T-\tau)\cdot\nabla\psi = \varepsilon\nabla^2\psi, \quad \psi(x,0)=\phi(x,T). \tag{4.1}$$

The above PDE mimics thresholding in the sense that it penalizes the reconstruction from $\vec{W}$ by introducing a vanishing viscosity $\varepsilon\nabla^2\psi$, which forces some information outside the inpainting region flows into the inpainting regions. Also, when $\varepsilon\to 0$, the solution of (4.1) converges to the viscosity solution of (2.3) [24, 22].

Since we generally expect volumes of surfaces to increase during inpainting, we choose the following PDE for the MST,

$$\phi_t + (c+\kappa_a-\kappa)|\nabla\phi| = 0, \quad \phi(x,0)=\phi_0(x), \quad c>0. \tag{4.2}$$

Note that the PDE (4.2) generates a mean curvature motion with increasing volumes of the domains enclosed by level sets of $\phi(x,t)$. The constant c can be regarded as a parameter that needs to be adjusted according to different surface inpainting scenarios. In our experiments, we solve PDE (4.2) efficiently via a combination of Algorithm 2 and Algorithm 3 recalled in Section 3

Algorithm 4 Surface Inpainting via MSR

```
Start from $\phi_0$, with inpainting region $D$. Choose some $\varepsilon > 0$.
while "Not converge" do
    1. Perform discrete MST by solving (4.2) and acquire $\vec{W}_{|i}$ by Algorithm 1.
    2. Perform IMST by solving (4.1) and obtain $\psi_\varepsilon$.
    3. Copy the known information to $\psi_\varepsilon$: $\psi_\varepsilon|_{D^c} \leftarrow \psi_0|_{D^c}$.
    4. Decrease amount of smoothing: $\varepsilon \searrow$.
end while
```

We test Algorithm 4 on both phantom (first two vessels in Figure 4.1) and real (last two vessels in Figure 4.1) surface inpainting scenarios. First row of Figure 4.1 shows four blood vessels with inpainting regions specified by red circles. For the two phantom inpainting scenarios, the inpainting regions are created manually, and the surface within those regions were chopped off. For the two real inpainting scenarios, we do not know the exact inpainting regions. Therefore in practice, we adopt a user interactive strategy to determine the inpainting regions. After several points have been selected on the surface, the inpainting regions are then generated automatically. Inpainting results are given in second and third row of Figure 4.1. Throughout the computations, the mesh size is again chosen to be 1. Decomposition levels are chosen to be 2 for the first phantom vessel and 1 for all the rest.

We want to point out that during the inpainting process, topological change may occur for some cases (e.g. second vessel in Figure 4.1). Although it violates the assumption in Proposition 2.3, topological change is still allowed for inpainting problems. The reason is that perfect reconstruction is only required at the very last stage of inpainting (i.e. when $\varepsilon \approx 0$) in order to ensure convergence, while topological changes most likely occur during the middle of the process if the parameters (e.g. c in (4.2)) are properly chosen.

Generally speaking, the constant c in (4.2) is related to the volume of the regions that needed to be inpainted. The larger is c, the larger is the volume that will be filled into the inpainting region (see Figure 4.2 as an example).

5. Other Applications of MSR for Shapes. In this section, we will briefly discuss some other applications of the MSR in shape processing and analysis, namely shape registration, mapping and classification.

5.1. Shape Registration. A frequent occurring problem in practice is to compare shapes that are acquired at different time, by different imaging systems or from different viewpoints. Shapes need to be aligned with one another so that intrinsic differences can be detected. There is a vast amount of research in shape, as well as image registration problems. We refer readers to the following survey papers [13, 54, 51, 16, 82, 4, 37]. In particular, a survey on hierarchical medical image registrations is given in [51], which is related to what we are about to discuss here.

