# Prone to Supine Colon Registration using Shock Graphs and Dynamic Time Warping

Serdar Kemal Balci [1] , Supervisor: Assist. Prof. Burak Acar

June 12, 2006

[1]Serdar Kemal Balci is a member of BUSIM/Volumetric Analysis & Visualization Group of Electrical and Electronics Engineering Department of Bogazici University. http://www.vavlab.ee.boun.edu.tr ( e-mail: serdar.balci@gmail.com )

## Abstract

Computed tomography colonography is a minimally invasive method that allows the evaluation of the colon wall and aids in detecting colonic polyps. Because imperfect cleansing and distension can cause portions of the colon wall to be collapsed, covered with water or retained stool, patients are scanned in both prone and supine positions. Both reading efficiency and computer aided detection of CTC images can be improved by accurate registration of data from the supine and prone positions. We propose an automatic registration algorithm based on matching the shock graphs of the complementary supine and prone data using dynamic time warping algorithm. Shock graphs provide a compact representation of shapes that can be used for registration and segmentation. In order to compute the shock graph of the segmented colon data we use Hamilton-Jacobi skeletons proposed by Siddiqi et al. The shock graphs of the complementary supine and prone CTC data sets will be registered using dynamic time warping algorithm, which estimates a nonlinear warping function from the supine data to the prone data based on minimizing a cost function subject to some proper constraints.

# Contents

# List of Figures

# 1 Introduction

Compared to endoscopy, computed tomography colonography (CTC) is a minimally invasive method for the examination of the colon CT volume data [1] [34]. In computed tomography a radiologist views a sequence of CT images in order to detect polyps. Detecting polyps in CT data allow the medical doctors for early diagnosis of colorectal cancer which is a common form of a cancer associated with high mortality rates. The standard imaging process consists of colon cleansing and air insufflation, followed by CT imaging of the abdomen/pelvis. Detection of polyps from a single scan is a difficult task as stools adhered to colon wall could resemble polyps and the location of polyps could be covered with liquid present in the colon. In order to detect polyps reliably two CT scans is obtained one in the supine position and one in the prone position. However, in order to benefit from these two scans, determining the corresponding positions in both scans is necessary. As the colon wall is a nonrigid structure its position and shape changes nonlinearly between the scans. Therefore, registering the supine data to the prone data is a necessary but difficult task. In computed tomography, radiologists are provided with the 3D rendering of the colon wall which is called virtual colonoscopy. In virtual colonoscopy the user inspects the colon wall by viewing 3D images taken on a centerline path through the colon. We aim at registering two centerline paths corresponding to supine and prone views by making use of dynamic time warping algorithm; therefore, allowing the radiologists to view the corresponding positions in both scans simultaneously. Registering the whole CT data might seem a better choice for our task; however, its computational complexity would make it a time consuming task.

Our method to solve the registration problem consists of two steps. First, we compute the shock graph of the colon in order to obtain the medial axis through the colon and radius values at each point on the medial axis. Afterwards we apply dynamic time warping algorithm to register the two shock graphs obtained from supine and prone scans. In the following paragraphs we will mention the methods available in literature to compute the shock graph of a shape.

The shock graph of a shape can be formed from the skeleton of a shape by writing the skeleton points and the corresponding radius of the maximal spheres into a graph. The graph formed this way can be used for registration purposes. The usual definition of the skeleton of a 3D shape is that it is the locus of the centers of the maximal spheres contained in the shape. However, constructing skeletons by drawing maximal spheres at all points inside the shape is impractical as determining the tangency between a point and a noisy shape contour is problematic. In the literature there are three main approaches to determine the skeleton points.

Blum(1973) [4] proposed a grassfire model to compute the skeleton of a shape. In Blums grassfire technique a shape is considered to be filled with dry grass and a fire is started at the shape boundary. The time of the arrival of the grassfire front at a point equals the distance of that point from the shape boundary. The skeleton points correspond to singular points which are formed when the fronts from two or more directions meet and extinguish themselves. The problem than reduces to locating these singular points.

Another method is based on a thinning process to realize the Blums grassfire formulation. Layers are peeled away from the shape boundary while retaining some special points. Although these methods may preserve the topology of the shape and are efficient to implement, these methods fail to localize the skeleton points accurately. As a result they can only provide a coarse approximation to the shapes when the shape is reconstructed from the skeleton [2] [5]. A third approach is to compute the skeleton using Voronoi diagrams. This approach ensure homotopy between objects and their skeletons and accurately localize skeleton points. However, the results are not invarient under Euclidean transforms [8]. In our project the focus will be on Blums grassfire model and we implement the method proposed by Siddiqi to compute the skeleton points.

Blum introduced skeletons in 1973. Osher and Sethian [20] proposed algorithms based on Hamilton Jacobi formulations for propagating fronts moving with curvature dependent speeds in 1988. Malladi [22] used level set approaches for front propagation in 1995. Siddiqi [15] proposed Hamilton-Jacobi Skeletons in

2002. In 2005 Shah [29] [28] introduced gray skeletons which are shape skeletons with a significance number attached to each of its points. Shah uses the angle between the front normals as the significance number which is related to the quantities such as the jump in the gradient of the distance function and the velocity of the grassfire along the skeleton. Siddiqi proposed that skeletons computed using Blums grassfire technique can be used for forming shock graphs [13]. These shock graphs can be used for shape recognition and indexing [10] [31] [32].

Level set methods are Eulerian in nature as they are restricted to grid points where the locations are fixed. Level set approaches outlined above are capable of preserving the singularities that develop during the curve evolution process. However, these methods do not provide a method to detect the singular points. The singular points may be calculated following a level set based evolution process and computing maximum curvature points at each iteration. However, such a method is sensitive to noise and the resulting skeletons are not guaranteed to be a connected set.

Siddiqi [15] uses Hamiltonian formalism for simulating the eikonal equation and offers a method to detect the shock points. Based on Blums grassfire flow model he computes a measure of average outward flux of the vector field underlying the Hamiltonian system. He shows that as the region over which this flux is computed shrinks to a point, the measure has different limiting behaviors depending whether or not that point is singular. Thus he provides an effective way of distinguishing singular points from the nonsingular ones. A simple threshold over the computed flux is used to coarsely locate the singular points. Afterwards a homotopy preserving thinning process is applied to finalize the skeleton computation [9] [21].

In this project the algorithm proposed by Siddiqi will be implemented to compute the skeleton and the shock graph of the segmented CTC data. The shock graphs of the complementary supine and prone CTC data sets will be registered using dynamic time warping algorithm [16] [30] [11].

# 2 Related Work

In commercial virtual colonoscopy applications radiologists are provided with both prone and supine scans. However, the registration has to be done manually by matching some anatomical points to each other.

Acar et al. [17] [3] introduced an automatic registration algorithm which is based on heuristics. They first compute the centerline of the colon and determine the major local extreme points and register the points in one data set to the other data set using linear stretching and shrinking operations. They do these stretching and shrinking operation recursively until all local extreme points are detected. They also evaluate the registration algorithm on 24 patient cases. By doing the path registration, the mean misalignment distance between prone and supine identical anatomic landmarks was reduced from 47.08 to 12.66mm, a 73% improvement. This algorithm relies on linear stretching operations which limited in its capability to register the two data sets. An algorithm taking into account the global structure of the data sets might yield better results.

Nain et al. [19] [7] improved Acar et al. [17] [3] technique by taking into account the global structure of the centerline of colon. Firstly, they compute the centerline of the colon by considering a heat equation. They suppose that the boundary of the colon surface to be held at a constant temperature initially. Then they compute the temperature distribution around the colon surface. The centerline is formed by the centers of mass of the loops which are the level sets of the temperature distribution around the colon surface. They also write the radius values corresponding to each point in the centerline. Supine to prone matching is done by using dynamic programming where they minimize the sum of Euclidean distance between the points in the two data sets. They claim that they achieved 94% frame matching in virtual colonoscopy. However, they did not evaluate their algorithm to match anatomical structures as opposed to Acar et al.'s [17] method.

In a sequence of papers Bouix et al. [25] [23] [24] formulated automatic centerline extraction using Siddiqi et al.'s [15] Hamilton-Jacobi Skeletons. Bouix et al. compute the 3D skeleton of the segmented colon data. To find the 3D skele-

ton of the colon Bouix uses Siddiqi et al. Hamiltonian formulation of Blum's [4] grassfire model. The algorithm proposed by Bouix provides a robust and efficient method for automatic centerline extraction. By writing the radius of maximal spheres to each skeletal point they also form a data set which they call shock graph.

