Department of Computer Science

The Use of Fast Marching Methods in Medical Image Segmentation

Varduhi Yeghiazaryan and Irina Voiculescu

CS-RR-15-07

Department of Computer Science, University of Oxford
Wolfson Building, Parks Road, Oxford, OX1 3QD

The Use of Fast Marching Methods in Medical Image Segmentation

Varduhi Yeghiazaryan and Irina Voiculescu

October 5, 2015

Abstract

Despite several decades of research into segmentation techniques, unsupervised medical image segmentation is barely usable in a clinical context, and still at vast user time expense. The Fast Marching method is an established segmentation technique for generic spaces, which typically requires manual initialisation. This paper illustrates unsupervised organ segmentation through the use of a novel automated labelling algorithm followed by a hypersurface front propagation method — Fast Marching. The labelling stage relies on a pre-computed image partition forest obtained directly from CT scan data.

We perform a systematic analysis of the effects of the Fast Marching method parameters, and compare the performance of the algorithm in different settings for a specific task. We also introduce novel approaches to the choice of some parameters of the Fast Marching relying on the results of hierarchical image segmentation algorithms.

We have implemented all procedures to operate directly on 3D volumes, rather than slice–by–slice, because our algorithms are dimensionality–independent. The results picture segmentations which identify abdominal organs (such as the liver and kidneys), but can easily be extrapolated to other body parts.

Quantitative analysis of our unsupervised segmentation compared against hand–segmented gold standards for kidney segmentation indicates an average Dice similarity coefficient of 90%. Results were obtained over volumes of CT data with 9 kidneys, computing both volume–based similarity measures (such as the Dice and Jaccard coefficients, true positive volume fraction) and size–based measures (such as the relative volume difference). Our analysis considers both healthy and diseased kidneys, although extreme pathological cases were excluded from the overall count. Such cases are difficult to segment both manually and automatically due to the large amplitude of Hounsfield unit distribution in the scan, and the wide spread of the tumorous tissue inside the abdomen. In the case of kidneys that have maintained their shape, the similarity range lies around the values obtained for inter–operator variability. Whilst the procedure is fully unsupervised, our tools also provide a light level of manual editing.

1 Introduction

In a clinical or clinical research context, the shapes of anatomical structures of interest are typically found from single slices of CT or MRI imaging data using commercial software (such as Mimics [1] or Simpleware [2]). These shapes are segmented from individual images manually: this involves setting a threshold then going through slice by slice and tidying up regions which have not been detected correctly. Processing a set of scan slices takes many user hours. The uptake of computer–based segmentation methods in research fields which process large volumes of visual data (such as scans) depends crucially on shortening this user time.

In this paper we refer to *segmentation* as the process of identifying and labelling a 3D region of semantic importance (such as an organ, blood vessel, tumour or bone). In doing so, we pre-process the data in order to group voxels with similar characteristics into regions — a process known in Computer Vision as (2D) *image segmentation.* In order to avoid ambiguity, we will refer to the vision segmentation technique as *partitioning* the image, and to the medical segmentation technique as *labelling* the features of interest.

Level set and Fast Marching methods have been previously used for medical image analysis. In [21, 19, 20, 22, 17] Malladi et al. use level set methods to identify features in 2D and 3D images. Fast Marching for feature labelling in medical images was investigated in a number of papers [16, 15, 18, 4, 5].

The behaviour of the Fast Marching method for interface motion depends on several parameters. These include the growth rate of the front interface, and the spacing of the lattice on which we discretise the model. Other factors that have impact on the success of the method are the choice of the seed points, i.e. the initial interface from which the motion begins, and the stopping criteria for this motion.

There is no detailed analysis of parameter choices for the method and their comparison for a specific task. Concentrating on other aspects of the algorithms, the literature just mentions empirical parameter choices made in their work. Our results fill this gap in the literature. We introduce novel approaches to the selection of some of the parameters. We also perform systematic experiments intended to identify the influence of various parameter choices on the performance of the Fast Marching method.