One major difficulty in aligning shapes is the alignment of their detailed features (e.g. gyri and sulci of cortical surfaces). The general idea of MSR based registration is to register the smooth approximations of the shapes first, which is normally easier

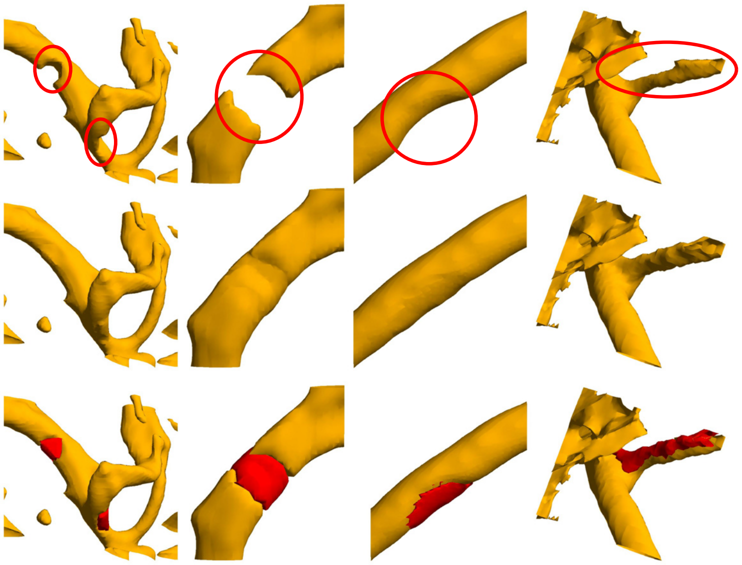

Fig. 4.1. *Blood vessel inpainting. Row 1: vessels before inpainting; row 2: vessels after inpainting; row 3: inpainted regions shown in red. The errors in terms of Hausdorff distance for the phantom cases are* 2.71 *and* 2.86 *respectively. The percentages of the volume of inpainted region over that of the entire shape are:* 5.3%, 19.2%, 6.7% *and* 5.7%.

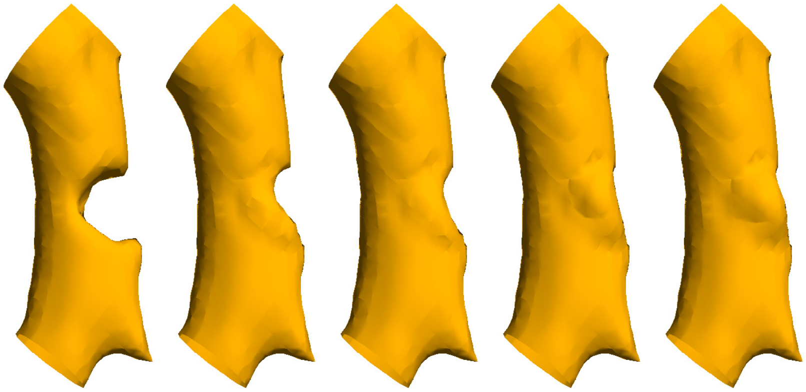

Fig. 4.2. *First figure shows the shape before inpainting. From the second to the last figure: results after inpainting using* $c = 0.1, 0.2, 0.3$ *and* 0.4.

to register than the original shape, and then progressively move back to the original shape.

Now let us be more specific. Denote S_0 and $\tilde{S}_0$ be the target and template shape. We need to register S_0 according to $\tilde{S}_0$. For simplicity, the MSR we consider here is taken to be discrete in time and continuous in space. We first obtain the MSRs for the two shapes denoted as $\left\{\{\vec{W}_{|i}\}_{i=1,2,\cdots,N}, S_N\right\}$ and $\left\{\{\vec{\tilde{W}}_{|i}\}_{i=1,2,\cdots,N}, \tilde{S}_N\right\}$. We then register S_N to $\tilde{S}_N$ and obtain a registered version of S_N denoted as S_N^r. Through the registration, the details $\vec{W}_{|N}$ is naturally inherited by S^r_N and then we can reconstruct S^r_{N-1} from S^r_N. At level $N-1$, S^r_{N-1} may or may not be well registered to $\tilde{S}_{N-1}$. If not, then we can perform another registration process using S^r_{N-1} as a potentially good initial guess. This process can be repeated until we are back to level 0, and a registration of S_0 to $\tilde{S}_0$ (i.e. S^r_0) is then obtained. Note that the registration process at each level except level N can be done efficiently, because good initial guesses are used and hence only slight adjustments are needed. The registration at level N can also be done efficiently because S_N and $\tilde{S}_N$ contain only large scale features and thus easy to register.