In our approach to the registration problem we will attempt to combine Bouix et al's medial axis extraction algorithm with dynamic time warping algorithm. By using Bouix et al's medial axis extraction algorithm we will obtain the shock graph of the colon data which we will use as an input to the dynamic time warping algorithm. The dynamic time warping algorithm will allow us to find the optimum warping function from the shock graph of the supine data to the shock graph of the prone data by minimizing a proper cost function.

# 3 Theory

In this section we will present the theoretical background of the methods and algorithms that we used in our registration method. Our registration method consists of two parts. First, extracting the centerline of the colon data using Hamilton-Jacobi Skeletons [15] and then matching the data obtained from the first part using dynamic time warping algorithm.

Firstly, we will review segmentation techniques and state our segmentation algorithm for extracting the boundaries of the colon data. Then we will introduce Blum's Grassfire Model and state two approaches to this model, namely level set methods and Hamiltonian formulation. Afterwards, we give the algorithm for computing Hamilton-Jacobi skeletons. We will conclude this part by giving the theoretical background for the dynamic time warping algorithm we used.

## 3.1 Segmentation

As the Hamilton-Jacobi Skeletonization algorithm works on binary images, it is necessary to segment the colon wall from the CT data before computing the shock graph of the colon. In literature there are eminent segmentation algorithms based on active contours to segment region of interest from medical images [18] [33] [14] [6] [27] [26] . These methods rely on curve evolution techniques and provide a reliable and robust segmentation of the regions of interest. In active contour models, a curve is initialized either inside or outside the region of interest. Afterwards the curve moves to the desired structures in the image according to an edge detector function which is derived from the data itself while maintaining the smoothness and closeness of the curve. To implement these active contour models level set methods formalized by Osher and Sethian [20] are used. The use of level set methods allow the topological changes during the segmentation to be handled implicitly.

The segmentation methods based on active contour models and level set methods are the most reliable models to segment region of interest from a CT data. However, as the amount of noise present inside the colon wall is within an acceptable level, we used a seed based connected threshold region growing algorithm. As the region inside the colon is insufflated with air during the CT scan a simple threshold based segmentation gives satisfactory result for our purposes. The algorithm begins from a voxel inside the colon called "seed" and adds neighboring voxels if they satisfy certain criteria. A criteria based on the interval of intensity values and on the connectedness of the region of interest is used. Using the region growing technique we achieved a segmentation which is considerably faster than the methods based on active contours and which gave comparable results for the case of the region inside the colon. After segmenting the colon from the CT data we computed its shock graph using the Hamilton Jacobi Skeletons which we explain more in detail in the following sections.

## 3.2 Blum's Grassfire Model

In 1973 Blum [4] proposed a grassfire model to compute the skeleton of a shape. In Blums grassfire technique a shape is considered to be filled with dry grass and a fire is started at the shape boundary. The time of the arrival of the grassfire front at a point equals the distance of that point from the shape boundary. The skeleton points correspond to singular points which are formed when the fronts from two or more directions meet and extinguish themselves. The problem than reduces to locating these singular points.

In Blum's grassfire model each point on the shape boundary is moving with unit speed in the inward normal direction. In the following we formulate Blum's grassfire model using concepts from curve evolution methods.

Let $C$ be the set of curves in $R^2$ given by:

$$C = \left\{ c : [a,b] \rightarrow \Omega, \;\; piecewise\, C^1, c(a) = c(b) \right\} \tag{1}$$

Also let $c'$ and $c''$ denote the first and second derivatives of $c$, respectively:

$$\begin{array}{rcl} c(q) & = & (c_1(q), c_2(q), c_3(q)) \\ c'(q) & = & \left( \frac{dc_1}{dq}, \frac{dc_2}{dq}, \frac{dc_3}{dq} \right) \\ |c'(q)| & = & \sqrt{\left( \frac{dc_1}{dq} \right)^2 + \left( \frac{dc_1}{dq} \right)^2 + \left( \frac{dc_1}{dq} \right)^2} \end{array} \tag{2}$$

Then the evolution of the grassfire front can be written as

$$\left\{ \begin{array}{c} \frac{\partial c}{\partial t} = N \\ c(0,q) = c_0(q) \end{array} \right. \tag{3}$$

which is also called as eikonal equation. The eikonal equation in (3) states that the curve $c(t,q)$ moves along its inward normal direction with unit speed. To

simulate the eikonal equation we will present the level set formulation by Osher and Sethian [20] and Hamiltonian formulation by Siddiqi et al. [14].

### 3.2.1 Level Set Formulation

Level set methods as introduced by Osher-Sethian [20] [27] provide an efficient and stable algorithm to solve the eikonal equation (3). The level set methods introduced by Osher-Sethian simulate a generalized version of equation (3).

$$\begin{cases} \frac{\partial}{\partial t} = FN \\ c(0,q) = c_0(q) \end{cases} \tag{4}$$

Equation (4) states that the curve $c(t,q)$ moves along its normal with a speed $F$, which may depend on $t, c, c', c''$. The level set formulation is based on the observation due to Osher-Sethian [**?**] that a curve can be seen as the zero level set of a function in higher dimension. For example, a curve in $R^2$ can be represented as the zero-level line of a function $R^2 \rightarrow R$. Suppose that there exists a function such that $u : R^+ \times R \rightarrow$ such that

$$u\,(t, c(t,q)) = 0,\ \forall q, \forall t \geq 0 \tag{5}$$

Then by differentiating (5) with respect to $t$, we get

$$\frac{\partial u}{\partial t} + \left\langle \nabla u, \frac{\partial c}{\partial t} \right\rangle = 0 \tag{6}$$

Setting the speed term in (4) into (6)

$$\frac{\partial u}{\partial t} + \langle \nabla u, FN \rangle = 0 \tag{7}$$

The unit normal is given as

$$N = -\frac{\nabla u}{|\nabla u|} \tag{8}$$

Putting this into (7)

$$\frac{\partial u}{\partial t}(t, c(t,q)) = F|\nabla u(t, c(t,q))| \tag{9}$$

This equation is valid only for the zero-level set of $u$ . But, Osher-Sethian show in [?] that $u$ can be regarded as defined on the whole domain $R^+ \times \Omega$ . Then the following PDE can be solved

$$\frac{\partial u}{\partial t}(t,x) = F|\nabla u(t,x)| \tag{10}$$

For $t > 0$ and $x \in \Omega$ if $F$ is defined on the whole space. Then once $u$ is calculated on $R^+ \times \Omega$ the curve $c$ can be obtained by extracting the zero level set of $u$ . The normal derivative is chosen to vanish on the boundary and $u$ is initialized to be the signed distance function to the initial curve $c_0$. Then, Osher-Sethian [?] give the following final model:

$$\begin{cases} \frac{\partial u}{\partial t}(t,x) = F|\nabla u(t,x)| & for(t,x) \in ]0, \inf[ \times \Omega \\ u(0,x) = \bar{d}(x, c_0) & (\bar{d}\, signed\, distance) \\ \frac{\partial u}{\partial N} = 0 & for(t,x) \in ]0, \inf[ \times \Omega \end{cases} \tag{11}$$

If the curve motion can be expressed as a velocity along the normal direction of the curve, the model in (11) is useful from several points of view. Firstly, the evolving function $u(t,x)$ remains a function during evolution as long as $F$ is smooth. However, the level set $u = 0$ , so the front $c(t,q)$, may change topology, break, merge as $u$ evolves. This is the main advantage of level set formulation as the topological changes should not be taken into account explicitly. Secondly, for

numerical approximations, a fixed discrete grid in the spatial domain and finite-difference approximations for spatial and temporal derivatives can be used. A third advantage is that geometric elements of the front such as the normal vector and the curvature can be expressed with respect to $u$. In addition, level set methods can be extended to higher dimensions.

The eikonal equation (3) can be obtained by setting $F = 1$ in equation (11). Then the level set solution to the eikonal equation can be given as

$$\begin{cases} \frac{\partial u}{\partial t}(t,x) = |\nabla u(t,x)| & for(t,x) \in ]0,\inf[\times\Omega \\ u(0,x) = \bar{d}(x,c_0) & (\bar{d}\, signed\, distance) \\ \frac{\partial u}{\partial N} = 0 & for(t,x) \in ]0,\inf[\times\Omega \end{cases} \tag{12}$$

The level set methods introduced by Osher-Sethian gives the governing equations(12) to simulate the eikonal equation (3). Once the initial surface in 3D is given, then by using equation (12) the intermediate steps of curve evolution can be obtained. Then, according to Blum's grassfire model determining the skeletal points reduces to finding singular points which form during curve evolution. Singular points during curve evolution correspond to points with maximum curvature values at each time step. Determining points with maximum curvature values is a difficult task because of the smooth contours present in the shape. Therefore, the level set methods by Osher-Sethian preserve the singular points which form during the simulation of the eikonal equation (3); however, they do not provide us with a direct method of determining the singular points, hence the skeletal points.