Current practice for medical imaging [20, 16, 15, 18, 29] leaves the initialisation of level sets or Fast Marching to the user. Even commercial software for processing scan data expects the user to hand–seed any region growing methods, and this on a slice–by–slice basis. In other cases, heavily data–specific methods are used for automated seed (initial region) choice [4, 5]. We present a new approach for fully automatic labelling and reconstruction of features, such as organs, which combines Fast Marching with a hierarchical image partitioning method.

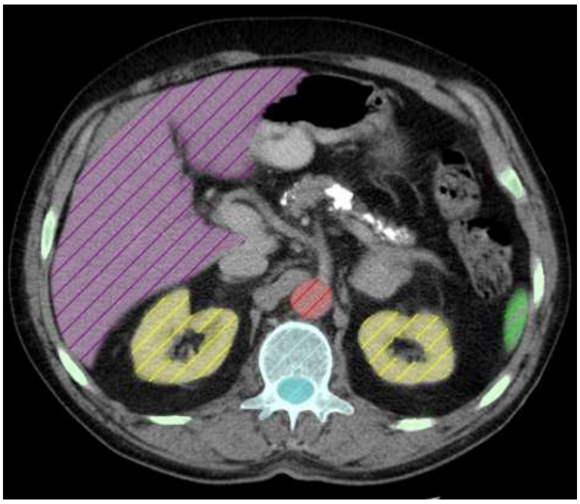


Figure 1: A CT axial slice with manually identified features. Aorta, kidneys, liver, ribs, spine, spinal cord and spleen are visible in their respective colour.

Following our previous work on decision-support systems for renal segmentation [9, 11], we illustrate unsupervised organ segmentation through the use of a novel automated labelling algorithm followed by a hypersurface front propagation method. The labelling stage relies on a pre-computed image partition forest obtained directly from CT scan data.

2 Context

We illustrate our algorithms with abdominal CT scan image volumes in DICOM format, typically comprising up to 50 axial slices, 512 × 512 voxels each. A scan slice is presented in Figure 1 with the features of interest labelled manually.

Fast Marching–based labelling is easily generalised to any dimensionality. We choose always to run it on 3D data volumes, preserving the original resolution.

We sometimes refer to the greyscale value of each voxel, appearing to be oblivious to its provenance. In practice, the CT scanner associates to each voxel a scalar Hounsfield Unit (HU) proportional to the radiodensity of the tissue at that point. The HU value range (several thousand units) is much larger than the available range of greyscale values (256). Instead of applying a lossy linear mapping from one to the other, we make use of a careful windowing mechanism [31], which preserves granularity detail in a given subrange of interest, and only shrinks the less relevant ranges of HUs in the image. In order to obtain greyscale data, we pre-process the raw data with windowing and then edge-preserving smoothing with several iterations of anisotropic diffusion filtering [23].

3 Image Partitioning

The pre-processing is followed by a partitioning step, where the 3D data volume is organised into a tree–based representation of adjacency graphs. The nodes of the trees are regions of voxels, and the parental relationship is that of inclusion. Essentially, the data is *partitioned* into a hierarchy of inter–connected regions and subregions.

We call this data structure an Image Partition Forest (IPF). We have described elsewhere [10, 12] how we use a watershed and waterfall approach [3] to construct the IPF in a bottom-up manner. A first layer is constructed by running the watershed transform [3, 12] on the original 3D image. This process views the gradient magnitude of the input image as a landscape and locates water catchment basins. Then four other layers of coarser detail are made by running the waterfall transform [3, 12] on the previous layer, which transforms the input partition into a 'stepped' landscape and performs watershed on it.

Figure 2 illustrates a 2D projection through a 3D IPF, and its different levels of coarseness. The bottom–most level is made up of the raw image voxels; it is followed by a layer of over-segmented regions (not shown), followed by increasingly coarse regions that belong together and gradually group themselves around organs and other features.

The most crucial aspect of automating the segmentation is the way in which we use the IPF to find a rough approximation of an organ and seed from it a hypersurface front propagation method which helps localise the organ boundaries more accurately.

