A Curvature Sensitive Demon's Algorithm for Surface Registration

Marcel Lüthi, Thomas Albrecht, Thomas Vetter

January 17, 2007

Registration, the problem of establishing correspondence between points of two objects, is one of the central problems in Computer Vision, Computer Graphics and Medical Imaging. In this paper we present a new appraoch for establishing point-to-point correspondence between two objects given as two-dimensional surfaces in three-dimensional space. In contrast to traditional approaches for surface registration, we do not register the surfaces directly. Rather, we represent the surfaces as the zero level-set of a signed distance function in three-dimensional space. Correspondence is then established for the distance function, and thus in particular for the zero-level set that represents the surface. Using this representation, the registration algorithm becomes independent of the surfaces topology and even topological changes can be dealt with. As the basis for our registration algorithm we use the well known Thirion's Demons algorithm. Our experiments showed that using this method, the surfaces are accurately matched. However, due to the lack of information on the surface, the correspondences do not always correspond to those a human expert would identify. Therefore we extend the algorithm, such that it considers the curvature information on the surface. We show that the new term fits naturally into the original formulation of Thirion's Demons. Moreover, we provide an additional interpretation of our extension in the variational framework.

We performed experiments on various synthetic and medical structures. Using the here presented representation, we were able to register extremely complex structures, such as the human skull, accurately. Our experiments show that for our data the extension yields considerably improvements compared to Thirion's original formulation.

1 Introduction

For many tasks in medical imaging, computer vision, and computer graphics, one is not interested in analyzing a single object by itself, but rather to combine information from several objects of the same class. For example to gather statistical knowledge about their shape, a large number of objects from this class are required. In order to be able to extract meaningful information from such a family of objects, its constituting parts (e.g. points) have to be brought into correspondence. This problem, known as the registration or correspondence problem, is therefore of large importance, and has accordingly been investigated extensively in the last decades.

In the most common case, the objects to be registered are given as surface meshes or n-dimensional images. The goal is to find a suitable transformation, such that a similarity measure for the objects is maximized. For surface registration, the similarity measure is usually given as a distance of the surfaces in Euclidean space. In image registration, the similarity is often measured as a function of the image intensities. Although the problems and the solution techniques appear to be similar at first glance, the algorithms that have been developed are very distinct, due to the inherently different nature of meshes and images.

In this article we present a method for the registration of surfaces using image registration algorithms. This approach allows us to register surfaces of arbitrary topology using the powerful methods developed in image registration. We represent a surface as a zero level set of a distance function that is defined over a rectangular domain and apply Thirion's Demons Algorithm [27] to register the two distance images. We show that this approach leads to a good fit between the two surfaces and that even topological changes are handled in a natural way. We also demonstrate that, despite the good match, the algorithm may fail to find meaningful correspondences within the surface. This is due to the lack of information on the surface points, that all have the same intensity. We derive a new cost function, that includes a curvature term to guide the registration in directions tangential to the surface. This additional term fits nicely into the algorithm formulated by Thirion. In addition, it admits a nice interpretation in the variational framework.

Although the method is general and can be applied to many registration tasks, our particular motivation is rooted in medical imaging. Our goal is to build statistical models of the femur-bone and the human skull. In the case of the statistical femur model, the main advantage of our method, compared to traditional surface registration methods, is that it automatically yields reasonable correspondences in a neighborhood around the surface. These correspondences are automatically obtained by warping the original image by the deformation field obtained from our algorithm. We will exploit this point in a later stage to automatically fit the inner structures of the bone. For the skull-model, the main motivation is different. The structure of a human skull is extremely complex and finding a parametrization seems to be infeasible. Further, the topology might even differ between two acquired data sets. Therefore, the method's ability to handle arbitrary and complex topologies as well as topological changes is the main driving force here.

The document is structured as follows: Section 2 gives an overview of related work in surface and image registration. We proceed by a short summary of different mathematical methods used in this paper in Section 3. Moreover, a detailed derivation of Thirion's Demons algorithm is provided. In section 4 we derive our extension to Demons algorithm. We discuss briefly the importance of the optical flow constraint. In section 5 we illustrate the performance and improvements of our formulations for different synthetic examples as well as application to medical data sets. We also discuss the applicability of the method for registration of CT-Images. A discussion of the results and an outline of future work is given in section 6.

2 Previous and related work

The registration problem is a central problem for many tasks in computer graphics, computer vision, and medical imaging. Therefore, a large amount of different registration methods can be found in the literature. For an overview of surface registration algorithms, we refer the reader to the comprehensive survey by Audette et al. [1]. A detailed treatment of image registration methods can be found in the recent books of Terry S. Yoo [32], Hajnal et al. [12] and Modersitzki [17]. Reflecting its popularity, Thirion's Demons algorithm has been studied particularly well [21, 31, 10, 13]. In [3], Cachier et al. showed how the Demons algorithm can be interpreted as a gradient descent scheme. They introduced a weighting scheme that improves numerical stability and leads to a smoother solution. Further, the method improves the solution in case the optical flow constraint is violated. Wang et.al. introduced a new "active force" in the algorithm, and they show that this force leads to faster convergence compared to the original formulation [30].

Level set methods have been widely used for image segmentation and shape representation [15, 6, 18, 4]. Recently, this representation also gained momentum for the registration problem. In [29], Vemuri et al. proposed a level-set registration method that moves the contour of a surface along its normal, until it matches the new shape. Maurel et al. [16] showed how landmark guided registration can be used within a level-set framework. [26, 25], Steinke, Schölkopf and Blanz use Support Vector Machine regression to approximate the distance function. The registration is performed using a gradient descent algorithm to minimize a cost function consisting of several terms, including a localization cost, a curvature term as well as a landmark term.

Closest to our work is the series of articles by Paragios et al. [23, 19, 20, 14]. Like Vemuri et al. they formulate a curve evolution algorithm. They extend the algorithm by a landmark term and use a gradient flow with respect to a specially tailored metric in order to favor translations, rotations and scalar motions. Their formulation of the registration problem differs from ours in the detail that they calculate a curve evolution while we calculate a deformation field, or warp, of the image domain. However, both methods are essentially equivalent and their extensions could be applied to our algorithm and vice versa.

3 Background

3.1 Surface versus Image Registration

There have traditionally been two approaches to the registration of medical images. Either the anatomical structure is segmented and the three dimensional surface extracted, or the registration is performed directly on the acquired image. In the first approach, the two surfaces are registered either by warping the surface-mesh or by choosing a suitable parametrization and performing the registration in the parameter-domain. In the second approach, the image-grid (i.e. the coordinate system) is warped such that the image intensities become similar.

For both approaches a great number of algorithms have been developed. Although the basic ideas are similar, their formulations differ greatly. In surface-registration, the prevalent approach is to define an admissible class of transformations, that depends only on a finite number of parameters. A cost function is formulated that consists usually of a similarity measure between the two surfaces and a regularization term. Registration is then performed by minimizing the cost-functions over the parameter domain.

Most image-based registration methods can be formulated in a variational framework. The images are assumed to be continuous functions defined on a given rectangular domain. Similar as in surface registration, an energy function is formulated, consisting of a similarity term and a regularizer. However, in this case the minimizer is a dense vector field defined over the whole domain and not only on the surface. Minimization is performed using methods from the calculus of variations, that is, a partial differential equation is solved. In both methods, the registration can be assisted by defining (manually or automatically) a set of landmark points to be match.

In the method proposed here, we use the Demons algorithm, which is an algorithm for image-based registration, for the registration of surfaces. The idea is to apply the algorithm to a distance image of the surfaces. By registering the images, the surfaces (i.e. the zero level set of the distance-images) are implicitly registered.

In this way, we unite some of the advantages of surface and image registration. We can take advantage of the focus on the relevant structures given by the prior segmentation, while keeping the topology-independence and efficiency of image registration.

3.2 Implicit Surfaces, Level Sets

In computer science and mathematics, there are several possible ways of representing a hypersurface $\Gamma \subset \mathbb{R}^n$. The most obvious and common representation is by specifying a parametrization $F : U \to \Gamma$ from some parameter domain $U \subset \mathbb{R}^n$ to Γ. Usually, a whole collection of parametrizations has to be given to parametrize the whole surface. For topologically complex surfaces, it is often very difficult to explicitly find

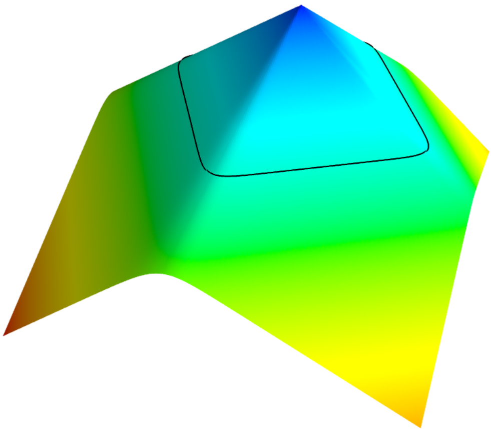


Figure 1: A level set function with its zero level set.

such parametrizations and many calculations are often rather difficult in the parametric case.

Another common way to describe a hypersurface is by representing it as the level set of an auxiliary function $\phi : \mathbb{R}^n \to \mathbb{R}$. This means that the hypersurface Γ is given as:

$$\Gamma = \{\phi = c\} := \{x \in \mathbb{R}^n \mid \phi(x) = c\},$$

for some value $c \in \mathbb{R}$, in which case we call Γ the c level set of ϕ. An example of a level-set function with its zero level set is shown in figure 1.

In the computer science community, this representation is called *implicit surface* because the surface is implicitly defined by the function ϕ. In the applied mathematics community, it is known as the *level set* representation and the function ϕ is called the level set function.

In principal, any function $\phi : \mathbb{R}^n \to \mathbb{R}$ can be used to represent the hypersurface Γ as long as $\Gamma = \{\phi = c\}$ for some $c \in \mathbb{R}$. In order to be sure that the c level set is actually a hypersurface and has no interior, we require ϕ to be non-constant in a neighborhood U of Γ. In our algorithms, we also need to calculate the gradient and curvature of ϕ so we require $\phi \in C^2(U)$.