### 3.2.2 Hamiltonian Formulation

Siddiqi et al. [14] [12] formulate the eikonal equation (3) using concepts from Hamiltonian physics. They also define a criteria to accurately localize the singular points which form during the curve evolution. They suggest that a measurement of the net outward flux of the of the gradient vector field of the Euclidean distance function can be used to localize points where conservation of the energy principle

is violated. As the singular points correspond to such points they provide a criteria to effectively localize singular points. Siddiqi et al. begin their formulation by writing an action function to be minimized as follows

$$Cq_0,t_0(\mathbf{q},t) = \int_\gamma \mathcal{L}dt \tag{13}$$

In this Lagrangian formulation (13) the independent variables are the coordinates $\mathbf{q}$ of particles and their velocities $\dot{\mathbf{q}}$. For example, in the context of the eikonal equation (3) these would be the positions of points along the curve $C$ and their associated velocities $N$. Each particle follows the path of least action in reaching a future location at a future time. In equation (13) $\gamma$ is an extremal curve connecting the points $(q_0,t_0)$ and $(q,t)$ and $L(\mathbf{q},\dot{\mathbf{q}})$ is the Lagrangian. In other words, of all possible paths connecting $(q_0,t_0)$ and $(q,t)$, the trajectory $\gamma$ followed by the particle is the one that minimizes the action function $Cq_0,t_0$. The associated Euler-Lagrange equation with (13) is

$$\frac{d}{dt}\frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} - \frac{\partial \mathcal{L}}{\partial \mathbf{q}} = 0 \tag{14}$$

and the momenta are derived quantities given by

$$\mathbf{p} = \frac{\partial \mathcal{L}}{\partial \dot{\mathbf{q}}} \tag{15}$$

For the case of a front moving with constant speed, the action function being minimized is the Euclidean length and hence $Cq_0,t_0$ can be viewed as a Euclidean distance function from the initial curve $C_0$. Furthermore, the magnitude of its gradient,$\nabla Cq_0,t_0$, is identical to 1 in its smooth regime. The Lagrangian associated with the action function minimized (13) is given by

$$\begin{aligned} \mathcal{L} &= \|\tfrac{\partial \gamma}{\partial t}\| \\ &= \|\dot{\mathbf{q}}\| \end{aligned} \tag{16}$$

As the action function minimizes the Euclidean length we redefine $Cq_0, t_0$ as follows

$$D = Cq_0, t_0 \tag{17}$$

where D is the Euclidean distance function to the initial contour $C_0$ and $\|\nabla D\| = 1$ in its smooth regime.

After defining the Lagrangian associated with the eikonal equation, Siddiqi et al. replaces Lagrangian $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}})$ with Hamiltonian $\mathcal{H}(\mathbf{q}, \dot{\mathbf{q}})$ in order to let the velocities become derived quantities. This way the equations become linear and hence trivial to simulate numerically. The transformation from Lagrangian to Hamiltonian can be done using Legendre transformation, which is given by

$$\mathcal{H}(\mathbf{q}, \dot{\mathbf{q}}) = \mathbf{p} \cdot \dot{\mathbf{q}} - \mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) \tag{18}$$

With $\mathbf{q} = (x, y, z)$ and $\mathbf{p} = (D_x, D_y, D_z)$, the curve $\tilde{C} \subset R^6$ associated with $C \subset R^3$ evolving according to the eikonal equation (3) is given by

$$\tilde{C} := \{(x, y, z, D_x, D_y, D_z) : (x, y, z) \in C,\ D_x^2 + D_y^2 + D_z^2 = 1,\ \mathbf{p} \cdot \dot{\mathbf{q}} = 1\} \tag{19}$$

where $\mathbf{p} \cdot \dot{\mathbf{q}} = 1$ results from Huygens principle.

The Hamiltonian function can be obtained by applying Legendre transformation to Lagrangian $\mathcal{L}(\mathbf{q}, \dot{\mathbf{q}}) = \|\dot{\mathbf{q}}\|$.

$$\begin{aligned} \mathcal{H} &= \mathbf{p} \cdot \dot{\mathbf{q}} - \mathcal{L} \\ &= 1 - \left(D_x^2 + D_y^2 + D_z^2\right)^{1/2} \end{aligned} \tag{20}$$

The associated Hamiltonian system is

$$\begin{aligned} \dot{\mathbf{p}} &= -\frac{\partial \mathcal{H}}{\partial \mathbf{q}} & \dot{\mathbf{q}} &= \frac{\partial \mathcal{H}}{\partial \mathbf{p}} \\ &= (0,0,0) & &= -(D_x, D_y, D_z) \end{aligned} \tag{21}$$

$\tilde{C}$ can be evolved under this system of equations, with $\tilde{C}(t) \subset R^6$ denoting the resulting contact surface. The projection of $\tilde{C}(t)$ onto $R^3$ will then give the parallel evolution of $C$ at time $t, C(t)$. The interpretation of equation (21) is quite intuitive. The gradient vector field $\mathbf{p}$ does not change with time and the points on the boundary of the surface move in the direction of the inward normal with unit velocity.A variety of methods including level set methods can be used to implement the equation (21).

The important part of equation (21) is that the formation of shocks can be made explicit. The key idea is to use a measure of the average outward flux of the vector field of $\dot{\mathbf{q}}$. Siddiqi et al. define the average outward flux as the outward flux through the boundary of a region containing the point, normalized by the area of the boundary

$$\frac{\int_{\delta R} \langle \dot{\mathbf{q}}, N \rangle ds}{area(\delta R)} \tag{22}$$

Where $ds$ is an element of the bounding surface $\delta R$ of the region $R$ and $N$ is the unit outward normal at each point of the surface. Using the divergence theorem

$$\int_{\delta R} \langle \dot{\mathbf{q}}, N \rangle ds = \int_R div(\dot{\mathbf{q}}) dv \tag{23}$$

where $dv$ is a volume element. Thus the outward flux is related to the divergence in the following way

$$div(\dot{\mathbf{q}}) = \lim_{\triangle v \to 0} \frac{\int_{\delta R} \langle \dot{\mathbf{q}}, N \rangle ds}{\triangle v} \tag{24}$$

For a volume with an enclosed surface an excess of outward of inward flow

through the surface indicates the presence of a source or a sink, respectively, in the volume. The divergence of the vector field at a point $div(\dot{\mathbf{q}})$ is the net outward flux per unit volume as the as the volume shrinks to zero. Equation (24) cannot be used where the vector field is singular as the it is not differentiable at singular points. However, we are interested in singular points; therefore,equation (24) provides a means to detect the singular points. According to equation (24) the net outward flux through the surface which bounds a finite volume is just the volume integral of the divergence of the vector field within that volume. Therefore, location where the flux is negative, and hence the energy is lost correspond to sinks or skeletal points.

A finite difference approximation to equation (24) can be used to find points where the flux is negative. Using a negative threshold over the average outward flux of the vector field will give a rough localization of the singular points. As a result Siddiqi et al. introduced a robust and efficient way of evolving curves in 3D using Hamiltonian formalism to the eikonal equation. In addition, they defined a criteria based on the average outward flux of the distance transform which provides an explicit method to localize skeletal points.

## 3.3 Hamilton-Jacobi Skeletons

Hamilton-Jacobi Skeletons [15] can be computed using the Hamiltonian formulation to the eikonal equation. Hamilton-Jacobi skeleton are based on applying a threshold to the divergence defined in equation (24) and have a number of useful properties such as:

- they are a thin set, or they do not contain interior points,
- they are homotopic to the original shape,
- they are invariant under Euclidean transformations such as rotations and translations

**Algorithm1:** Average Outward Flux

Compute the Euclidean distance transform $D$ of the object
Compute the gradient vector field $\nabla D$
Compute the average outward flux of $\nabla D$ using Eq. (21)
**for**(each point $\mathbf{x}$ in the interior of the object)
$Flux(\mathbf{x_i}) = \frac{1}{n}\sum_{i=1}^{i=26}\langle \hat{\mathbf{N}}_i, \nabla D(\mathbf{x}_i)\rangle$
(where $x_i$ is a 26-neighbor of $\mathbf{x}$ and $\hat{\mathbf{N}}_i$ is the outward normal at $\mathbf{x}_i$ of the unit sphere centered at $x_i$ )

Figure 1: **Algorithm1** : Computing average outward flux

- given the radius of the maximal inscribed circle or sphere associated with each skeletal point the object can be reconstructed exactly

The skeletal points can be localized by using finite difference approximation to equation (24) and applying a threshold to the average outward flux field.

$$Flux(\mathbf{x}) = \frac{1}{n}\sum_{i=1}^{i=26}\langle \hat{\mathbf{N}}_i, \nabla D(\mathbf{x}_i)\rangle \tag{25}$$

where $x_i$ is a 26-neighbor of $\mathbf{x}$ and $\hat{\mathbf{N}}_i$ is the outward normal at $\mathbf{x}_i$ of the unit sphere centered at $x_i$. Figure 1 gives the algorithm to compute the average outward flux.