5.2. Shape Mapping. Shape mapping is an important problem in both medical shape analysis and computer aided designs (CAD). It can also be interpreted as a shape parametrization problem. Indeed, for a given shape, if we can find a good one-to-one mapping (e.g. a diffeomorphism) of it to some other shape with a certain good parametrization, then this parametrization can be inherited to the original shape.

Shape mapping has many applications in various fields of science and engineering, including reparameterization of surfaces, repair of CAD models, and texture mapping which is used in computer graphics to enhance the visual quality of polygonal models. We refer readers to the following survey papers and book for recent developments on this topic [38, 73, 45].

The idea of using MSR for shape mapping is very similar to that of registration. To map two shapes S_0 and $\tilde{S}_0$, we start with finding a mapping $\mathcal{M}_N$ between S_N and $\tilde{S}_N$, which is usually easy to find. Here S_N and $\tilde{S}_N$ are smooth approximations to S_0 and $\tilde{S}_0$ as defined in the previous section. Then based on the details $\vec{W}_{|N}$ and $\vec{\tilde{W}}_{|N}$, one can obtain the mapping $\mathcal{M}_{N-1}$ between S_{N-1} and $\tilde{S}_{N-1}$, which can be further updated if necessary. Repeat this process and finally we can get the desired mapping $\mathcal{M}_0$ between S_0 and $\tilde{S}_0$. Note that if the initial mapping $\mathcal{M}_N$ is homeomorphic and the underlying surface evolution equation of the MST does not develop singularities, then the final mapping $\mathcal{M}_0$ is automatically homeomorphic even if we do not update $\mathcal{M}_i$ for each level i.

5.3. Shape Classification. The major task for shape classification is to group shapes according to their geometric similarities [53, 80, 25]. For this purpose, important geometric features of the shapes need to be extracted and used properly. Therefore, MSR comes in naturally for this task, because we have features of the shape stored in $\{\vec{W}_{|i}\}$ in a multiscale fashion.

For example, if we define $\|S_i\|_p^p := \int_{x\in S_i} |\vec{W}_{|i}(x)\cdot\vec{n}_{S_i}(x)|^p ds$, where n_{S_i} is the outer normal vector of S_i, then for each shape S_0 we obtain a multiscale feature vector $(\|S_i\|_p)_i$. Then for a group of shapes, we can use their multiscale feature vectors to classify them by multidimensional scaling [21] for example. We also note that the quantity $\|S_0\|_p := \sum_{i=1}^{N} \|S_i\|_p$ can be regarded as a measure of geometric complexity of the shape S_0.

The idea of using feature vector $(\|S_i\|_p)_i$ is motivated by the recently proposed concept "shape DNA" [66, 61]. In [66, 61], the authors regarded eigenvalues (in ascending order) of Laplace-Beltrami operator for a given shape as its "DNA". The shape DNA is also a multiscale feature vector in the sense that small eigenvalues correspond to low frequency eigenfunctions and large eigenvalues correspond to high frequency eigenfunctions.

6. Conclusion and Discussion. In this paper, we introduced a novel multiscale representation (MSR) for shapes which is intrinsic to the shape itself, does not need any parametrization, and the details of the MSR reveals important geometric information. Based on the MSR, we then proposed a surface inpainting algorithm and applied it to recover corrupted blood vessels. This technique is especially useful to study arteriosclerosis and vessel occlusions. Numerical results showed that the inpainting regions were nicely filled in according to the neighboring geometry of the vessels and allowed us to accurately estimate the volume loss of vessels. We also briefly described the possible application of the MSR to shape registration, mapping and classification.