The representation by a level set function is independent of the hypersurface's topology. For our application in image registration this is a great advantage, because it allows us to compare two hypersurfaces whose topology is not exactly the same. For instance, one surface can have more holes than the other one, but the level set functions can still be compared, whereas their parametrizations would have to be very different and therefore hard to compare.

Another advantage of the level set representation is the fact that many calculations become much less complicated than in the parametric case. In the parametric case, all calculations are performed in the $\mathbb{R}^{n-1}$-dimensional parameter domain, and to be able to correctly calculate quantities like scalar products, norms and curvature, we have to work with the Riemannian metric of the parametrization. In the level set case, all calculations are performed in the space surrounding the surface, so the Riemannian metric is simply the regular scalar product in $\mathbb{R}^n$.

The main disadvantage of the level set representation of a hypersurface, both mathematically and from a practical computer science point of view, is that by representing an $\mathbb{R}^{n-1}$-dimensional surface by a $\mathbb{R}^n$-dimensional function we turn an $(n-1)$-dimensional problem into an n-dimensional one. As a result, the algorithmic complexity, especially the memory requirement poses many additional challenges.

In practice, the most common choice of a level set function to represent a given $\Gamma \subset \mathbb{R}^n$ is to use the signed distance function to Γ:

$$\phi(x) := d_\Gamma(x) = \begin{cases} \operatorname{dist}(x, \Gamma) & x \in \operatorname{outside}(\Gamma) \\ 0 & x \in \Gamma \\ -\operatorname{dist}(x, \Gamma) & x \in \operatorname{inside}(\Gamma), \end{cases} \tag{1}$$

where $\operatorname{dist}(x, \Gamma)$ is the Euclidian distance from x to Γ and the inside and outside of Γ have to be assigned in some meaningful way. Γ is then given as the zero level set of d_Γ. It is well known that d is globally Lipschitz-continuous and that for $\phi \in C^2(U)$ there exists a $\delta > 0$ such that $d_\Gamma \in C^2(\Gamma_\delta)$, where $\Gamma_\delta = \{x \in \mathbb{R}^n \mid |d_\Gamma(x)| < \delta\}$. In addition, $|\nabla d_\Gamma| \equiv 1$ in Γ_δ, and for a point $x \in \Gamma$, the outer unit normal vector $\nu(x)$ is given by

$$\nu(x) = \nabla d_\Gamma(x).$$

The distance function is usually calculated on a rectangular domain $\Omega \subset \mathbb{R}^n$ with $\Gamma \subset \Omega$. Apart from the convenient properties outlined above, the main reason for using the signed distance function is that level set algorithms are usually calculated on all of Ω and not just on the zero level set that represents Γ. When the distance function is used to represent Γ, all other level sets essentially carry the same information as the zero level set and effectively guide the algorithm towards a meaningful solution both close to and away from the zero level set Γ. Further, the distance image exhibits "ridges" at the median of the structure. These "ridges" will be automatically registered by the algorithm. An example of the distance function for a two-dimensional slice through the femur is given in figure 2.

3.3 Thirion's Demons

In his landmark paper, Thirion [27] proposed a method for three-dimensional, non-rigid image registration. The basic idea Thirion formulated was inspired by Maxwell's demons in thermodynamics. Due to its excellent performance and relative ease of implementation, the algorithm became quickly one of the most widely used registration methods in medical imaging. Some of the ideas in his original paper were initially derived heuristically, and were later rigorously studied and formalized.

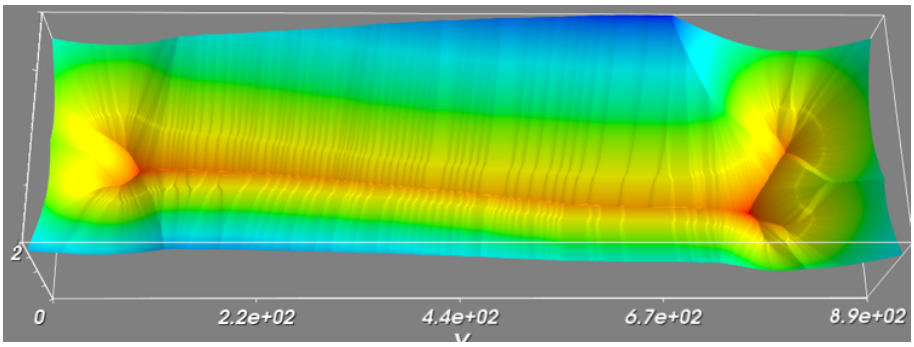


Figure 2: The distance image of a 2D outline of a femur. The yellow line represents the zero level set. The ridges through the median are a useful feature for the registration algorithm.

The general scheme is outlined in Algorithm 1. It iteratively calculates a displacement field u. In each iteration, u is updated with an update term v and subsequently smoothed with a Gaussian kernel.

We describe here two different derivations of the algorithm, due to Modersitzki [17], that are especially well suited to motivate our proposed extension of the algorithm. In particular we derive an explicit expression for the update term v.

3.3.1 Demons algorithm as Optical Flow

In this first derivation, we interpret the Demons algorithm as a special case of optical flow, a well-known image processing technique. Let $I_0 : \Omega \subset \mathbb{R}^n \to \mathbb{R}$ be an image. We assume that I_0 is continuously deformed over the time interval $[0,1]$ and call the resulting image I_1. The intermediate images are denoted by I_t with $t \in (0,1)$. We admit only deformations that are given by a warp of the image domain Ω. Every such warp can be represented by a flow field

$$\gamma = \gamma(x,t) : \Omega \times [0,1] \to \Omega,$$

and so the image deformation is given as:

$$I_t = I(x,t) : \Omega \times [0,1] \to \mathbb{R}$$

$$I(x,t) := I_0(\gamma(x,t)). \tag{2}$$

Algorithm 1 An outline of the demons algorithm

```
  $u(x_0) = 0,\ \forall x_0 \in \Omega$ {Set the initial displacement to 0}
  **for** $k = 0$ to $N$ **do**
    For each grid-position $x_0$ calculate $v^{(k+1)}(x_0)$ from $I_0(x_0 + u^{(k)}(x_0))$ and $I_1(x_0)$
    $u^{(k+1)} = u^{(k)} + v$ {Sum up the displacements}
    $u^{(k+1)} = G_\sigma \star u^{(k+1)}$ {Smooth the displacement field with the a Gaussian kernel $G_\sigma$}
  **end for**
```

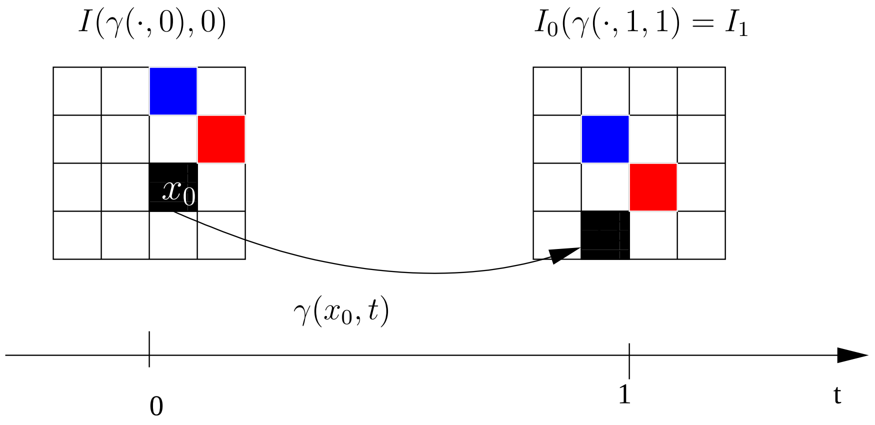


Figure 3: A particle x_0 moves along its trajectory $\gamma(x_0,t)$

In this formulation, the registration task consists of finding a warp $\gamma(\cdot,t)$ such that the deformation I_t, defined by γ, deforms two given images I_0 and I_1 into each other. If we assume that we have found such a γ, the value $\gamma(x,0)=x$ represents the position of a pixel $x\in\Omega$ in the image I_0 and $\gamma(x,1)$ the position of the corresponding pixel in the image I_1. More precisely:

$$I_0(x) = I(x,0),\ \forall x_0 \in \Omega \text{ and} \tag{3}$$

$$I_1(x) = I(x,1),\ \forall x_0 \in \Omega, \tag{4}$$

The curve $\gamma_x := \gamma(x,\cdot) : [0,1] \to \Omega$ then describes the trajectory from x to its corresponding point $\gamma(x,1)$ as illustrated in figure 3.

The assumption that it is possible to find a homotopy I_t between the two given images I_0 and I_1 implies the requirement that I_0 and I_1 have the same intensity values, i.e. $I_0(\Omega) = I_1(\Omega)$. It follows that this is also the case for each I_t, $t\in[0,1]$, and that the intensity values $I_t(\gamma(x_0,t)$ of the trajectory $\gamma(x_0,t)$ are constant for each $x_0\in\Omega$.

Now consider an arbitrary but fixed x_0. To simplify the notation, write $x(t)=\gamma(x_0,t)$ and differentiate (2) with respect to t. We get

$$0 = \frac{d}{dt}I(x(t),t) = \frac{\partial}{\partial t}I(x(t),t) + \nabla I(x(t),t)\cdot\frac{\partial}{\partial t}x(t). \tag{5}$$

This equation is commonly known as the *optical flow constraint*. While it may seem to be unrealistic to assume that it holds for most image pairs, it is nevertheless used to derive the Demons Algorithm.We shall see in the next section that the Demons algorithm can also be formulated as a minimization problem, that is well-defined even if the optical flow constraint is not strictly satisfied.