However, as the computation of equation (24) is local, global properties such as the preservation of the object's topology are not ensured. In addition the skeleton obtained by applying a threshold on the divergence field is not guaranteed to be a thin set. Therefore the divergence computation has be combined with topology preserving thinning process, such that as many points as possible are removed without altering the object's topology.

A point is called "simple" if its removal does not change the topology of the object. Hence in 3D, its removal must not disconnect the object, create a hole,

or create a cavity. Siddiqi et al. [15] use a formal definition for a simple point introduced by Malandain et al. [9]. Before giving the algorithm for the topology preserving thinning algorithm we present a review of basic concepts in digital topology.

### 3.3.1 Digital Topology

In 3D digital topology, the input is a binary image stored in a 3D array. In a 3D cubic lattice, a point is viewed as a unit cube with 6 faces, 12 edges and 8 vertices. For each point three types of neighborhoods are defined as

**6-neighbors:** two points are 6-neighbors if they share a face

**18-neighbors:** two points are 18 neighbors if they share a face or an edge

**26-neighbors:** two points are 26-neighbors if they share a face, an edge or a vertex

Using the definitions above Malandain et al. [9] defines three types of connectivity, denoted by "n-connectivity" where $n \in \{6, 18, 26\}$. Malandain et al. also define "n-neighborhoods" for $x$, called $N_n(x)$.(see Figure 2 ). A n-neighborhood without its central point is defined as $N_n^* = N_n\{x\}$. To characterize a point as a simple point Malandain et al. also gives the following definitions.

- An object **A** is n-adjacent to an object **B**, if there exist two points $x \in \mathbf{A}$ and $y \in \mathbf{B}$ such that $x$ is an n-neighbor of $y$
- An "n-path " from $x_1$ to $x_k$ is a sequence of points $x_1, x_2, \ldots, x_k$, such that for all $x_i$, $1 < i \leq < k$,$x_{i-1}$ is n-adjacent to $x_i$.
- An object represented by a set of points $O$ is n-connected, if every pair of points $(x_i, x_j) \in O \times O$, there is an n-path from $x_i$ to $x_j$.

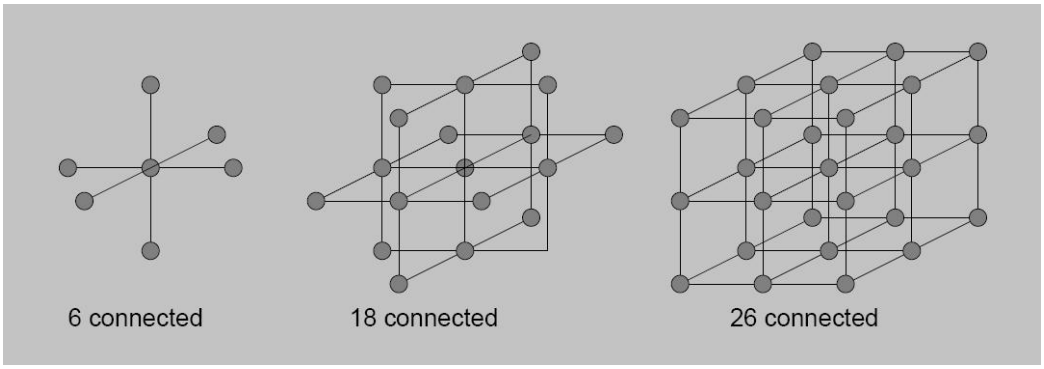



Figure 2: 6-neighborhoods, 18-neighborhoods and 26-neighborhoods in a cubic lattice [25].

Based on these definitions, Malandain et al. provide a topological classification of a point $x$ in a cubic lattice by computing two numbers.

1. $C^*$: the number of 26-connected components 26-adjacent to $x$ in $O \cap N_{26}^*$
2. $\bar{C}$: the number of 6-connected components 6-adjacent to $x$ in $\bar{O} \cap N_{18}$

A point in 3D cubic lattice is simple if it satisfies the following criteria

$$\begin{aligned} C^* &= 1 \\ \bar{C} &= 1 \end{aligned} \tag{26}$$

In addition, in 3D cubic lattice an end point of a 26-connected curve is defined as a point $x$ whose 26-neighborhood $O \cap N26^*$ contains only one object which is in the of the object.

### 3.3.2 Distance Ordered Thinning

For a volume with an enclosed surface, an excess of inward flow through the surface indicates the presence of a sink in the volume. These sink points correspond

exactly to skeletal points. According to equation (25) the total outward flux of the gradient vector field of the Euclidean distance function is negative at such points; therefore, they correspond to skeletal points. In addition, the magnitude of the total outward flux is proportional to the amount of energy absorbed. Therefore, the points where the outward flux is below a threshold value can be used a candidates for skeletal points. However, using only a threshold value does not assure the preservation of the object's topology the thinness of the result. Therefore the flux computation has be combined with topology preserving thinning process, such that as many points as possible are removed without altering the object's topology.

The solution proposed by [21] is to order the thinning process such that simple points closest to the surface are removed first. The simple points closest to the surface correspond to points having the lowest Euclidean distance $D(x)$ values. This distance ordered thinning algorithm can be combined with the flux value obtained from (25) such that points having a value smaller than a threshold value are considered as candidate skeletal points. The thinning process should stop when all the points left are not simple or are not end points. End points should be handled explicitly as they are simple points; but, they must not be removed during the thinning process. Using a distance ordered thinning algorithm combined with flux value obtained from (25), preserving the end points in the set assure that the resulting set is thin. In addition, the thinning algorithm preserves the topology of the skeleton. To make the thinning algorithm efficient a heap implementation is used. Firstly, the simple points on the shape boundary is inserted into a list. Then for all points in this list the neighboring simple points are inserted into heap if they are simple. Then the points in the heap which has the lowest distance from the surface is removed first. The implementation of Heap makes the thinning algorithm considerably fast as inserting and removing from a heap can be done in $O(\log(n))$. Figure 3 summaries the distance ordered thinning algorithm.

**Algorithm2:** Distance Ordered Homotopic Thinning

**for** (each point **x** on the boundary of the object)
    **if** (**x** is simple ) **then**
        insert into Heap (**x**) with $D(\mathbf{x})$ as the sorting key for insertion

**while** (Heapsize $> 0$)
    **x** = Remove Maximum from Heap
    **if** (**x** is simple) **then**
        **if** (**x** is an end point of a 3D curve) and (Flux(**x**) $<$ Thresh) **then**
            mark **x** as a skeletal point
    **else**
        Remove **x** from the Heap
    **for** (all neighbors **y** of **x**)
        **if** (**y** is simple) **then**
            insert into Heap (**x**) with $D(\mathbf{y})$ as the sorting key for insertion

Figure 3: **Algorithm2** : Distance Ordered Homotopic Thinning

### 3.3.3 Obtaining the Shock Graph

The result of the distance ordered thinning algorithm gives the skeleton of the shape. However, the skeleton might consists of several branches. In order to obtain the centerline through the colon pruning the skeleton is necessary in order to obtain the centerline. Pruning the skeleton corresponds to obtaining the longest path in the skeleton. As the longest path should end at end end point, we begin by marking all end points in the skeleton. To find the longest path we begin by selecting an end point at random. Then, we compute the endpoint which has the longest distance along the path of the skeleton to the selected point. As the skeleton of the colon is guaranteed to be a thin set without loops, the found end point is assured to be an end point of the longest path. Then, we compute the farthest end point to the found end point. The longest path through the colon, i.e. the centerline, is path connecting these last two endpoints.

As the region obtained from segmentation could contain disconnected regions, the centerline has to be computed for all connected sets. Afterwards we obtain the centerline of the colon by combining the centerlines of all connected sets. The locations between the successive centerlines are interpolated linearly in order to obtain a single connected centerline through the colon. The shock graph of the colon can be formed by writing radius values obtained from the distance transform of the segmented shape to each point in the centerline. Again the radius values of missing parts are linearly interpolated to give a single connected shock graph. Figure 4 gives the pseudo-code of the algorithm for obtaining the shock graph of the colon.