There are still many interesting aspects of both the MSR itself and its applications worth discovering. For example, a rigorous analysis of how Algorithm 1 approximates the continuous version in Definition 2.1 needs to be done. Another future work is to carry out the ideas explained in Section 5 in practice.

We further note that the MSR using the v_n in (1.2) is not ideally sparse (as shown in Figure 3.1 and 3.3, especially for earlier levels). This raises the question that what kind of evolution PDE will produce a sparse MSR for piecewise smooth shapes? Generally speaking, mean curvature motion ($v_n = -\kappa$) generates a sparse MSR for piecewise flat surfaces, while volume preserving mean curvature motion ($v_n = \kappa_a - \kappa$) produces a sparse MSR for surfaces that are close to spheres. It will be interesting to find a surface evolution such that the corresponding MSR is sparse for all piecewise smooth shapes or a subclass of them.

REFERENCES

[1] L. Alvarez, F. Guichard, P.L. Lions, and J.M. Morel. Axioms and fundamental equations of image processing. *Archive for Rational Mechanics and Analysis*, 123(3):199–257, 1993.

[2] N. Amenta, M. Bern, and M. Kamvysselis. A new Voronoi-based surface reconstruction algorithm. pages 415–421, 1998.

[3] L. Antiga, B. Ene-Iordache, and A. Remuzzi. Computational geometry for patient-specific reconstruction and meshing of blood vessels from MR and CT angiography. *IEEE transactions on medical imaging*, 22(5):674–684, 2003.

[4] M.A. Audette, F.P. Ferrie, and T.M. Peters. An algorithmic overview of surface registration techniques for medical imaging. *Medical Image Analysis*, 4(3):201–217, 2000.

[5] F. Bernardini, J. Mittleman, H. Rushmeier, C. Silva, G. Taubin, et al. The ball-pivoting algorithm for surface reconstruction. *IEEE Transactions on Visualization and Computer Graphics*, 5(4):349–359, 1999.

[6] M. Bertalmio, AL Bertozzi, G. Sapiro, et al. Navier-Stokes, fluid dynamics, and image and video inpainting. 1, 2001.

[7] M. Bertalmio, L.T. Cheng, S. Osher, and G. Sapiro. Variational problems and partial differential equations on implicit surfaces. *Journal of Computational Physics*, 174(2):759–780, 2001.

[8] M. Bertalmio, G. Sapiro, V. Caselles, and C. Ballester. Image inpainting. pages 417–424, 2000.

[9] M. Bertalmio, L. Vese, G. Sapiro, and S. Osher. Simultaneous structure and texture image inpainting. *IEEE Transactions on Image Processing*, 12(8):882–889, 2003.

[10] A. Bertozzi, S. Esedoglu, and A. Gillette. Analysis of a two-scale Cahn-Hilliard model for image inpainting. *Multiscale Modeling and Simulation*, 6(3):913–936, 2007.

[11] A.L. Bertozzi, S. Esedoglu, and A. Gillette. Inpainting of binary images using the Cahn-Hilliard equation. *IEEE Transactions on image processing*, 16(1):285, 2007.

[12] L. Bronsard and B. Stoth. Volume-preserving mean curvature flow as a limit of a nonlocal Ginzburg-Landau equation. *SIAM Journal on Mathematical Analysis*, 28:769, 1997.

[13] L.G. Brown. A survey of image registration techniques. *ACM computing surveys (CSUR)*, 24(4):325–376, 1992.

[14] J.F. Cai, R.H. Chan, L. Shen, and Z. Shen. Convergence analysis of tight framelet approach for missing data recovery. *Advances in Computational Mathematics*, pages 1–27, 2008.