For instance, since the typical HU range (and hence windowed greyscale) for kidneys is known from radiology studies [31], we can choose as kidney candidates those regions in the IPF which are within the desired mean greyscale range. Furthermore, filtering regions which correspond roughly to kidneys can also be based on anatomical knowledge. The IPF candidate regions should be in a high layer of the IPF (layers 3–5); in an axial slice, the right kidney should lie 'west' of the spine and anatomically close to it; also, the right kidney should span a reasonable number of voxels (depending on the number of slices in the image).

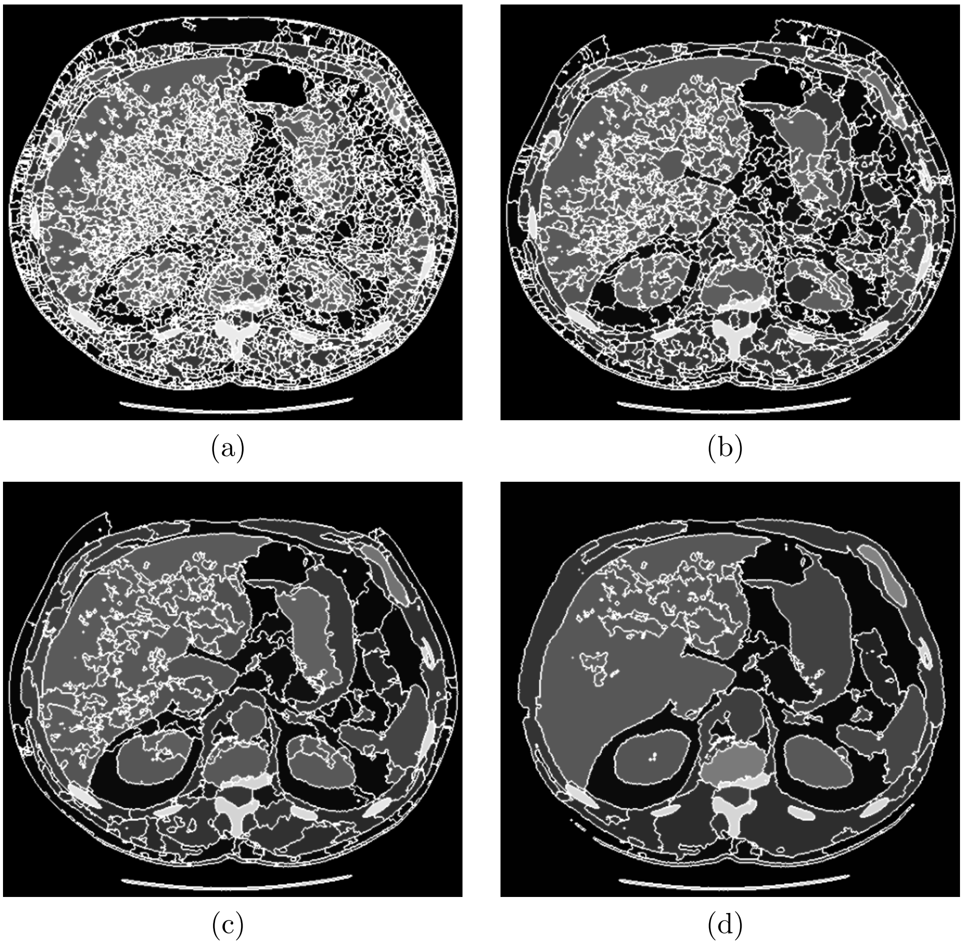


Figure 2: Four of the six layers (finest layers not shown) in a 3D IPF, representing image partitions in decreasing detail level.

4 Feature Labelling with Fast Marching

4.1 Fast Marching basics

The Fast Marching method is an efficient iterative algorithm, introduced by Sethian [25, 28], for numerical approximation of the development of fronts propagating through $\mathbb{R}^n$ space.