## 3.4 Dynamic Time Warping

To register the shock graphs of supine and prone CTC data sets a warping function from the supine data to the prone data will be estimated based on minimizing a cost function. The minimization problem can be stated as follows. Both the supine and prone data sets will be considered as two sequences:

**Algorithm3:** Obtaining Centerline and Shock Graph

**for** (all connected sets in the skeleton)
    Find end points in the connected set
    **x** = PickAnEndPoint
     y is the farthest point from x according to its distance along skeleton
    **y** = FarthestPoint( **x** )
    **z** = FarthestPoint( **y** )
     c is the centerline in the current dataset
    **c** = GetConnectingPath( **y**, **z** )
     s is the shock graph
    **s** = WriteRadiusValues( **c** )

C is the connection of centerlines in the subsets
**C** = connect( $\mathbf{c}_1$, $\mathbf{c}_2$, ... )
**S** = connect( $\mathbf{s}_1$, $\mathbf{s}_2$, ... )

interpolate the broken parts
**C** = LinearInterpolateMissingParts( **C** )
**S** = LinearInterpolateMissingParts( **S** )

Figure 4: **Algorithm3** : Obtaining Centerline and Shock Graph

$$\begin{pmatrix} f_{x,y,z,r} & i = 1,\ldots,M \\ g_{x,y,z,r} & i = 1,\ldots,N \end{pmatrix} \tag{27}$$

where $x, y, z$ denote $x, y, x$ coordinates of skeletal points and $r$ denotes the radius at that position. $M, N$ are the lengths of the supine and prone data sets respectively.

The distance between two elements of $f_{x,y,z,r}$ and $g_{x,y,z,r}$ can be given by:

$$d(i,j) = \begin{cases} \alpha\left[(f_x(i) - g_x(j))^2 + (f_y(i) - g_y(j))^2 + (f_z(i) - g_z(j))^2\right] + \\ (1-\alpha)\,(f_r(i) - g_r(j))^2 \end{cases} \tag{28}$$

where the subindices $x, y, z, r$ of $f$ and $g$ denote the x,y,z coordinates of the skeleton and the maximal fitting sphere to the corresponding location, respectively. $\alpha$ is the weighting coefficient which is selected to yield optimum results. The dynamic time warping algorithm computes the best possible warp between $f_{x,y,z,r}$ and $g_{x,y,z,r}$ by minimizing the total distance [11] [30]. The alignment warp $\Phi$ relates the indices of the two sequences $f_{x,y,z,r}$ and $g_{x,y,z,r}$

$$\begin{aligned} \Phi(k) &= \left(\Phi_f(k), \Phi_g(k)\right) \\ &\quad where \\ k &= 1,2,\ldots,K \\ 1 \leq \Phi_f &\leq M \qquad \Phi_f \in \mathbf{N} \\ 1 \leq \Phi_f &\leq N \qquad \Phi_f \in \mathbf{N} \end{aligned} \tag{29}$$

The alignment warp $\Phi$ aligns the indices of $f_{x,y,z,r}$ and $g_{x,y,z,r}$ via the following point to point mapping

$$f\left(\Phi_f(k)\right) \Longleftrightarrow g\left(\Phi_g(k)\right) \quad 1 \leq k \leq K \tag{30}$$

We define $D$ as the cost of matching one data set to the other. $D$ can be given

the sum of distance of the corresponding points of $f_{x,y,z,r}$ and $g_{x,y,z,r}$ as defined in equation (28). $D$ gives the total distance between the two sequences as follows

$$D\left(f_{x,y,z,r}, g_{x,y,z,r}\right) = \sum_{k=1}^{K} d\left(f_{x,y,z,r}\left(\Phi_f(k)\right), g_{x,y,z,r}\left(\Phi_g(k)\right)\right) \tag{31}$$

The optimum alignment minimizes the total cost in equation (31)

$$\begin{array}{rcl} D\left(f_{x,y,z,r}, g_{x,y,z,r}\right) & = & \min_{\Phi} D_{\Phi}\left(f_{x,y,z,r}, g_{x,y,z,r}\right) \\ & = & \min_{\Phi_f,\Phi_g} \sum_{k=1}^{K} d\left(f_{x,y,z,r}\left(\Phi_f(k)\right), g_{x,y,z,r}\left(\Phi_g(k)\right)\right) \end{array} \tag{32}$$

The minimizer of this function provides us with the optimum warping function. However, we want the optimum warping function $\Phi$ to satisfy some additional criteria. First of all, the warping function $\Phi$ should match the end points of the sequences $f_{x,y,z,r}$ and $g_{x,y,z,r}$

$$\begin{array}{lll} \textit{beginning point} & \Phi_f(1) = 1 & \Phi_f(1) = 1 \\ \textit{end point} & \Phi_f(K) = M & \Phi_f(K) = N \end{array} \tag{33}$$

As this equation suggest the lengths of the the matched sequence do not have be same. In addition to the end points, the warping function $\Phi$ has to satisfy a monotonicity condition as we want conserve the order of sequences.

$$\begin{array}{l} \Phi_f(k+1) > \Phi_f(k) \\ \Phi_g(k+1) > \Phi_g(k) \end{array} \tag{34}$$

Furthermore, in order minimize the potential loss of information in the two sequences we apply a continuity constraint on the warping function $\Phi$

$$\begin{array}{l} \Phi_f(k+1) - \Phi_f(k) \leq 2 \\ \Phi_g(k+1) - \Phi_g(k) \leq 2 \end{array} \tag{35}$$

The minimization problem in equation (32) can be solved using dynamic programming [16]. The solution makes use of the fact that each portion of the optimum warping function $\Phi$ has to be also optimum. As we are working in the discrete domain this idea corresponds to a recursive structure which can be given as

$$\begin{aligned} D_{opt}(1,1) &= d(1,1) \quad initialization \\ D_{opt}(\Phi_f(k),\Phi_g(k)) &= \min \left\{ \begin{array}{l} D_{opt}(\Phi_f(k)-2,\Phi_g(k)-1)+3d\left(\Phi_f(k),\Phi_g(k)\right) \\ D_{opt}(\Phi_f(k)-1,\Phi_g(k)-1)+2d\left(\Phi_f(k),\Phi_g(k)\right) \\ D_{opt}(\Phi_f(k)-1,\Phi_g(k)-2)+3d\left(\Phi_f(k),\Phi_g(k)\right) \end{array} \right\} \end{aligned} \tag{36}$$

where the initialization step matches the first index of each sequences, hence satisfying the end point conditions. From this recursion we can note that $D_{opt}$ satisfies both the monotonicity and continuity conditions in equations (33,35). The iterations end when

$$D_{opt}(\Phi_f(K),\Phi_g(K)) = D_{opt}(M,N) \tag{37}$$

The recursive structure in equation (36)can be turned into an iterative structure by using dynamic programming [16]. The order of the complexity of the algorithm is given by $O(M,N)$ where $M$, and $N$ are the lengths of the supine and prone data sets respectively.

# 4 Results

We evaluated our registration algorithm on a database of ten CT scans. The CT images consists of [512x512x258] voxels sampled at an interval of [0.78125,0.78125,1.5]mm's in $x$, $y$ and $z$ directions, respectively. For each of the ten data sets, a radiologist marked five points by simultaneously viewing the supine and prone views. The

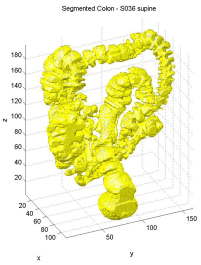
marked points might correspond to anatomical points or recognized polyp structures. We took the points marked by the radiologist as the gold standard and evaluated our algorithm based on a performance of matching those points.

We are also provided with a centerline drawn by the radiologist. We first segmented the colon data and then computed its centerline using the Hamilton-Jacobi skeletons. As the points marked by the radiologist might not lie on the medial axis we found, we marked the points on the skeleton closest to the marked points as marked points. Then we measured the distance along path between corresponding points in supine and prone data sets before and after registration for all ten CT scans. We present the reduction in the distance along path before and after the registration as our performance measure.