[15] J.F. Cai, R.H. Chan, and Z. Shen. A framelet-based image inpainting algorithm. *Applied and Computational Harmonic Analysis*, 24(2):131–149, 2008.

[16] R.J. Campbell and P.J. Flynn. A survey of free-form object representation and recognition techniques. *Computer Vision and Image Understanding*, 81(2):166–210, 2001.

[17] T.F. Chan, S.H. Kang, and J. Shen. Euler's elastica and curvature-based inpainting. *SIAM Journal on Applied Mathematics*, pages 564–592, 2002.

[18] T.F. Chan and J. Shen. Variational image inpainting. *Commun. Pure Appl. Math*, 58:579–619, 2005.

[19] T.F. Chan, J. Shen, and H.M. Zhou. Total variation wavelet inpainting. *Journal of Mathematical Imaging and Vision*, 25(1):107–125, 2006.

[20] Y.G. Chen, Y. Giga, and S. Goto. Uniqueness and existence of viscosity solutions of generalized mean curvature flow equations. *J. Differential Geom*, 33(3):749–786, 1991.

[21] M.A.A. Cox. *Multidimensional scaling*. CRC Press, 2000.

[22] M.G. Crandall, H. Ishii, and P.L. Lions. Users guide to viscosity solutions of second order partial differential equations. 27:1–67, 1992.

[23] M.G. Crandall and P.L. Lions. Viscosity solutions of Hamilton-Jacobi equations. *Transactions of the American Mathematical Society*, pages 1–42, 1983.

[24] MG Crandall and PL Lions. Two approximations of solutions of Hamilton-Jacobi equations. *Mathematics of Computation*, pages 1–19, 1984.

[25] L. da Fontoura Costa and R.M. Cesar. *Shape analysis and classification: theory and practice*. CRC, 2001.

[26] I. Daubechies. Ten lectures on wavelets. CBMS-NSF Lecture Notes, SIAM, nr. 61, 1992.

[27] J. Davis, S.R. Marschner, M. Garr, and M. Levoy. Filling holes in complex surfaces using volumetric diffusion. pages 428–861, 2002.

[28] JA Dobrosotskaya and AL Bertozzi. A Wavelet-Laplace Variational Technique for Image Deconvolution and Inpainting. *IEEE Transactions on Image Processing*, 17(5):657–663, 2008.

[29] B. Dong, Y. Mao, I. D. Dinov, Z. Tu, Y. Shi, Y. Wang, and A. W. Toga. Wavelet-based representation of biological shapes. *Mathematics Department, UCLA, CAM Report*, pages 07–36, 2007.

[30] M. Elad, JL Starck, P. Querre, and DL Donoho. Simultaneous cartoon and texture image inpainting using morphological component analysis (MCA). *Applied and Computational Harmonic Analysis*, 19(3):340–358, 2005.

[31] J. Escher and G. Simonett. The volume preserving mean curvature flow near spheres. *Proceedings of the American Mathematical Society*, 126(9):2789–2796, 1998.

[32] S. Esedoglu and J. Shen. Digital inpainting based on the Mumford–Shah–Euler image model. *European Journal of Applied Mathematics*, 13(04):353–370, 2002.

[33] L.C. Evans and J. Spruck. Motion of level sets by mean curvature I. *Journal of Differential Geometry*, 33(3):635–681, 1991.

[34] L.C. Evans and J. Spruck. Motion of level sets by mean curvature II. *Transactions of the American Mathematical Society*, 330(1):321–332, 1992.

[35] L.C. Evans and J. Spruck. Motion of level sets by mean curvature III. *Journal of Geometric Analysis*, 2(2):121–150, 1992.

[36] L.C. Evans and J. Spruck. Motion of level sets by mean curvature IV. *Journal of Geometric Analysis*, 5(1):77–114, 1995.

[37] B. Fischer and J. Modersitzki. Ill-posed medicine—an introduction to image registration. *Inverse Problems*, 24:034008, 2008.