The term $\frac{\partial}{\partial t}x(t)$ can be interpreted as the velocity of the particle and is hence denoted $v(t)$. Using a finite differences to approximate the time derivative in the optical flow constraint (5), we get:

$$0 \approx \frac{I(x(t),t+\tau) - I(x(t),t)}{\tau} + \nabla I(x(t),t)\cdot v(t). \tag{6}$$

By choosing $t = 0$ and $\tau = 1$ we obtain

$$-\nabla I_0(x(0), 0) \cdot v(0) \approx (I(x(0), 1) - I(x(0), 0))$$

and since $x(0) = \gamma(x_0, 0) = x_0$ and by using (3) this simplifies to

$$-\nabla I_0(x_0) \cdot v(0) \approx (I_1(x_0) - I_0(x_0)).$$

Premultiplying both sides by $\nabla I_0(x_0)$, we get an explicit expression for the approximated velocity $\tilde{v} \approx v(0)$ that depends only on the given two images:

$$\tilde{v} = -\frac{I_1(x_0) - I_0(x_0)}{\nabla I_0(x_0) \cdot \nabla I_0(x_0)} \nabla I_0(x_0) + w = -\frac{I_1(x_0) - I_0(x_0)}{|\nabla I_0(x_0)^2|} \nabla I_0(x_0) + w. \tag{7}$$

Here $w \in \operatorname{span}(\nabla I_0(x_0))^\perp$ is any term orthogonal to $\nabla I_0(x_0)$. We stress that any such w fulfills the optical flow constraint (5).[1] Thirion suggests to choose $w = 0$, which means that the calculated flow field can only move in the direction of ∇I_0. This is a rather strong restriction.

To prevent numerical instabilities when $|\nabla I_0(x_0)|$ is small, an additional regularization term $\kappa^2 = (I_0(x_0) - I_1(x_0))^2$ is introduced and we define:

$$\tilde{v} := -\frac{I_1(x_0) - I_0(x_0)}{|\nabla I_0(x_0)^2| + \kappa^2} \nabla I_0(x_0). \tag{8}$$

As an approximation of the time derivative $v(0) = \frac{\partial}{\partial t} x(0) = \frac{\partial}{\partial t} \gamma(x_0, 0)$, $\tilde{v}$ is used for a first order Taylor series approximation of γ:

$$\gamma(x_0, 1) \approx \gamma(x_0, 0) + \frac{\partial}{\partial t} \gamma(x_0, 0) \approx x_0 + \tilde{v}$$

Obviously, this approximation by itself is rather crude, but it can be exploited in a fixed-point iteration scheme:

In the first step, $\tilde{v} =: \tilde{v}^{(0)}$ is calculated for the two given images I_0 and I_1 to give a first approximation of $\gamma(x, 1)$. This in turn yields a first approximation of the deformation of I_0 into I_1, since $I_0(x_0 + \tilde{v}_0) \approx I_0(\gamma(x_0, 1)) = I_1(x_0)$.

Proceeding iteratively, in the k-th iteration step, $\tilde{v}^{(k+1)}$ is calculated as in equation (8), but with $I_0(x_0 + \sum_{i=0}^{k} \tilde{v}^{(i)}) =: I_0(x_0 + u^{(k)})$ as the source images, or more formally

$$\tilde{v}^{(k+1)} = -\frac{I_0(x + u^{(k)}(x)) - I_1(x)}{|\nabla I_0(x + u^{(k)}(x))|^2 + \kappa^2} \nabla I_0(x + u^{(k)}(x)). \tag{9}$$

The approximation of $\gamma(x_0, 1) \approx x_0 + \sum_{i=0}^{k} \tilde{v}^{(k)} = x_0 + u^{(k)}$ becomes more accurate in each iteration. We define $\tilde{v}^{(k+1)}$ as the update vector $v^{(k+1)}$ in the Demons algorithm 1.

These calculations, which we have defined for a fixed but arbitrary $x_0 \in \Omega$ are carried out for all $x \in \Omega$. To get a coherent deformation field $\gamma(x, 1) = x + u(x)$, the displacement field $u^{(k+1)}$ is smoothed in each step of the Demons algorithm.

[1] In the optical flow literature, the fact that v is only determined up the its orthogonal component is known as the *aperture problem*. It means that motion perpendicular to the gradient cannot be determined by this algorithm.

3.3.2 Variational formulation of the demons algorithm

Next we present the derivation of the algorithm using the variational framework introduced in [17]. Although the resulting algorithm is the same, it provides an alternative view on the algorithm, which will later be useful to interpret our extension.

As before, we are looking for a domain warp

$$\gamma = \gamma(x,t) : \Omega \times [0,1] \to \Omega,$$

in order to deform two given images I_0 and I_1 into each other via the homotopy $I_t(x) = I_0(\gamma(x,t))$. Without loss of generality, we assume $\gamma(\cdot,1)$ to be given in the form

$$\gamma(x,1) = x + u(x),$$

with a displacement field

$$u = u(x) : \Omega \to \Omega.$$

We now define a distance measure between two images I and J:

$$\mathcal{D}(I,J) := \frac{1}{2}\int_\Omega \frac{(I(x) - J(x))^2}{\|\nabla I(x)\|^2 + \kappa^2}\,dx. \tag{10}$$

In this setting, the registration problem consists of finding the displacement field u such that the distance

$$\mathcal{D}[u] := \mathcal{D}(I_0(\cdot + u), I_1) = \frac{1}{2}\int_\Omega \frac{(I_0(x+u(x)) - I_1(x))^2}{\|\nabla I_0(x)\|^2 + \kappa^2}\,dx \tag{11}$$

is minimized. Clearly, this optimization problem is ill-posed. In the spirit of Tikhonov optimization [28], we add a regularization term

$$\mathcal{S}[u] := \frac{1}{2}\sum_{l=1}^{d}\int_\Omega |\nabla u_l|^2\,dx \tag{12}$$

to $\mathcal{D}$, where d denotes the dimension of the image. The goal is now to find a minimizer $u : \mathbb{R}^n \to \mathbb{R}^n$ of the functional

$$\mathcal{J}[u] := \mathcal{D}[u] + \alpha\mathcal{S}[u]. \tag{13}$$

Intuitively, such a minimizer will displace each $x_0 \in \Omega$ such that the images are similar in terms of the distance measure $\mathcal{D}$. The regularization term $\mathcal{S}$ will favor smooth displacements.

It is not assumed, as in the previous section, that the deformed image $I_0(x + u(x))$ is exactly equal to $I_1(x)$ and that they share the same intensity values. In this formulation, we are simply asking for the distance between $I_0(x+u(x))$ and I_1 to be minimal, according to the distance measure $\mathcal{D}$. The *optical flow constraint* is not necessary in this variational interpretation of the Demons algorithm.

From the calculus of variations (see e.g. [8]), we know that a minimizer of the functional (13) is given by the solution to the Euler-Lagrange equation

$$\frac{(I_0(x+u(x)) - I_1(x))}{\|\nabla I_0(x)\|^2 + \kappa^2} \nabla I_0(x+u(x)) - \alpha \Delta u(x) = 0, \ \forall x \in \Omega. \tag{14}$$

The partial differential equation (14) can be approximately solved by employing a fixed-point iteration scheme (see [17] for details)

$$\frac{u^{k+1}(x) - u^{(k)}(x)}{\tau} - \alpha \Delta u^{(k)} = -\frac{(I_0(x+u^{(k)}(x)) - I_1(x))}{\|\nabla I_0(x)\|^2 + \kappa^2} \nabla I_0(x+u^{(k)}(x)).$$

Setting $\tau = 1$, assuming for the moment that $\alpha = 0$ and interpreting $u^{(k)}$ as a sum of previous updates, i.e. $u^{(k)} = \sum_{i=0}^{k}, \tilde{v}^{(i)}$, we arrive at

$$\tilde{v}^{(k+1)} = -\frac{I_0(x+u^{(k)}(x)) - I_1(x)}{\|\nabla I_0(x)\|^2 + \kappa^2} \nabla I_0(x+u^{(k)}(x)).$$

Comparing this equation to equation (9), we see that the only difference is that in (9) the terms in the denominator are evaluated at the new position $x + u^{(k)}(x)$. Assuming that the initial images are roughly aligned, and noticing that the denominator can be regarded as a scalar controlling the step length, this difference is negligible. In particular, if the two images are the distance images of two surfaces, $\|\nabla I(x)\| = 1$ almost everywhere. Alternatively, one can consider the interpretation of Pennec et al. in [22] $\|\nabla I_0(x + u(x))\|^2 + \kappa^2$.

If $\alpha \neq 0$, the effect of the term $\alpha \triangle u^{(k)}$ can be seen as the smoothing of $u^{(k)}$ by heat diffusion, which can be calculated by a convolution with a Gaussian kernel in each iteration [9]. Hence this numerical scheme corresponds to the one outlined in algorithm 1.

3.4 Mean Curvature

The *mean curvature* of a hypersurface in $\mathbb{R}^n$ is defined as the mean of its principal curvatures. For more insight, we refer to a book on elementary differential geometry, such as [5].

If the hypersurface Γ is given as a level set of a level set function $\phi : \Omega \subset \mathbb{R}^n \to \mathbb{R}$, the mean curvature of Γ at a point $x \in \Gamma$ is given by the very compact expression:

$$H(x) = H(\phi(x)) = \operatorname{div}\left(\tfrac{\nabla \phi(x)}{|\nabla \phi(x)|}\right). \tag{15}$$

The sign of the mean curvature depends on the choice of the in- or outward pointing unit normal vector, which in this form is determined by the level set function ϕ. Here, we assume that $\phi < 0$ on the inside of the shape outlined by Γ so that

$$\nu(x) = \frac{\nabla \phi(x)}{|\nabla \phi(x)|} \tag{16}$$

is the outward pointing normal vector. This means that the sign of H is chosen so that a circle has *positive* mean curvature.

As outlined above, we represent the hypersurfaces to be registered by level set functions. While we are only interested in the hypersurfaces itself, which are given as the zero level sets of the functions, the algorithm is evaluated on all of Ω, the domain of the level set functions. In a similar fashion, we wish to evaluate H not only on Γ but on all of Ω. So as an additional input to our algorithm, we need to compute the function $H : \Omega \subset \mathbb{R}^n \to \mathbb{R}$.

Here, several practical and theoretical problems arise. First of all, there may be points at which $\nabla\phi$ vanishes or is not defined. For instance, if we use $\phi = d_\Gamma$ as the level set function, there may be points, which are equidistant to two different parts of Γ, at which $\nabla\phi$ is not defined, cf. figure 2. In practice, this is not a grave problem, because the gradient can be defined in a weak sense and the numerical approximation $\nabla\phi_h$ can be calculated in any case. However, this numerical approximation may very well be zero at these points. So to be able to calculate H everywhere, we introduce a regularizing parameter ε and replace H by

$$H_\varepsilon = H_\varepsilon(\phi(x)) = \operatorname{div}\left(\tfrac{\nabla\phi}{\sqrt{\varepsilon^2+|\nabla\phi|^2}}\right).$$

The next problem is of a more practical nature. The direct numerical calculation of H_ε for a discrete function ϕ produces many artifacts and cannot be used for further analysis. This effect is much worse if we wish to use a CT-scan or X-ray image as input.