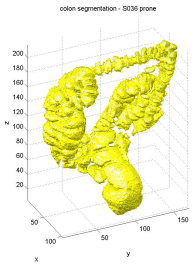
In this section we will provide the results of our registration algorithm. We will first present the results of segmentation followed by the results of applying Hamilton-Jacobi skeletons on the colon data. Then we will state the results of registering the shock graph of corresponding supine and prone data sets using dynamic time warping algorithm

## 4.1 Segmentation of the Colon Data

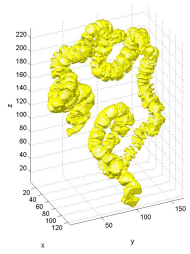
Figure 5 shows the results of segmentation for three patients. As it can be seen from the figure, seed based region growing algorithm give considerably good results. However, it can be observed that the segmented colon consists of several disconnected regions. This is due to collapsing of the colon at those regions. Although the colon is insufflated with air during the CT scan at some locations the colon might collapse. The region growing algorithm cannot handle these collapses as it only segments regions filled with air. Better results can be obtained by using active contour models for the segmentation as they are capable of segmenting the colon wall by making use of curve evolution techniques.



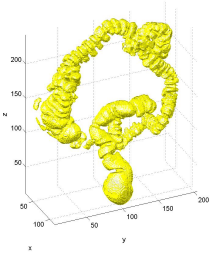

(a) Supine data (data:S036)



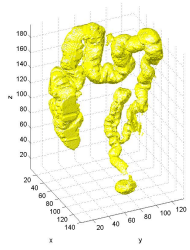

(b) Prone data (data:S036)



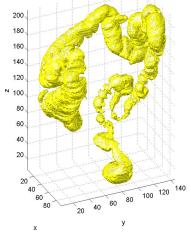
(c) Supine data (data:Y031)



(d) Prone data (data:Y031)



(e) Supine data (data:Y057)

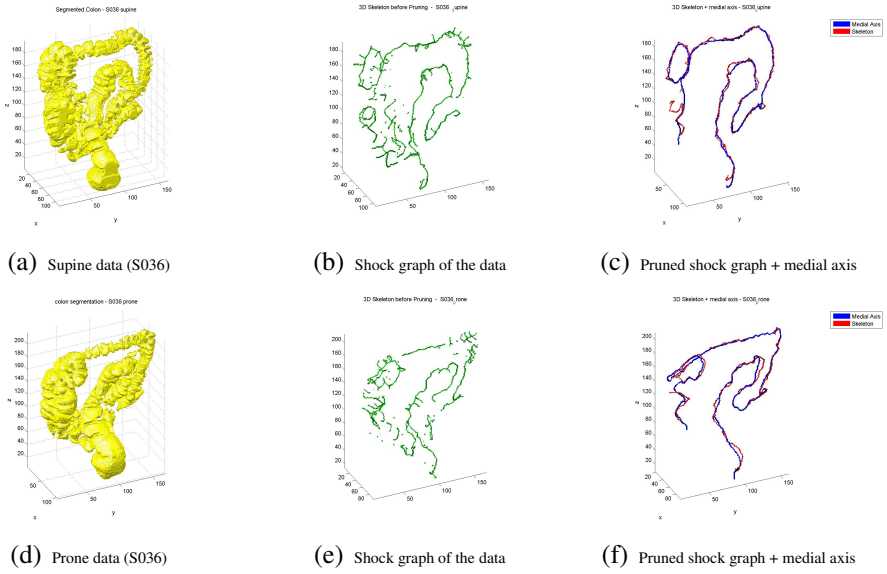


(f) Prone data (data:Y057)

Figure 5: The segmentation results for three different patients. The segmentation is done using seed based region growing algorithm.

## 4.2 Hamilton-Jacobi Skeletons

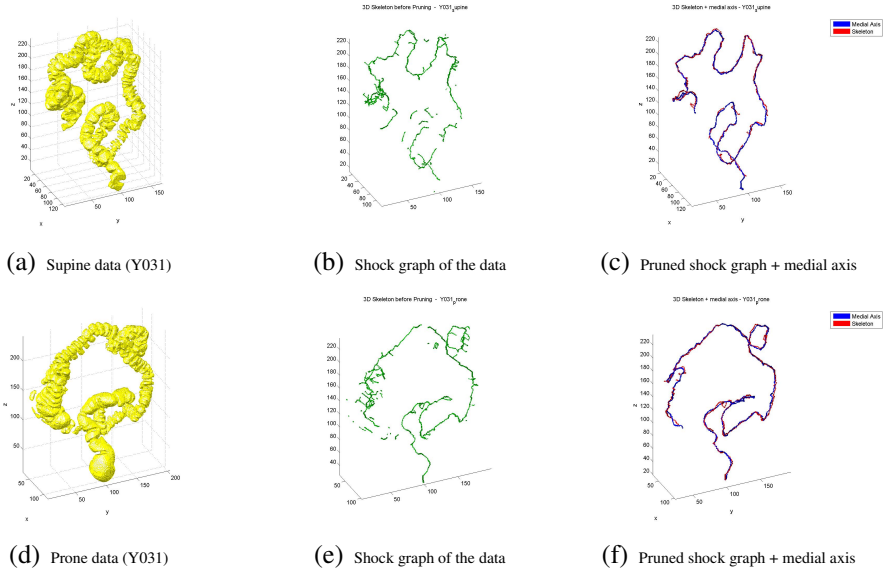
Figure 6 7 and 8 show three results for the Hamilton-Jacobi skeletons of the segmented colon data. As the figures also indiccate the applied algorithm successfully located skeletal points inside the colon. The centerline through the colon can be obtained by pruning the skeleton of the colon. Figures 6(c), 6(f), 7(c), 7(f), 10(c) and 6(c) indicate that the centerlines obtained by pruning the skeleton of the colon give very close results to the medial axis drawn by the radiologist. Therefore, we claim that extracting centerline of the colon using Hamilton-Jacobi skeletons is a reliable method if a good segmentation of the colon is provided. The bottle neck of our algorithm is that the extracted shock graph heavily depends on the segmentation. As we used a simple region growing algorithm to segment the colon data, the colon consists of several regions, which causes the skeleton to be a collection of connected sets. A single connected set through the colon can be obtained if a good segmentation of the colon is provided. We segmented the air inside colon wall; however, if methods to segment the colon wall can be used, our algorithm would perform much better.




(a) Supine data (S036) (b) Shock graph of the data (c) Pruned shock graph + medial axis

(d) Prone data (S036) (e) Shock graph of the data (f) Pruned shock graph + medial axis

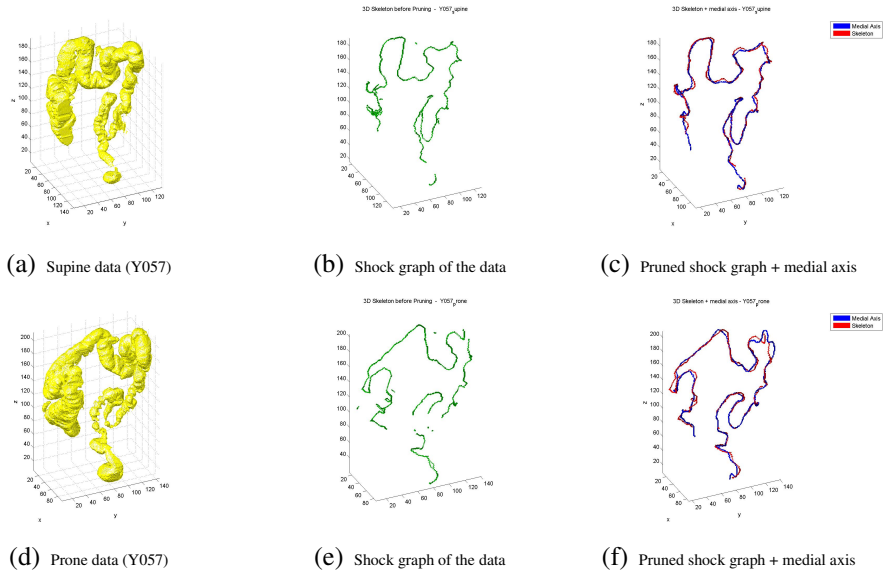
Figure 6: Shock graph of the colon data of the first patient (S036). a) Segmented colon (supine data) b) Hamilton-Jacobi skeleton before pruning c) Pruned skeleton(red) is shown together with the radial axis drawn by the radiologist (blue). d)-f) Same as a)-c) but for the prone data.




(a) Supine data (Y031) (b) Shock graph of the data (c) Pruned shock graph + medial axis

(d) Prone data (Y031) (e) Shock graph of the data (f) Pruned shock graph + medial axis

Figure 7: Shock graph of the colon data of the second patient (Y031). a) Segmented colon (supine data) b) Hamilton-Jacobi skeleton before pruning c) Pruned skeleton(red) is shown together with the radial axis drawn by the radiologist (blue). d)-f) Same as a)-c) but for the prone data.