A propagating front is a closed hypersurface, each point of which moves in the direction of the surface normal with some speed F. Let $S(t) \subset \mathbb{R}^n, t \in [0, \infty)$ be the propagating interface in space $\mathbb{R}^n$. If the speed function $F = F(\vec{x})$ does not change sign, the equation of the evolution of the front can be represented in terms of the arrival time function $T : \mathbb{R}^n \to \mathbb{R}$, such that

$$T(\vec{x}) = t \iff \vec{x} \in S(t)$$

as the eikonal equation (see, for instance, [29]):

$$|\nabla T| = \frac{1}{F} \tag{1}$$

Fast Marching discretises the eikonal equation (1) with upwind approximation schemes, in $\mathbb{R}^3$:

$$\left[\begin{array}{r} \max\left(\frac{T_{ijk}-T_{i-1jk}}{\Delta x}, -\frac{T_{i+1jk}-T_{ijk}}{\Delta x}, 0\right)^2 \\ +\max\left(\frac{T_{ijk}-T_{ij-1k}}{\Delta y}, -\frac{T_{ij+1k}-T_{ijk}}{\Delta y}, 0\right)^2 \\ +\max\left(\frac{T_{ijk}-T_{ijk-1}}{\Delta z}, -\frac{T_{ijk+1}-T_{ijk}}{\Delta z}, 0\right)^2 \end{array}\right]^{1/2} = \frac{1}{F_{ijk}}$$

where $\Delta x, \Delta y, \Delta z$ are the spacing of the discretization lattice, T_{ijk} and F_{ijk} the values of these functions on the points of the lattice. Using the fact that the front crosses each point only once, an efficient $O(N \log N)$ algorithm, where N is the total number of points in the lattice, is deviced to determine the values T_{ijk} in the order the front propagates (details in [25, 26, 27, 28, 29, 30]). An $O(N)$ complexity implementation of Fast Marching can be found in [34, 24].

4.2 Seeds

Seeds are points which initialise the front (we denote the set of these as $S(0) \in \mathbb{R}^n$). There can be more than one seed for a given front. In current practice for medical imaging, the choice of seeds is left to the user [20, 16, 15, 18, 29]. In other cases, heavily data specific approaches are used for automated seed extraction [4, 5] in multi–stage segmentation. Yet we seed our Fast Marching automatically from a subset of the candidate regions output via the IPF. Our take to the seeding process relies on the automatic generation of the IPF structure. Whilst filtering the candidate regions relies on anatomical structure, this can be hard–coded into the unsupervised segmentation, as seen in Section 3. Only voxels which are in the expected healthy-tissue greyscale range are considered and used as seeds, as shown in Figure 4(d).

The candidate regions chosen using the IPF do not approximate the kidneys sufficiently well themselves. Local heterogeneities of high gradient magnitude value are usually missed and most regions have displaced edges. Because of this, these candidate regions may miss necessary pieces of tissue or contain chunks of external tissue. This is why the Fast Marching is employed.

The set of seed points is the zero level set of function $S(\vec{x})$ before the front expansion begins. This means that the time function $T(\vec{x})$ has to be equal to 0 at, and only at, these points:

$$T(\vec{x}) = 0 \iff \vec{x} \in S(0)$$

The requirements [29] for the set of seed points in Fast Marching Methods are:

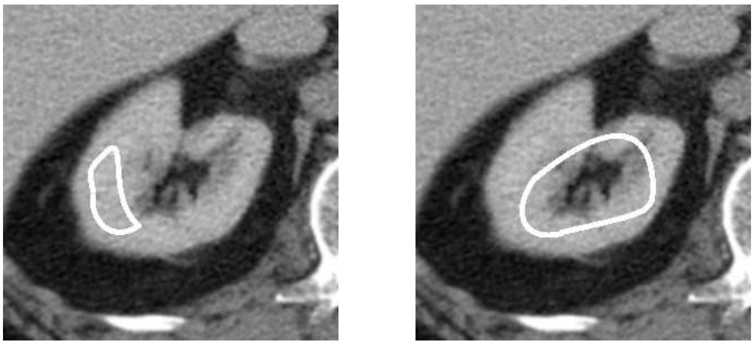
- all the seed points are within the target region to be identified, OR
- the seed points comprise a closed front which encloses the target region (or target regions) to be identified.

In the first case an expanding wave is generated beginning from the seed points to grow until the boundary of the target region. In the second case, the propagating front has a negative speed term and it develops inwards.