[38] M.S. Floater and K. Hormann. Surface parameterization: a tutorial and survey. *Advances in multiresolution for geometric modelling*, 1, 2005.

[39] Y. Giga. Surface evolution equations: a level set method. 2002.

[40] C. Gotsman, X. Gu, and A. Sheffer. Fundamentals of spherical parameterization for 3 D meshes. *ACM Transactions on Graphics*, 22(3):358–363, 2003.

[41] IR Greenshields. 3D shape approximants via spherical wavelet decompositions [clinical pelvimetry]. pages 31–5a, 2001.

[42] X. Gu, Y. Wang, T. F. Chan, P. M. Thompson, and S.-T. Yau. Brain surface conformal mapping. *Human Brain Mapping*, 2003.

[43] X. Gu, Y. Wang, TF Chan, PM Thompson, and S.T. Yau. Genus zero surface conformal mapping and its application to brain surface mapping. *IEEE Transactions on Medical Imaging*, 23(8):949–958, 2004.

[44] X. Gu and S.T. Yau. Global conformal surface parameterization. pages 127–137, 2003.

[45] X.D. Gu and S.T. Yau. *Computational conformal geometry*. International Press, 2008.

[46] F. Guichrad and JM Morel. Image Analysis and PDEs. IPAM GBM Tutorials.(2001). March 27 - April 6, 2001.

[47] S. Haker, S. Angenent, A. Tannenbaum, R. Kikinis, G. Sapiro, and M. Halle. Conformal surface parameterization for texture mapping. *IEEE Transactions on Visualization and Computer*

Graphics, 6(2):181–189, 2000.

[48] H. Ishii. A generalization of the Bence, Merriman and Osher algorithm for motion by mean curvature. *Curvature flows and related topics*, pages 111–127.

[49] M. Jin, F. Luo, and X.D. Gu. Computing general geometric structures on surfaces using Ricci flow. *Computer-Aided Design*, 39(8):663–675, 2007.

[50] M. Jin, Y. Wang, S.T. Yau, and X. Gu. Optimal global conformal surface parameterization. pages 267–274, 2004.

[51] H. Lester and S.R. Arridge. A survey of hierarchical non-linear medical image registration. *Pattern Recognition*, 32(1):129–149, 1999.

[52] D. Levin. Mesh-independent surface interpolation. *Geometric Modeling for Scientific Visualization*, 3, 2003.

[53] S. Loncaric. A survey of shape analysis techniques. *Pattern recognition*, 1998.

[54] J.B.A. Maintz and M.A. Viergever. A survey of medical image registration. *Medical image analysis*, 2(1):1–36, 1998.

[55] S. Mallat. A wavelet tour of signal processing. 2nd ed. New York: Academic, 1999.

[56] B. Merriman, J.K. Bence, and S. Osher. Diffusion generated motion by mean curvature. *In J.E. Taylor, editor, Computational Crystal Growers Workshop, AMS, Rhode Island*, pages 73–83, 1992.

[57] B. Merriman, J.K. Bence, and S.J. Osher. Motion of multiple junctions: A level set approach. *Journal of Computational Physics*, 112(2):334–363, 1994.

[58] D. Nain, S. Haker, A. Bobick, and A. Tannenbaum. Shape-driven 3D segmentation using spherical wavelets. *Lecture Notes in Computer Science*, 4190:66, 2006.

[59] D. Nain, S. Haker, A. Bobick, and A.R. Tannenbaum. Multiscale 3d shape analysis using spherical wavelets. *Lecture Notes in Computer Science*, 3750:459, 2005.

[60] K. Ni, D. Roble, and T. Chan. A texture synthesis approach to elastica inpainting. 2007.

[61] M. Niethammer, M. Reuter, F.E. Wolter, S. Bouix, N. Peinecke, M. Koo, and M.E. Shenton. Global medical shape analysis using the Laplace-Beltrami spectrum. *Lecture Notes in Computer Science*, 4791:850, 2007.