To overcome this problem, we calculate H_ε on a smoothed version of ϕ. While the most obvious possibility for smoothing ϕ is the convolution with a Gauss kernel, we use a different approach. In image processing, several methods for edge-preserving smoothing have been studied. One of the most successful methods has a close connection to our problem: the *mean curvature flow.* Starting from an initial image, or level set function ϕ^0, the time dependent partial differential equation

$$\frac{\partial}{\partial t}\phi(x,t) = |\nabla\phi(x,t)|\, H_\varepsilon(\phi(x,t)) \quad \text{in } \Omega \times \mathbb{R}_0^+ \tag{17}$$

is solved. The time-dependent solution $\phi : \Omega \times \mathbb{R}_0^+ \to \mathbb{R}$ is equal to the original image at time 0, i.e. $\phi(x,0) = \phi^0(x)$. For each $t \in \mathbb{R}^+$, $\phi(x,t)$ is a smoothed version of the image. The larger t is, the more the image is smoothed.

The time derivative in equation 17 can be discretized by a difference quotient:

$$\frac{\partial\phi}{\partial t} \approx \frac{\phi^m - \phi^{m-1}}{\tau},$$

where $\phi^m = \phi(\cdot, m\tau)$ and $m \in \mathbb{N}$. If the timestep size τ is chosen small enough, the equation (17) can thus be time-discretized by an explicit Euler scheme:

$$\phi^m = \phi^{m-1} + \tau|\nabla\phi^{m-1}|\, H_\varepsilon(\phi^{m-1}). \tag{18}$$

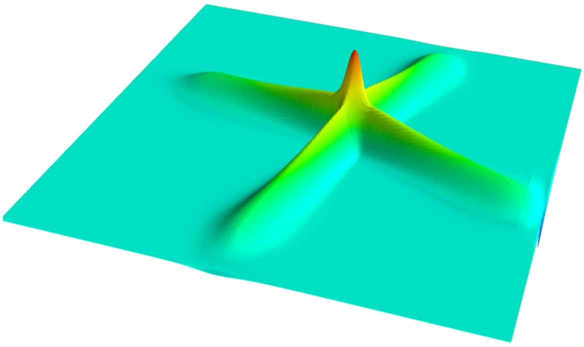


Figure 4: The curvature image for the level set function from Figure 1

This means that in each time step, ϕ^{m-1} is updated by $\tau|\nabla\phi^{m-1}|H_\varepsilon(\phi^{m-1})$.

For this update term, the mean curvature of the previous timestep $H_\varepsilon(\phi^{m-1})$ has to be calculated. To obtain the mean curvature of a mildly smoothed version of ϕ^0, we calculate the algorithm for a few timesteps with a small timestep size and keep the mean curvature from the last timestep's update term. This is the mean curvature feature function $H : \Omega \to \mathbb{R}$ we use in our registration algorithm. Figure 4 shows the function $H_\varepsilon(\phi^{m-1})$, the result of running this algorithm for $m = 10$ timesteps with timestep size $\tau = 0.05$, with the level set function from Figure 1 as input.

For the rest of the registration algorithm, we have the choice to use either the smoothed version ϕ^{m-1} or the original version ϕ^0 of our level set function.

Remark. If the level set function ϕ is given as the signed distance function d_Γ of Γ, we have $|\nabla\phi| \equiv 1$, and so the mean curvature of the level sets of ϕ is given by the Laplacian of ϕ:

$$H(x) = \operatorname{div} \nabla\phi(x) = \triangle\phi(x).$$

This means that at least for small t, the mean curvature flow is essentially the same as the heat equation:

$$\frac{\partial\phi}{\partial t} = \triangle\phi \quad \text{in } \Omega \times \mathbb{R}_0^+.$$

As is well known [9] the heat equation can be solved by a convolution with a Gauss kernel, so at the expense of some precision, the calculation of the curvature image can be significantly sped up by simply calculating the Laplacian of the distance map ϕ after it has been smoothed with a Gauss kernel.

4 Curvature Demon

As has been shown in section 3.3.1, the demons algorithm can be interpreted as a special case of an optical flow algorithm. A main point of criticism of optical flow algorithms,

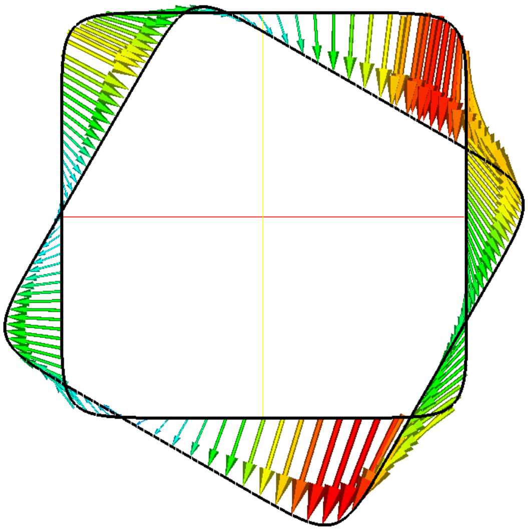


Figure 5: A registration result using the demons algorithm. The shapes are well matched.

is that the optical flow constraint is usually not appropriate for real images. The aim of a registration algorithm is to find corresponding points in two images. The optical flow constraint is the assumption that corresponding points have exactly the same intensity value. (i.e. the image intensities are not the same in the two images). Since we register the distance images, the optical flow constraint always holds in the vicinity of the zero-level set (i.e. the surface). Figure 5 shows, that the images are indeed registered such that the shape of the zero level set matches perfectly. However, for the purpose of establishing meaningful point-to-point correspondence, matching the shape well is a necessary but not sufficient criterion. Figure 5 also shows that the deformation field tends to point in the direction orthogonal to the shape, but does not necessarily match features such as corners or bumps in a surface.

To gain more insight into the algorithm's behavior, we recall equation (8), where the velocity is computed at a point x_0:

$$v = \frac{I_1(x_0) - I_0(x_0)}{|\nabla I_0(x_0)^2| + \kappa^2} \nabla I_0(x_0) + w.$$

We defined w to be an arbitrary vector orthogonal to the gradient at x_0, that is $w \in \text{span}(\nabla I_0(x_0))^\perp$. In the original formulation, this term was chosen to be the zero vector. Note that since we register two distance images, the gradient at x_0 (and thus v) is orthogonal to the level set at this point. Hence the level sets move mainly in their normal direction, and the only forces in different directions are due to smoothing. Our goal is to choose a suitable vector w at each point x_0, such that, in addition to the intensity-matching, points with similar curvature are matched.

By considering the equivalent variational formulation introduced in section 6.1.1, the derivation of a term w that has this effect becomes almost trivial. As discussed above, we look for a minimizer of the functional

$$\mathcal{J}[u] := \mathcal{D}[u] + \alpha\mathcal{S}[u].$$

where $\mathcal{D}$ is the similarity measure

$$\mathcal{D}[u] := \frac{1}{2}\int_\Omega \frac{(I_0(x+u(x)) - I_1(x))^2}{|\nabla I_0(x)|^2 + \kappa^2}\,dx$$

and $\mathcal{S}$ the regularization term. We notice that similarity is only defined in terms of the intensity difference of the images. We extend the functional to include a curvature term

$$\mathcal{C}[u] := \frac{1}{2}\int_\Omega \frac{(H_0(x+u(x)) - H_1(x)^2)}{|\nabla H_0(x)^2| + \kappa_H^2}\,dx.$$

where $H_0(x)$ and $H_1(x)$ is the mean curvature at point x for I_1 and I_2 respectively. $\kappa_H^2 = (H_0(x) - H_1(x))^2$ is introduced to prevent numerical instabilities when ∇H_0 is small. The new functional is defined as

$$\mathcal{J}[u] := \mathcal{D}[u] + \beta\mathcal{C}[u] + \alpha\mathcal{S}[u]. \tag{19}$$

Following the procedure outlined in section 6.1.1, we minimize the functional (19) by formulating the Euler-Lagrange equation and solve it using a fixed-point iteration scheme. In addition to the update term

$$v = -\frac{I_0(x+u) - I_1(x)}{|\nabla I_0(x)|^2 + \kappa^2}\nabla I_0(x)$$

corresponding to the similarity measure $\mathcal{D}$, the new functional yields an additional update

$$w = -\frac{H_0(x+u) - H_1(x)}{|\nabla H_0(x)|^2 + \kappa_H^2}\nabla H_0(x).$$

from the curvature term $\mathcal{C}$, that excerts a force towards the points with similar curvature.

By projecting w onto $(\nabla I_0)^\perp$,

$$w^\perp(x_0) := w(x_0) - (\nabla I_0(x_0) \cdot w(x_0)) \cdot \frac{\nabla I_0(x_0)}{|\nabla I_0(x_0)|}$$

and coming back to the optical-flow formulation, we can interpret $w^\perp$ as being the orthogonal term in (4) that pulls v towards matching points with similar curvature.

4.1 The importance of the optical flow constraint

In the computer vision community, optical flow algorithms have been around for a long time. However, they were rarely applied directly to the original images, but rather to the corresponding gradient images. The main reason is that the optical flow constraint is violated in almost all practical situations (e.g. due to different light conditions, etc.).

In our formulation, the optical flow constraint is clearly satisfied for the distance images. For the newly added curvature term, this is unfortunately not the case. However, we argue that this is not a big drawback of our method. By interpreting the Demon's algorithm in the variational framework, the optical flow constraint does not appear in the derivation, but the more general problem of finding the minimum distance between two images is solved. This problem has a meaningful interpretation even for images that violate the optical flow constraint. Furthermore, we use the forces from the curvature term to move the correspondences in the direction tangential to the surface, that is to establish the effective point-to-point correspondences. These correspondences can not unambiguously be defined pointwise. Therefore an approximate solution is usually sufficient.