(a) Supine data (Y057) (b) Shock graph of the data (c) Pruned shock graph + medial axis

(d) Prone data (Y057) (e) Shock graph of the data (f) Pruned shock graph + medial axis

Figure 8: Shock graph of the colon data of the third patient (Y057). a) Segmented colon (supine data) b) Hamilton-Jacobi skeleton before pruning c) Pruned skeleton(red) is shown together with the radial axis drawn by the radiologist (blue). d)-f) Same as a)-c) but for the prone data.

## 4.3 Registration

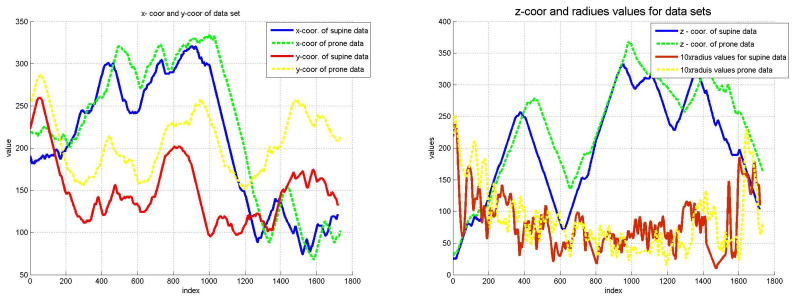


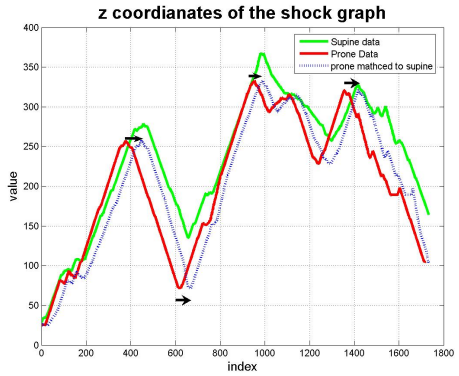

Figure 9: Sequences in supine and prone data sets are shown together (data: S036). The first figure shows x and y coordinates of the shock graph. The second figure whos the z coordinates and the corresponding radius values. Supine data is shown with solid lines and prone data is shown with dashed lines. Dynamic time warping finds an optimum warping function such that the shape of the prone data resembles the prone data while fulfilling proper constrains.

After obtaining the shock graph of the colon (see figure 9) we use the shock graphs corresponding to supine and prone data sets as input to the dynamic time warping algorithm. Figure 10 shows that dynamic time warping algorithm finds a warping function such that the Euclidean distance between the supine and prone data sets are minimized. Note that in figure 10 the peaks of the prone data are matched to the peaks of the supine data.

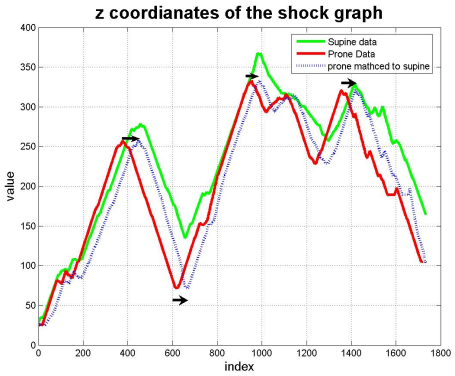
Figures 11 and 12 summaries our results for matching the ten data sets we used. Figure 11 clearly shows that for each patient the sum of distances decreased after applying our algorithm. Figure 12 shows the overall performance of our algorithm. The regression line on the plot suggest that our algorithm has achieved its goal of decreasing the distance between marked points. We can roughly state that our algorithm decreases the distance between the marked points by a constant of 0.24 while adding a constant of 5.0. However, we should make some remarks for our data sets. Although the marked points are provided by a radiologist and we take them as the gold standard, the marked points are determined according to a medial axis drawn by the radiologist. We map the marked points to its closest points in the shock graph. Therefore, the results might not be considered reliable

for our purposes. However, the result clearly indicate that our algorithm significantly decreased the distance between marked points after applying our matching algorithm.

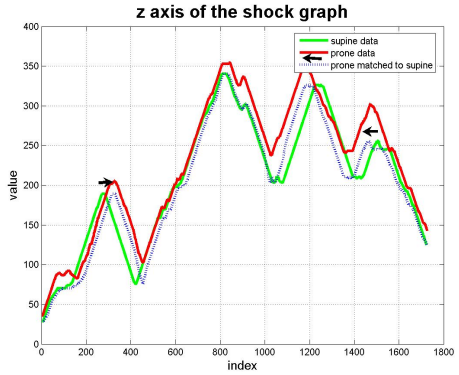
After applying our algorithm the mean distance between the marked points decreased from 230mm to 73mm which accounts for a %68 decrease on average. In the mean time, the standard deviation of the distance between marked points decreased from 88.5mm to 28.1mm which makes a %68 decrease on average.




(a) z-coordinates of shock graph (S036)




(b) z-coordinates of shock graph (Y031)




(c) z-coordinates of shock graph (Y057)

Figure 10: Supine to prone matching using dynamic time warping. Prone data is matched to the supine data. Note the nonlinear stretching and shrinking operations. Supine data is shown in green, prone data in red and the matched prone data is shown in blue dashed lines.

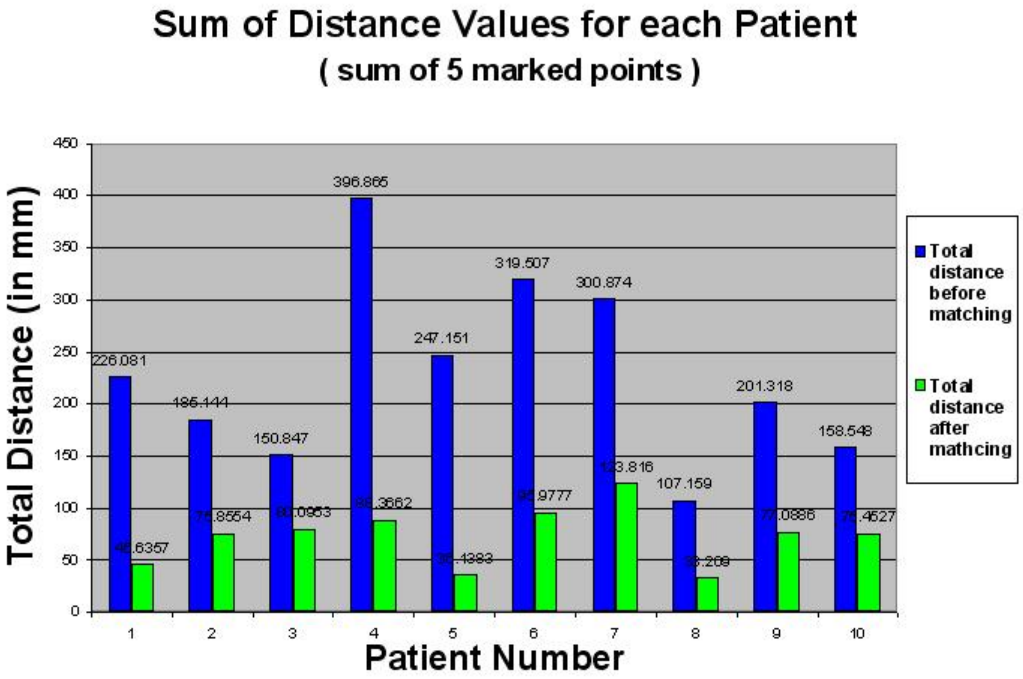



Figure 11: The sum of total distance along path for all five marked points for each patient. Note the decrease in matching error after applying dynamic time warping algorithm.