[62] S. Osher and R.P. Fedkiw. Level set methods and dynamic implicit surfaces. 2003.

[63] S. Osher and J. Sethian. Fronts propagating with curvature-dependent speed- Algorithms based on Hamilton-Jacobi formulations. *Journal of Computational Physics*, 79:12–49, 1988.

[64] M. Pauly, L.P. Kobbelt, and M. Gross. Point-based multiscale surface representation. *ACM Transactions on Graphics (TOG)*, 25(2):177–193, 2006.

[65] E. Praun and H. Hoppe. Spherical parametrization and remeshing. *ACM Transactions on Graphics*, 22(3):340, 2003.

[66] M. Reuter, F.E. Wolter, and N. Peinecke. Laplace–Beltrami spectra as Shape-DNAof surfaces and solids. *Computer-Aided Design*, 38(4):342–366, 2006.

[67] J. Rubinstein and P. Sternberg. Nonlocal reaction-diffusion equations and nucleation. *IMA Journal of Applied Mathematics*, 48(3):249–264, 1992.

[68] S.J. Ruuth. Efficient algorithms for diffusion-generated motion by mean curvature. *Journal of Computational Physics*, 144(2):603–625, 1998.

[69] S.J. Ruuth and B. Merriman. Convolution-generated motion and generalized Huygens' principles for interface motion. *SIAM Journal on Applied Mathematics*, pages 868–890, 2000.

[70] S.J. Ruuth, B. Merriman, and S. Osher. Convolution-generated motion as a link between cellular automata and continuum pattern dynamics. *Journal of Computational Physics*, 151(2):836–861, 1999.

[71] S.J. Ruuth and B.T.R. Wetton. A simple scheme for volume-preserving motion by mean curvature. *Journal of Scientific Computing*, 19(1):373–384, 2003.

[72] P. Schröder and W. Sweldens. Spherical wavelets: Efficiently representing functions on the sphere. pages 161–172, 1995.

[73] A. Sheffer, E. Praun, and K. Rose. Mesh parameterization methods and their applications. *Foundations and Trends® in Computer Graphics and Vision*, 2(2):105–171, 2006.

[74] Y. Shi, P.M. Thompson, I. Dinov, S. Osher, and A.W. Toga. Direct cortical mapping via solving partial differential equations on implicit surfaces. *Medical image analysis*, 11(3):207–223, 2007.

[75] Y.H.R. Tsai, L.T. Cheng, S. Osher, and H.K. Zhao. Fast sweeping algorithms for a class of Hamilton-Jacobi equations. *SIAM journal on numerical analysis*, 41(2):673–694, 2004.

[76] J. Verdera, V. Caselles, M. Bertalmio, and G. Sapiro. Inpainting surface holes. 2, 2003.

[77] Y. Wang, X. Gu, T. Chan, P.M. Thompson, and S.T. Yau. Intrinsic brain surface conformal mapping using a variational method. pages 241–252, 2004.

[78] M. D. Wheeler, Y. Sato, K. Ikeuchi, A.C. Inc, and C. A. Cupertino. Consensus surfaces for modeling 3D objects from multiple rangeimages. pages 917–924, 1998.

[79] R.T. Whitaker. A level-set approach to 3D reconstruction from range data. *International Journal of Computer Vision*, 29(3):203–232, 1998.

[80] D. Zhang and G. Lu. Review of shape representation and description techniques. *Pattern Recognition*, 37(1):1–19, 2004.

[81] H.K. Zhao, S. Osher, B. Merriman, and M. Kang. Implicit and non-parametric shape reconstruction from unorganized points using variational level set method. *Computer Vision and Image Understanding*, 80(3):295–319, 2000.

[82] B. Zitova and J. Flusser. Image registration methods: a survey. *Image and vision computing*, 21(11):977–1000, 2003.