Figure 6 shows the registration results for two one-dimensional images, where the optical flow constraint is violated almost everywhere. We observe that one image has been shifted to the left, such that the extreme values coincide. The experiment illustrates a typical situation for the curvature images. The curvature at corresponding points is not exactly the same, but in both images a local maximum is attained. Thus the algorithm finds meaningful correspondences for the curvature images.

5 Examples

5.1 2D Synthetic Example

In order to exhibit the improvement of our algorithm over Thirion's original Demons algorithm, we first consider a synthetic example. We try to register two squares with rounded corners. As described before, each shape is represented by its signed distance function. One of the two distance functions, for this specific example can be seen in Figure 1. The shape is given as the zero level set of this function, marked in black.

When we use Thirion's Demons algorithm to register the two distance images these simple shapes are matched perfectly. That is, the first image is warped so that it is indistinguishable from the other. This deformation implies a correspondence between the two shapes. As the deformation is calculated in the form of a deformation field, we can display the transformation of the points on the first shape and see to which point on the other shape they are deformed to.

The result can be seen in Figure 7a. We observe that the arrows indeed go exactly from one shape to the other, which is equivalent to an exact match. However, we cannot be satisfied with the correspondence results. In this example, it is quite clear that it does not make sense to match the corner of one shape to the edge of the other. Obviously, we would expect corners to be matched to corners and edges to edges. Figure 7b shows that our extended Demons algorithm does just that. The improvement is due

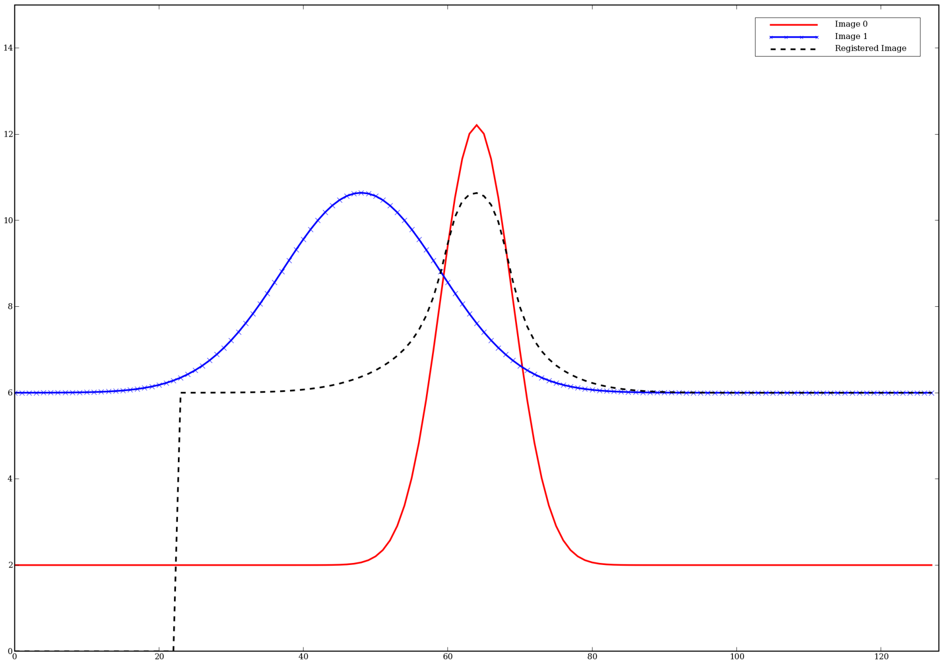


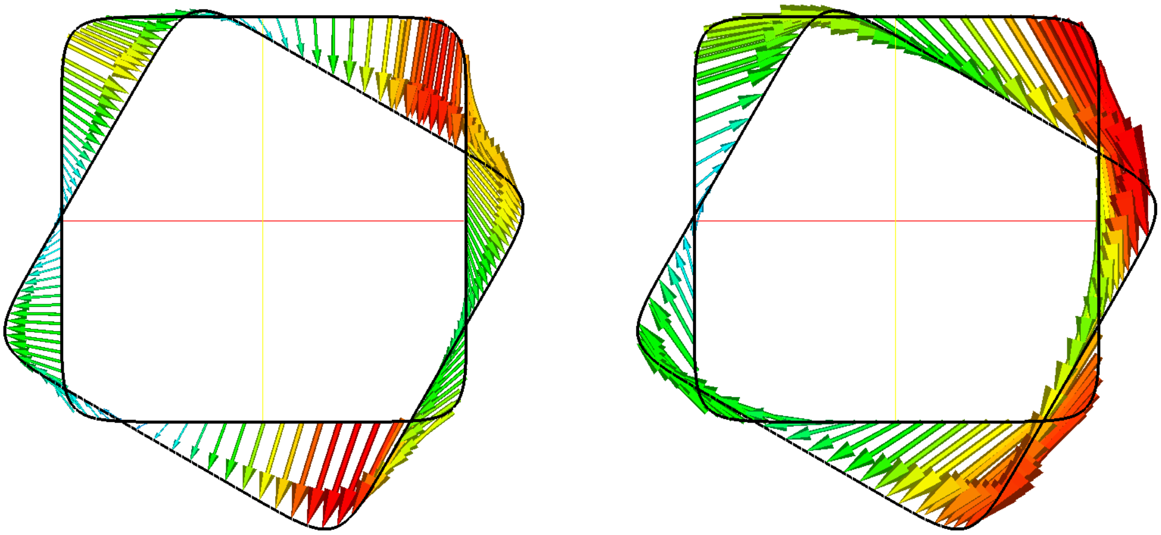
Figure 6: Registration of two one-dimensional images, where the optical flow constraint is violated almost everywhere

to the additional matching of the curvature images. One of these images is shown in Figure 4.

We observe that, apart from a small influence from the smoothing of the deformation field, the original Demons algorithm admits only deformations in the direction of the image gradient, i.e. normal to the shape outline. This can be observed in Figure 8, which shows the complete dense deformation fields calculated by the algorithms. While the original Demons' deformation field is mostly normal to the level sets of the first shape's distance function, the extended version admits a tangential component for matching the curvature. The resulting field looks like a combination of a rotation and a translation, which is in fact the most plausible transformation to match the two shapes.

5.2 2D Medical Example

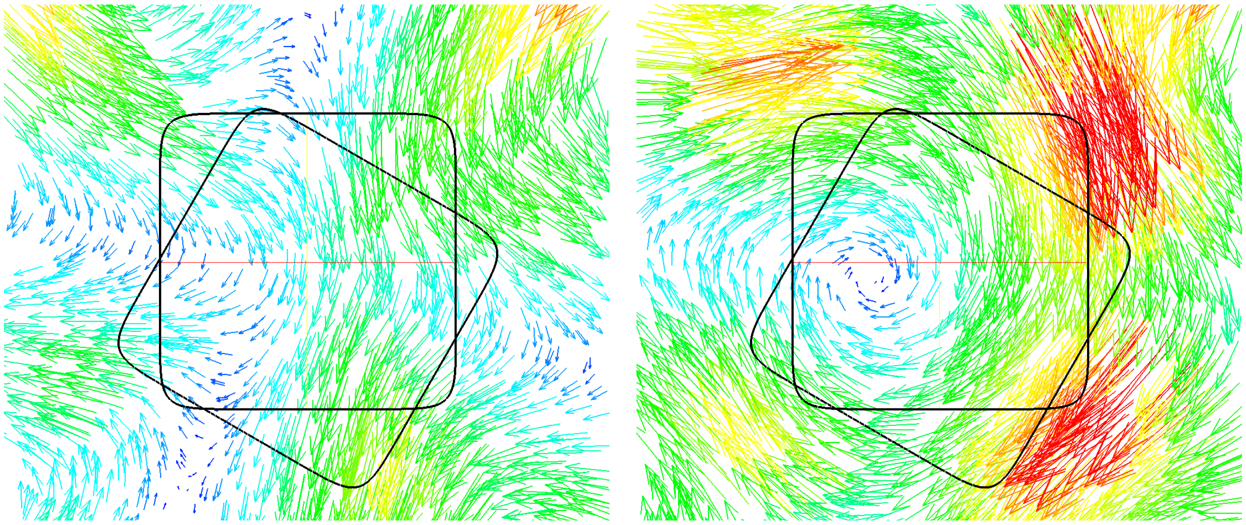
In this example we exhibit the application of our algorithm on a 2-dimensional registration problem. We will try to register two 2-dimensional slices of a 3D bone shape. Figure 9a shows how the first of the two slices from Figure 9b is obtained as the intersection of the bone with a plane. The second slice is obtained in the same way, but from