However, for the purpose of identifying features on medical images it is more reasonable to define seed points within target regions and propagate outwards. This is because some physiological features are not simply connected in the CT image space. This means that identifying such features with the front propagating inwards in Fast Marching would require several runs of the method. Also, some elaborate stopping criteria would be needed to correctly locate the different borders of features.

The procedure is much simpler in case of using a front propagating outwards. We use an initial curve which encloses only points which belong to the feature. This approach is illustrated in Figure 3 on the example of a kidney. When such a front expands, it approaches both the inner and outer boundaries of the kidney, and it shapes only the area of the feature (excluding the renal pelvis). In contrast, a curve, which encloses a part of (or the whole) renal pelvis, approaches only one of the inner and outer borders — the inner in case of a negative speed term, and the outer otherwise.

Even a single seed point can usually be sufficient for the method to identify the feature successfully. This is just the degenerate case of an initial



(a) A suitable initial front for a kidney. When developing outwards it approaches both the inner and outer borders of the kidney, thus shaping the organ.

(b) An unsuitable candidate for a kidney. If expanding, it only approaches the outer border. If contracting, on the contrary, the inner boundary alone is shaped.

Figure 3: Two candidates for an initial front for Fast Marching within the feature.

front like the one in Figure 3a. Thus, an initial curve can be replaced with a set of points (or a single point) within the feature.

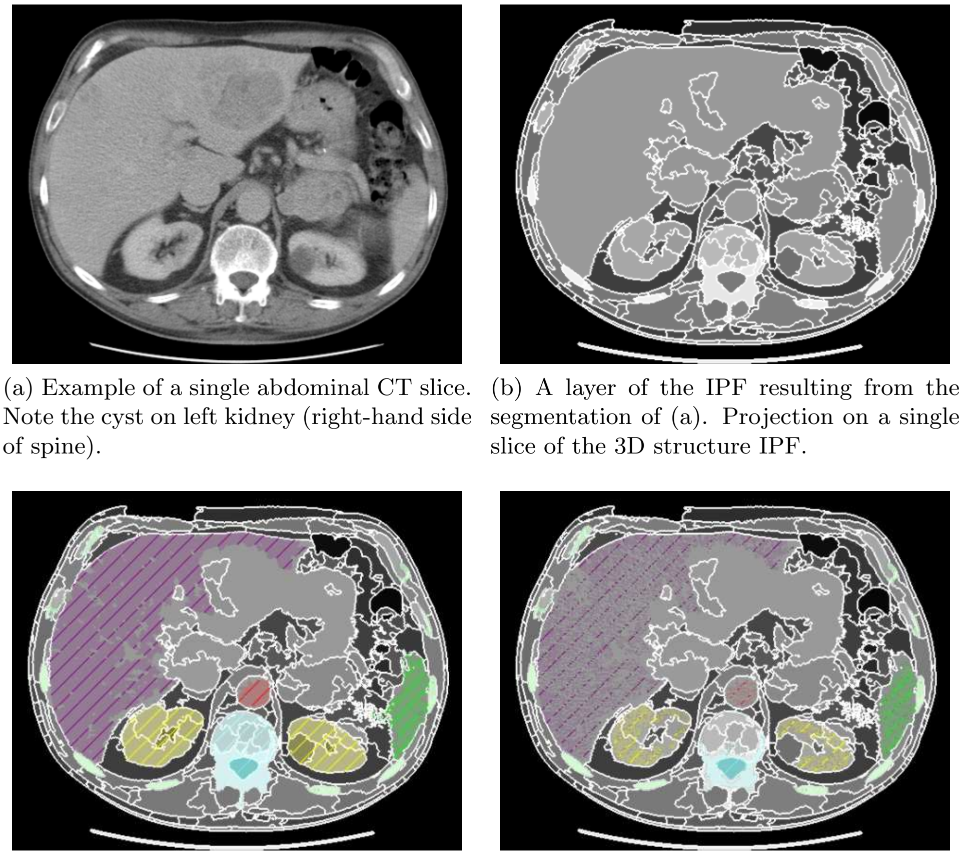
Figure 4b depicts a layer of the IPF constructed by the partitioning algorithm for the example image in Figure 4a. It can be seen that the main features are well outlined. However, further analysis reveals that the borders between the regions in some cases do not follow precisely the physical boundaries between features. Hence, they cannot be used as end-results of feature segmentation. The result of the filtering process for candidate regions, reported in Section 3, is depicted in Figure 4c and illustrates some of possible shortcomings of this intermediate result.