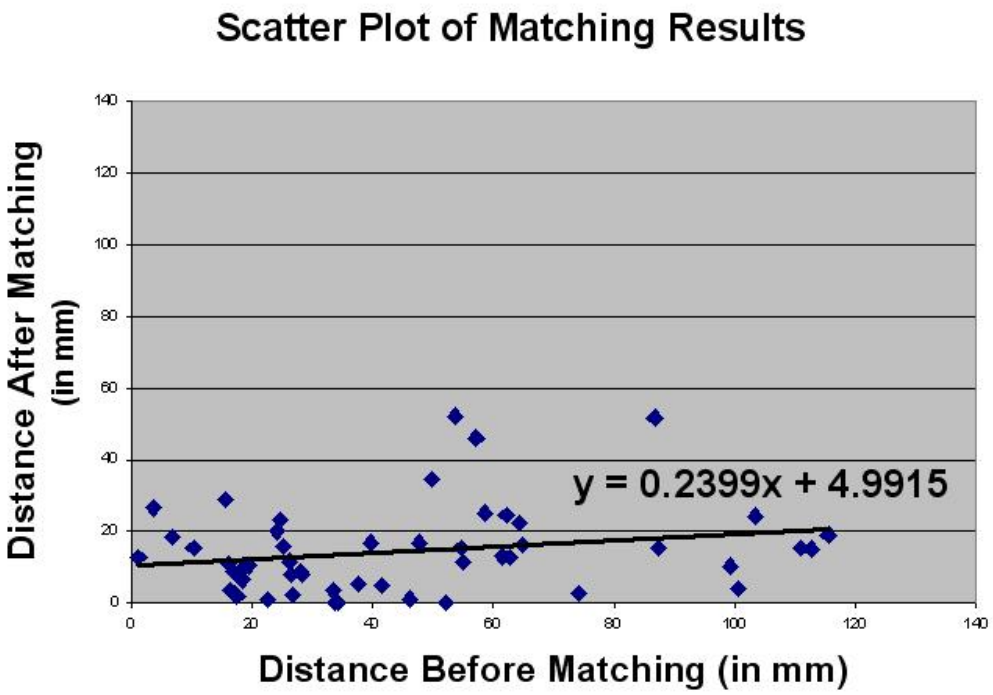



Figure 12: The scatter plot of marked points before and after matching. x-axis shows the distance between marked points before matching and y-axis shows the distance between marked points after applying dynamic time warping algorithm to the shock graph of the colon. The linear regression line suggests that our algorithm decreases the matching error with a linear constant of 0.24 while adding a constant of 5mm to each marked point.

# 5 Conclusion

We proposed an automatic registration algorithm based on computing the shock graph of the colon and applying dynamic time warping algorithm to match the shock graphs corresponding to supine and prone data. We begin by segmenting the colon from the CT scan by using a seed based region growing algorithm. Then we compute the shock graph of the colon by Siddiqi [15] et al.'s Hamilton-Jacobi skeletons. Thereby, we obtain the centerline through the colon and radius of maximal spheres at each centerline point. We use the computed shock graph of the supine and prone data sets as an input to the dynamic time warping algorithm. Using dynamic time warping algorithm we obtain a warping function which minimizes the Euclidean distance between the two data sets (see 32). BY using the computed warping function we register the prone data to the supine data.

We provide a performance measure for our algorithm where we register 50 points marked by a radiologist in a total of 10 CT scan sets. Our algorithm decreased the mean of the distance between marked points from 230mm before mathing to 73mm after matching, a %68 percent decrease on average. Also the standard deviation decreased from 88.5mm before matching to 28.1mm after matching. The results we achieved with our algorithm are comparable to the results obtained by Acar [17] et al..

Our algorithm can be further improved if segmentation methods based on active contours are used, as in our algorithm the computation of a reliable shock graph of the colon is heavily dependent on segmentation results.

# References


[1] J van der Peijl J Florie R E van Gelder F Gerritsen J Stoker A H de Vries, R Truyen. Feasibility of automated matching of supine and prone ct-colonography examinations. *British Journal of Radiology*, 2006.

[2] F Preteux B Longuet A Manzanera, TM Bernard. Medial faces from a concise 3d thinning algorithm. *International Conference on Computer Vision*, page 337343, 1999. Kerkyra, Greece.

[3] DS Paik P Li J Yee CF Bealieu RB Jeffrey. B Acar, S Napel. Registration of supine and prone ct colonography data: Method and evaluation. *Radiological Society of North America 87th Scientific Sessions,*, 2001.

[4] H Blum. Biological shape and visual science. *Journal of Theoretical Biology*, 38:205287, 1973.

[5] GS di Baja C Arcelli. A width-independent fast thinning algorithm. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 7(4):463474, 1985.

[6] T Chan and L Vese. Active contours without edges. *IEEE Trans. Image Processing*, 10(1):266–277, 2001.

[7] R Kikinis W Grimson D Nain, S Haker. An interactive virtual endoscopy tool. *Satellite Workshop at the Fourth International Conference on Medical Image Computing and Computer-Assisted Intervention (MICCAI'2001)*, 2001.

[8] DJ Robinson DJ Sheehy, CG Armstrong. Shape description by medial surface construction. *IEEE Transactions on Visualization and Computer Graphics*, 2(1):6272, 1996.

[9] N Ayache G Malandain, G Bertrand. Topological segmentation of discrete surfaces. *International Journal of Computer Vision*, 10(2):183197, 1993.

[10] N Gagvani S Dickinson H Sundar, D Silver. Skeleton based shape matching and retrieval. *International Conference on Shape Modeling and Applications*, page 290, 2003.

[11] Aang K and Gasser T. Alignment of curves by dynamic time warping. *The Annals of Statistics*, 25(3):1251–1276, 1997.

[12] BBCW Shu K Siddiqi. Geometric shock-capturing eno schemes for subpixel interpolation, computation and curve evolution. *Graphical Models And Image Processing*, 59(5):278–301, 1997.

[13] SJ Dickinson K Siddiqi, A Shokoufandeh and SW Zucker. Shock graphs and shape matching. *International Journal of Computer Vision*, 35(1):1332, 1999.

[14] SW Zucker K Siddiqi, A Tannenbaum. Hamiltonian approach to the eikonal equation. *Lecture Notes In Computer Science*, (1654):1–13, 1999.

[15] Allen Tannenbaum Steven W. Zucker Kaleem Siddiqi, Sylvain Bouix. Hamilton-jacobi skeletons. *International Journal of Computer Vision*, 48(3):215 – 231, July 2002.

[16] Bing-Hwang Juang Lawrence Rabiner. *Fundamentals of Speech Recognition*. Prentice Hall, 1993.

[17] S. Acar B. Paik D. S. Jeffrey R. B. Beaulieu C. F. Li, P. Napel. Registration of central paths and colonic polyps between supine and prone scans in computed tomography colonography: Pilot study. *Medical Physics*, 31(10):2912–2923, October 2004.

[18] D Terzopoulos M Kass, A Witkin. Snakes: Active contour models. *International Journal of Computer Vision*, 1:321–331, 1988.

[19] S. Grimson W. E. L. Cosman E. Wells W. W. Ji H. Kikinis R. Westin C.-F. Nain, D. Haker. Intra-patient prone to supine colon registration for synchronized virtual colonoscopy. *Lecture Notes In Computer Science*, 2489:573–580, 2002.

[20] S Osher and JA Sethian. Fronts propagating with curvature dependent speed: Algorithms based on hamilton-jacobi formulation. *Journal of Computational Physics*, 79:12–49, 1988.

[21] C Pudney. Distance-ordered homotopic thinning: A skeletonization algorithm for 3d digital images. *Computer Vision and Image Understanding*, 72(3):404413, 1998.

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

[23] A Tannenbaum S Bouix, K Siddiqi. Flux driven fly throughs. *Ieee Computer Society Conference On Computer Vision And Pattern Recognition*, I-449-I-454 2003.

[24] A Tannenbaum S Bouix, K Siddiqi. Flux driven automatic centerline extraction. *Medical Image Analysis*, 9:209–221, 2005.

[25] K Siddiqi S Bouix. Divergence-based medial surfaces. *Lecture Notes In Computer Science*, 1842:603–620, 2000.

[26] P Olver A Tannenbaum A Yezzi S Kichenassamy, A Kumar. Conformal curvatures flows: From phase transitions to active vision. *Arch. Rational Mech. Anal.*, 134(3):275–301, 1996.

[27] R Fedkiw S Osher. *Level Set Methods and Dynamic Implicit Surfaces*. Springer Verlag, 2003.

[28] Jayant Shah. Gray skeletons and segmentation of shapes. *Computer Vision and Image Understanding*, 99:96109, 2005.

[29] Jayant Shah. Skeletons of 3d shapes. *Lecture Notes in Computer Science*, 3459:339 350, Jan 2005.

[30] PR Cohen T Oates; L Firoiu. Clustering time series with hidden markov models and dynamic time warping. *Proceedings of the IJCAI-99 Workshop*

*on Neural, Symbolic and Reinforcement Learning Methods for Sequence Learning*, pages 17–21, 1999.

[31] BB Kimia TB Sebastian. Curves vs skeletons in object recognition. *Proc. International Conference on Image Processing*, 3:22–25, 2005.

[32] Benjamin Kimia Thomas Sebastian, Philip Klein. Shock-based indexing into large shape databases. *Lecture Notes in Computer Science*, 2352:731, Jan 2002.

[33] T Coll F Dibos V Caselles, F Catte. A geometric model for active contours. *Numer. Math.*, 1(66):1–31, 1993.

[34] D. J. Vining. Virtual endoscopy: Is it a reality? *Radiology*, 200:30–31, 1996.