(a) Correspondence found by Thirion's Demons algorithm

(b) Correspondence found by the curvature guided Demons algorithm

Figure 7: Comparison of the original and enhanced Demons registration



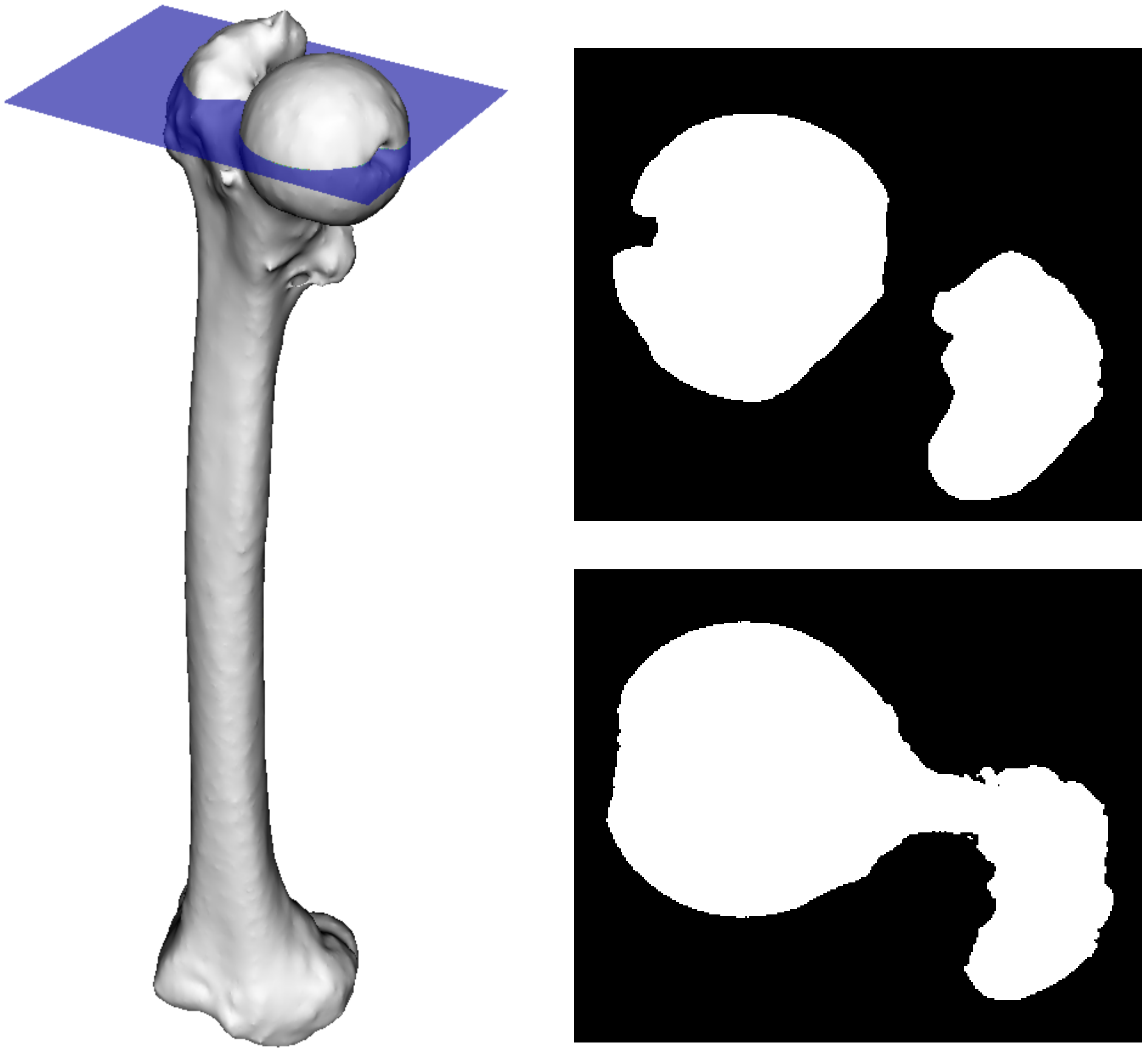
(a) Deformation field found by Thirion's Demons algorithm

(b) Deformation field found by the curvature guided Demons algorithm

Figure 8: Comparison of the dense deformation fields calculated by the original and the enhanced Demons registration

a different bone. In order to thoroughly test the algorithm, we purposely selected two slices with different topologies.

When seen as part of the 3-dimensional registration of the two complete bones, this particular difference in topology of the two slices is not significant, because both bones are simply connected. The topology would be the same for both slices, if we took the slice at a slightly different position. In this isolated 2-dimensional registration problem however, one of the shapes is connected and the other one is not and yet they have to be registered.

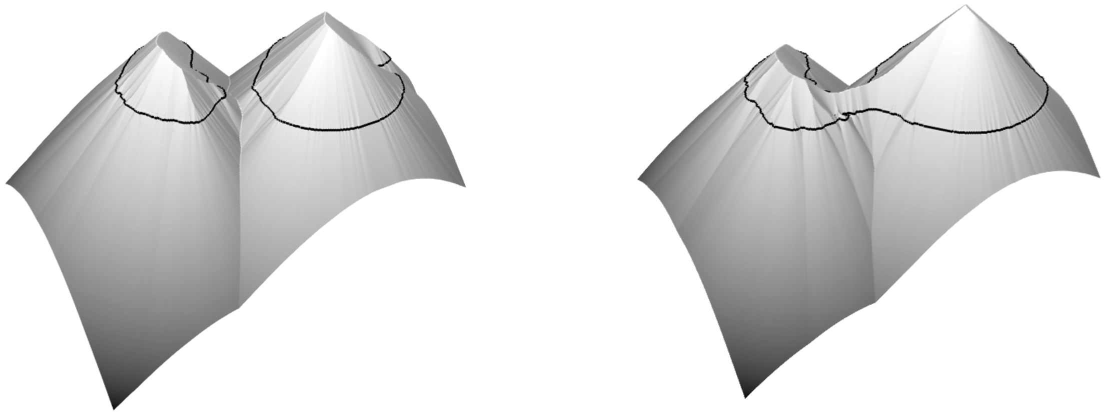


(a) A slice of the bone

(b) Two slices as binary images

Figure 9: Illustration of the process of taking a "slice" of a 3D bone. The first slice is to be registered to the second slice. The second slice is not only taken from another position but from another bone.

In order to register these shapes, one could try to directly register the binary images seen in Figure 9b. But they are discontinuous and contain no information inside or outside of the boundary. A much better approach is to register the distance functions of the boundaries. These distance functions can be interpreted as “mountains”, whose contour line at height zero is the boundary of the shapes. The “mountains” for this example are shown in Figure 10.



(a) The first mountain

(b) The second mountain

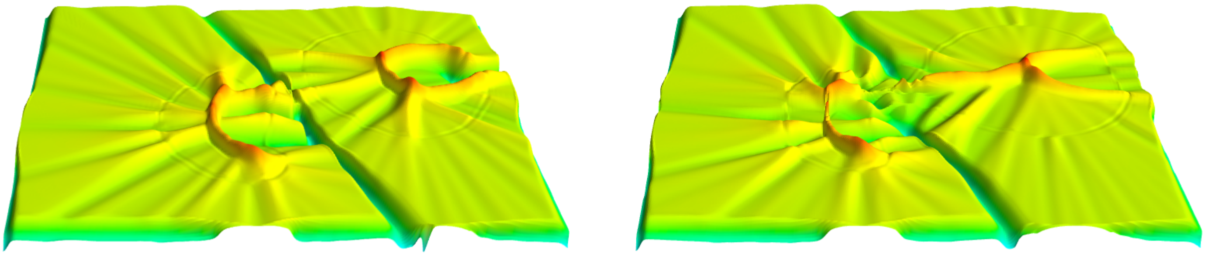
Figure 10: The level set functions of the binary images in figure 9b.

It now becomes clear how the difference in topology can be handled in this setting. It is not at all clear how to define correspondence between the two shapes with different topology. But it is obviously possible to find a deformation from one function into the other in such a way that the cost functional is minimized. The results show that this minimal solution gives a reasonable registration of the two shapes with different topology.

As outlined in Section 4, we aim at matching not only the distance maps but also the curvature maps. They are shown in Figure 11.

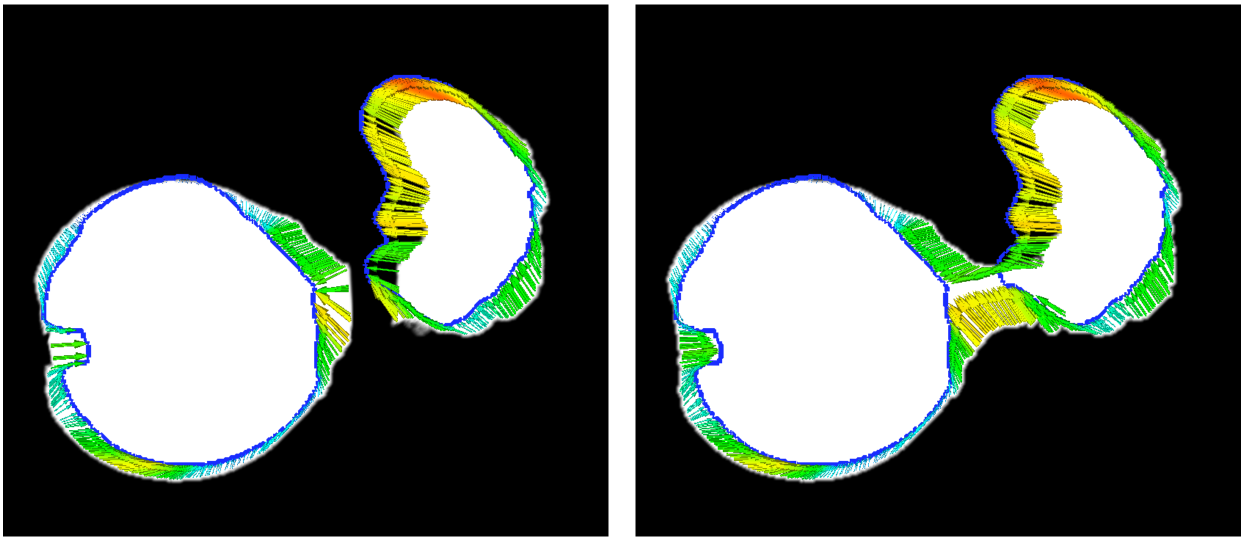
The resulting registration can be interpreted in a number of different ways. First, in Figure 12, we show the result of registering the connected to the non-connected shape. Warping the non-connected shape with the resulting deformation field yields the shape shown in Figure 12a. It looks like the connected shape, only cut in two so that it best matches the blue non-connected shape. The arrows show the correspondences. Figure 12b visualizes the same registration. Here, the arrows show the correspondence between the connected and non-connected shape. Wherever there is a meaningful correspondence, it is found. Points that simply have no correspondence, are registered to the space where a connection could be in the blue shape.

Figure 12 shows the opposite registration, from the non-connected to the connected shape. In Figure 13a, the most reasonable connection is added and in Figure 13b the non-connected is matched to the connected as good as possible.