In order to seed the Fast Marching, within a given candidate region for a specific feature we choose only points within a fixed greyscale range inferred from the typical radiodensity HU of the feature. This is repeated for all organs of interest within the abdomen. In Figure 4d the seed point sets chosen from these regions for all the features are depicted.

4.3 Speed functions

Fast Marching for feature segmentation places important requirements on its speed term, i.e. the rate of development of the front interface. The speed function has to converge to zero rapidly in the vicinity of the feature boundary to be identified. This is devised to make sure that the expanding interface does not go beyond the border of the targeted object but rather shapes it precisely.

The features' boundaries usually correspond to noticeable changes in the image and hence are characterized by high values of the gradient magnitude



(a) Example of a single abdominal CT slice. Note the cyst on left kidney (right-hand side of spine).

(b) A layer of the IPF resulting from the segmentation of (a). Projection on a single slice of the 3D structure IPF.

(c) The regions approximating the features in the IPF. Kidney region includes the cyst. Liver is undersegmented.

(d) The seed points extracted from the regions in (c). Only points in healthy-tissue range for that organ are included.

Figure 4: Image partitioning and seed extraction from the resulting IPF

function. In contrast, the area within the organ is commonly monotonous and monochrome. This means that the same gradient magnitude function tends to have smaller values within the feature. As a consequence, the speed term has to be chosen according to these properties of the gradient image. The speed must have relatively high values at points with small gradient magnitude and vice versa.

In [15], Malladi and Sethian use the speed term

$$F(\vec{x}) = \frac{1}{1 + |\nabla G_\sigma * I(\vec{x})|}$$

for shape recovery on medical images, where $G_\sigma * I(\vec{x})$ is the original image smoothed with Gaussian filter [29]. While in [16] the same authors suggest

$$F(\vec{x}) = e^{-\alpha|\nabla G_\sigma * I(\vec{x})|}, \quad \alpha > 0,$$

as a speed term for the same task, without discussing the relative merits of these choices.

In [4] and [5] two other speed terms are suggested but again there is no comparative analysis.

We show a series of systematic experiments with the speed term expressions like in equations (2) and (3). These are carried out on raw images and on images pre-processed with windowing, smoothing.

In this paper we illustrate the effects by running the Fast Marching algorithm on a single CT slice. We always start the method at the same point on the right kidney and compare the development of the interface visually. The more precisely the right kidney is shaped the more successful we consider the speed term expression (in CT axial slices, the right kidney is on the left of the image).

$$F(\vec{x}) = \frac{1}{\left(1 + \left(\frac{|\nabla I(\vec{x})|}{C}\right)^n\right)^m}, \quad C > 0, \quad n, m \in \mathbb{N} \tag{2}$$

$$F(\vec{x}) = e^{-C|\nabla I(\vec{x})|}, \quad C > 0 \tag{3}$$

4.3.1 Base image: inversed gradient magnitude speed terms

To start with, we use the original single CT image data (without any smoothing techniques) to construct the speed term for Equation (1). The simplest expression is $F(\vec{x}) = \frac{1}{1+|\nabla I(\vec{x})|}$, where $I(\vec{x})$ is the *HU* value proportional to the radiodensity of the tissue at point $\vec{x}$ on the CT scan, and $\nabla I(\vec{x})$ is the *gradient function* of the HU image. Figure 5 depicts the results for this speed term with closed white contours corresponding to several level sets of $S(\vec{x})$ with a fixed level step. It is easy to see that, although they follow some patterns in the image, the expansion of the interface does *not* capture