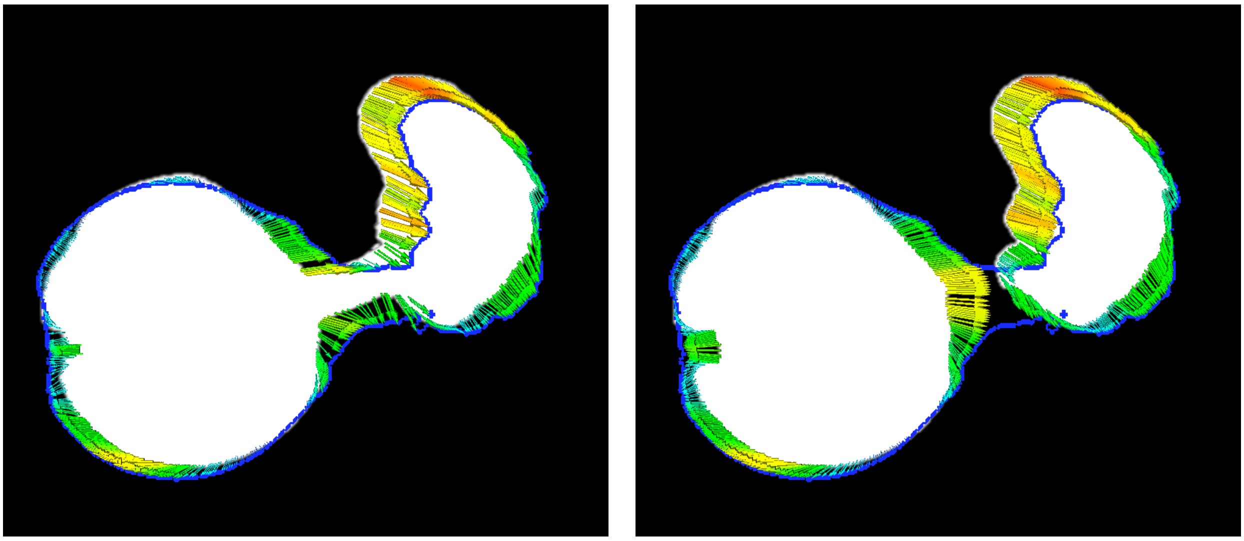
(a) The first curvature map **(b)** The second curvature map

Figure 11: Two curvature maps.



(a) The “invented” cut **(b)** The best match

Figure 12: Two ways of looking at the same registration.



(a) The “invented” connection

(b) The best match

Figure 13: Two ways of looking at the same registration.

5.3 3D Medical Examples

5.3.1 Skull

As previously mentioned, the main motivation for the level-set representation of the surfaces is to be able to register surfaces of arbitrary topology. In one particular project, we need to register hand-segmented surfaces of human skulls. The structure of a skull is extremely complex and contains many fine structures. It is unavoidable that some data is lost during the acquisition process, because of the limited resolution of the acquisition devise or due to segmentation artifacts. Therefore, our data-sets are usually noisy and often do not share the same topology. An example of two typical skull-surfaces is given in figure 14. It can be seen that, especially in the area around the orbita, the data is extremely noisy.

Figure 15 shows a registration result using our algorithm. To illustrate the deformations induced on the surface, we applied a checkerboard texture to the reference skull. We observe that the deformation appear to be very smooth. Visual inspection also reveals that the algorithm found reasonable correspondences, that is corresponding structurs are well matched.

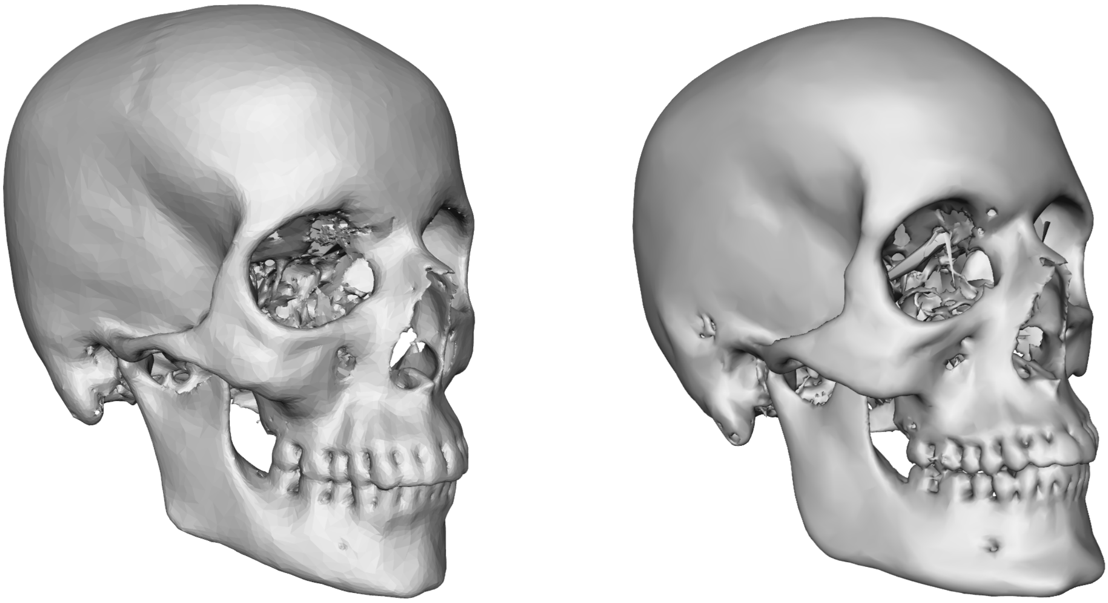


Figure 14: Two examples of hand segmented skulls.

5.3.2 Femur

In this example, we use our extended Demons algorithm to register 3D surfaces of femur bones. Obviously, the visualization and analysis of the registration result is much harder than in the 2D example 5.2. So in Figure 16a, we show only the deformation field

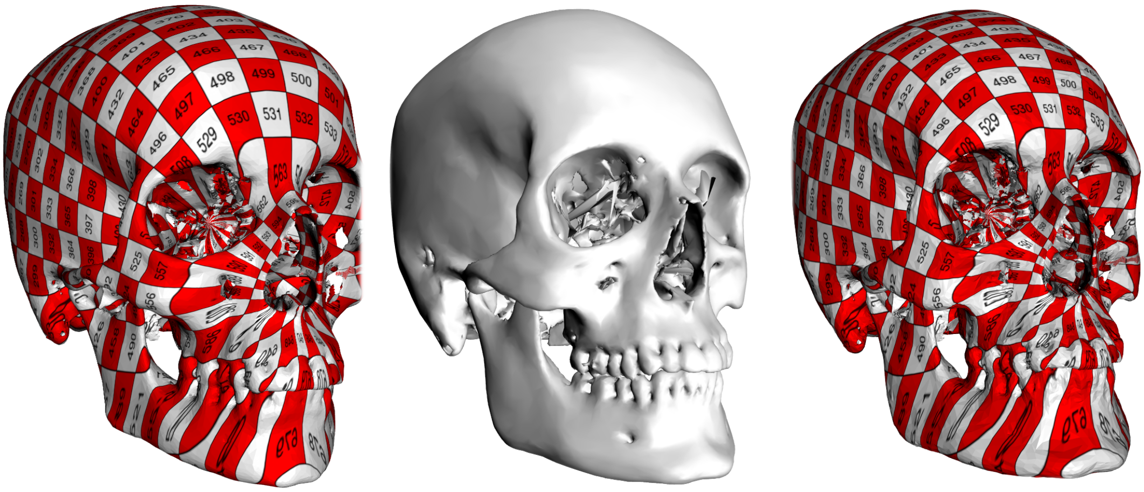
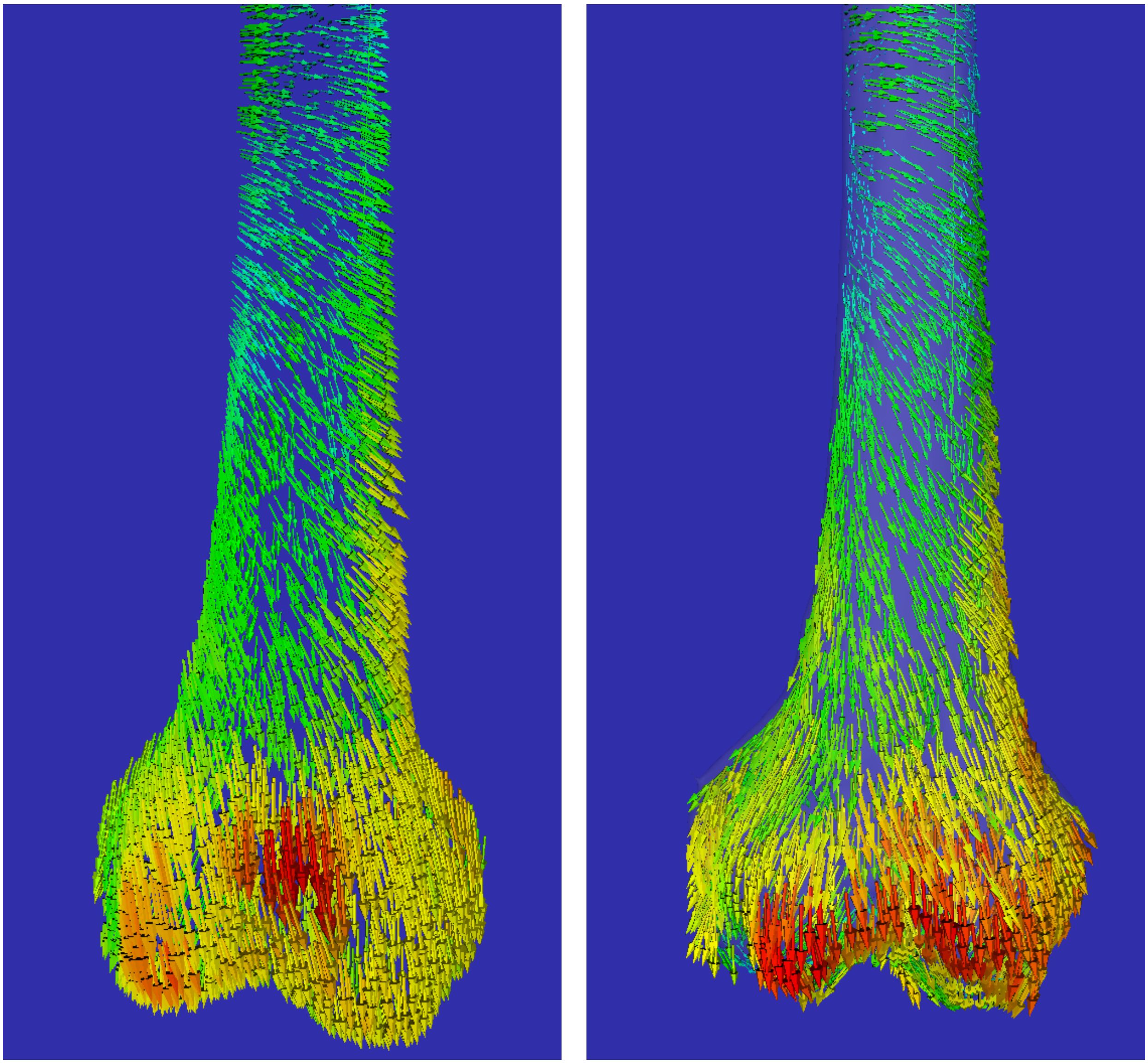


Figure 15: A reference skull (left) is registered onto another skull (center) and deformed to match the other's shape (right)

calculated by the registration algorithm. One bone can be made out as outlined by the tails and the other by the tips of the arrows.

In Figure 16b, only the tangential components of the vector field are shown. As we have explained before, in the original Demons algorithm the tangential component of the deformation field is very small everywhere. Clearly, a sufficient match of the interesting functional feature points is not possible with a deformation field that does not admit a component tangential to the bone surface.


(a) The deformation field

(b) The tangential component

Figure 16: Two ways of looking at the same registration.

6 Conclusion

We have presented a method for the registration of two surfaces by representing the surfaces as the level-set of a distance function and applying Thirion's Demons algorithm for image registration. We showed that the algorithm leads to a very close match of the surfaces. However, our experiments also showed that due to the lack of features on the surface, the correspondences are not always the required ones. Therefore we extended the algorithm to include a curvature term that matches points with similar curvature. We showed that this additional term can be seen as a particular choice of the, so far arbitrarily defined, tangential component of the velocity field in Thirion's Demons algorithm. We also provided an additional interpretation in the variational framework, where the new term becomes an additional cost term in the energy functional.

Further, our experiments also confirmed that the algorithm can be used to register arbitrary complex objects, such a the human skull, and is able to handle topological changes in a natural way.

The largest concern when using level set functions to represent surfaces is that in order to register two n-dimensional surfaces, we have to deal with an $n+1$-dimensional registration problem. In our current implementation, both the distance and curvature maps are stored as 3-dimensional images, which requires large amounts of memory and computation time.

6.1 Future Work

We plan to extend the presented algorithm in two ways, namely to reduce the memory overhead and to investigate additional cost-terms in the energy functional to guide the registration on the surface.

6.1.1 Function Representation

In order to reduce the memory overhead of our method, we to plan implement a more efficient representation of the functions to be registered, in terms of both memory consumption and computation time.

The formulation of the Demons Algorithm as a variational problem, see , leads very naturally to a finite element formulation of the algorithm. An implementation using the finite element method would allow us to represent the functions as adaptive finite element functions, see [2], [24], which could be refined, even to sub-pixel resolution around the zero level-set which represents the surface, while remaining very coarse further away. In this way memory consumption and computation time would be focused on a neighborhood of the surface.

6.1.2 Gradient Fields

In our current algorithm, we aim at simultaneously minimizing the L^2-distance between the two distance maps and of their curvature maps. The curvature maps can be interpreted as the second derivative of the distance functions. We plan to investigate how the first derivative, i.e. the gradient field of the distance maps can also be incorporated into our registration algorithms. The use of gradient fields in a distance measure for image registration has been studied by Droske and Rumpf [7] as well as Haber and Modersitzki [11]. However, their focus lies mostly on multimodal image registration.

7 References

[1] Michel A. Audette, Frank P. Ferrie, and Terry M. Peters. An algorithmic overview of surface registration techniques for medical imaging. *Medical Image Analysis*, 4:2o1–217, 2000.

[2] Dietrich Braess. *Finite elements. Theory, fast solvers and applications in elasticity theory. 3rd corrected and enlarged ed.* Berlin: Springer. xvi, 342 p., 2003.

[3] Pascal Cachier, Xavier Pennec, and Nicholas Ayache. Fast non rigid matching by gradient descent: Study and improvements of the demons algorithm. Technical report, 1999.

[4] Tony F. Chan and Luminita A. Vese. Active contours without edges. *IEEE Trans. Image Process.*, 10(2):266–277, 2001.

[5] Manfredo Perdigao do Carmo. *Differential geometry of curves and surfaces.* Englewood Cliffs, N. J.: Prentice-Hall, Inc. VIII, 503 p., 1976.

[6] M. Droske, M. Meyer, M. Rumpf, and C. Schaller. An adaptive level set method for interactive segmentation of intracranial tumors. *Neurosurgical Research*, 27, 2005.

[7] M. Droske and M. Rumpf. A variational approach to non-rigid morphological registration. *SIAM Appl. Math.*, 64(2):668–687, 2004.

[8] Mariano Giaquinta and Stefan Hildebrandt. *Calculus of variations 1. The Lagrangian formalism.* Grundlehren der Mathematischen Wissenschaften. 310. Berlin: Springer-Verlag. xxix, 475 p. , 1996.

[9] David Gilbarg and Neil S. Trudinger. *Elliptic partial differential equations of second order. Reprint of the 1998 ed.* Classics in Mathematics. Berlin: Springer. xiii, 517 p., 2001.

[10] A. Guimond, A. Roche, N. Ayache, and J. Meunier. Three-dimensional multimodal brain warping using the demons algorithm and adaptive intensity corrections. *Medical Imaging, IEEE Transactions on*, 20(1):58–69, 2001.

[11] Eldad Haber and Jan Modersitzki. *Intensity Gradient Based Registration and Fusion of Multi-modal Images.* 2006.

[12] Joseph V. Hajnal, Derek L.G. Hill, and David J. Hawkes, editors. *Medical Image Registration.* CRC Press, 2001.

[13] P. Hellier, C. Barillot, I. Corouge, B. Gibaud, G. Le Goualher, D.L. Collins, A. Evans, G. Malandain, N. Ayache, G.E. Christensen, and H.J. Johnson. Retrospective evaluation of intersubject brain registration. *Medical Imaging, IEEE Transactions on*, 22(9):1120–1130, 2003.

[14] X Huang, N Paragios, and D. Metaxas. Shape registration in implicit spaces using information theory and free form deformations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2006.

[15] Ravikanth Malladi, James A. Sethian, and Baba C. Vemuri. Shape modeling with front propagation: A level set approach. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(2):158–175, 1995.

[16] P. Maurel, R. Keriven, and O. Faugeras. Reconciling landmarks and level sets. In *Pattern Recognition, 2006. ICPR 2006. 18th International Conference on*, volume 4, pages 69–72, 2006.

[17] Jan Modersitzki. *Numerical Methods for Image Registration.* Oxford Science Publications, 2004.

[18] Stanley Osher and Ronald P. Fedkiw. Level set smethods: An overview and some recent results. Technical report, UCLA Center for Applied Methematics, 2000.

[19] Nikos Paragios, Mikael Rousson, and Visvanathan Ramesh. Matching distance functions: A shape-to-area variational approach for global-to-local registration. In *ECCV (2)*, pages 775–789, 2002.

[20] Nikos Paragios, Mikael Rousson, and Visvanathan Ramesh. Non-rigid registration using distance functions. *Computer Vision and Image Understanding*, 89(2-3):142–165, 2003.

[21] X. Pennec, P. Cachier, and N. Ayache. Understanding the “demon’s algorithm”: 3D non-rigid registration by gradient descent. In C. Taylor and A. Colchester, editors, *Proc. of 2nd Int. Conf. on Medical Image Computing and Computer-Assisted Intervention (MICCAI’99)*, volume 1679 of *LNCS*, pages 597–605, Cambridge, UK, September 1999. Springer Verlag.

[22] Xavier Pennec, Pascal Cachier, and Nicholas Ayache. Understanding the "demon’s algorithm": 3d non-rigid registration by gradient descent. In *MICCAI*, pages 597–605, 1999.

[23] Mikael Rousson, Nikos Paragios, and Rachid Deriche. *Implicit Active Shape Models for 3D Segmentation in MR Imaging.* 2004.

[24] Alfred Schmidt and Kunibert G. Siebert. *Design of adaptive finite element software. The finite element toolbox ALBERTA. With CD-ROM.* Lecture Notes in Computational Science and Engineering 42. Berlin: Springer. xii, 315 p. EUR 64.15 , 2005.

[25] Bernhard Schölkopf, Florian Steinke, and Volker Blanz. Object correspondence as a machine learning problem. In *ICML ’05: Proceedings of the 22nd international conference on Machine learning*, pages 776–783, New York, NY, USA, 2005. ACM Press.

[26] Florian Steinke, Bernhard Schölkopf, and Volker Blanz. Support vector machines for 3d shape processing. *Computer Graphics Forum*, 24(3):285–294, 2005.

[27] J.-P. Thirion. Image matching as a diffusion process: an analogy with maxwell’s demons. *Medical Image Analysis*, 2(3):243–260, 1998.

[28] Andrey N. Tikhonov and Vasiliy Y. Arsenin. *Solutions of ill-posed problems. Translation editor Frity John.* Scripta Series in Mathematics. New York etc.: John Wiley &amp; Sons; Washington, D.C.: V. H. Winston &amp; Sons. XIII, 258 p. 14.80; $ 25.00 , 1977.

[29] B. C. Vemuri, J. Ye, Y. Chen, and C. M. Leonard. Image registration via level-set motion: applications to atlas-based segmentation. *Med Image Anal*, 7(1):1–20, Mar 2003.

[30] He Wang, Lei Dong, Jennifer O'Daniel, Radhe Mohan, Adam S. Garden, K Kian Ang, Deborah A. Kuban, Mark Bonnen, Joe Y. Chang, and Rex Cheung. Validation of an accelerated 'demons' algorithm for deformable image registration in radiation therapy. *Physics in Medicine and Biology*, 50, 21 June 2005.

[31] Zhiyong Xie, Lydia Ng, and James C. Gee. Two algorithms for non-rigid image registration and their evaluation. *Medical Imaging 2003: Image Processing*, 5032(1):157–164, 2003.

[32] Terry S. Yoo, editor. *Insight into Images, Principles and Practise.* A K Peters, ellesley, Massachusetts, 2004.