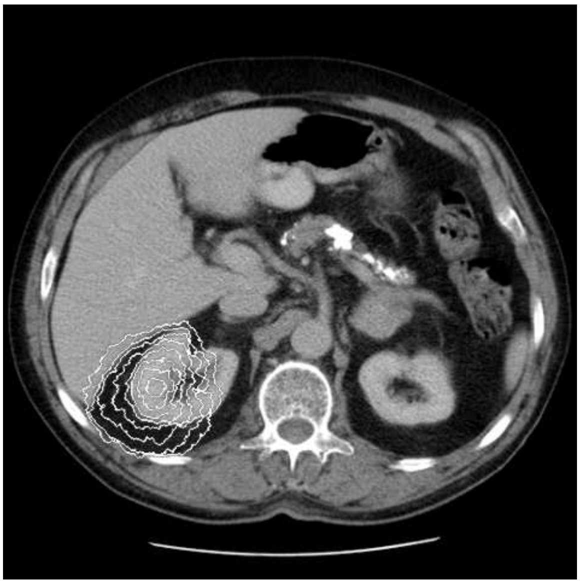
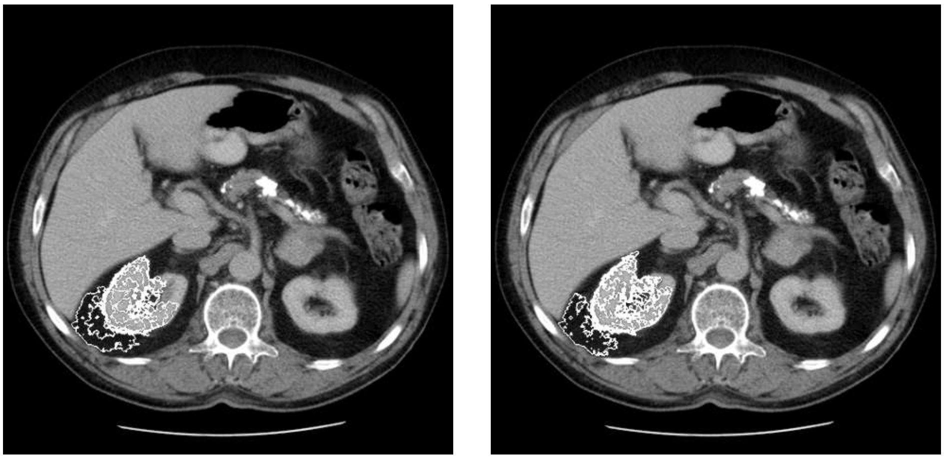


Figure 5: Fast Marching with speed term $F(\vec{x}) = \frac{1}{1+|\nabla I(\vec{x})|}$, closed white contours correspond to several level sets of $S(\vec{x})$ with a fixed level step. The impact of the image data is weak and the kidney boundary is not detected (right kidney on left of image).



(a) $F(\vec{x}) = \frac{1}{1+|\nabla I(\vec{x})|^2}$. The sensitivity to feature changes in the image is still very low and the feature boundary is crossed early.

(b) $F(\vec{x}) = \frac{1}{1+|\nabla I(\vec{x})|^3}$. The improvement upon the previous expressions is not sufficiently good.

Figure 6: Fast Marching with speed terms $F(\vec{x}) = \frac{1}{1+|\nabla I(\vec{x})|^2}$ and $F(\vec{x}) = \frac{1}{1+|\nabla I(\vec{x})|^3}$. Note that the thicker white line of the front indicates the overlap of several level sets.

the kidney. This speed term is not sufficiently sensitive to the pronounced changes in the image.

Subsequently, a faster decreasing candidate for the speed term is found to be more successful in capturing the variation of the image. We increase the impact of the image data on the development of the interface. The goal is to suppress the expansion of the interface when it reaches the border of the right kidney and, at the same time, make it faster in relatively homogeneous areas of the organ.

We then study the inverse of the squared gradient magnitude of the HU value and then the inverse of the cubed gradient magnitude for the speed term. The outcome for these expressions is shown in Figure 6 (we always add 1 to the denominator to avoid the degenerate case).

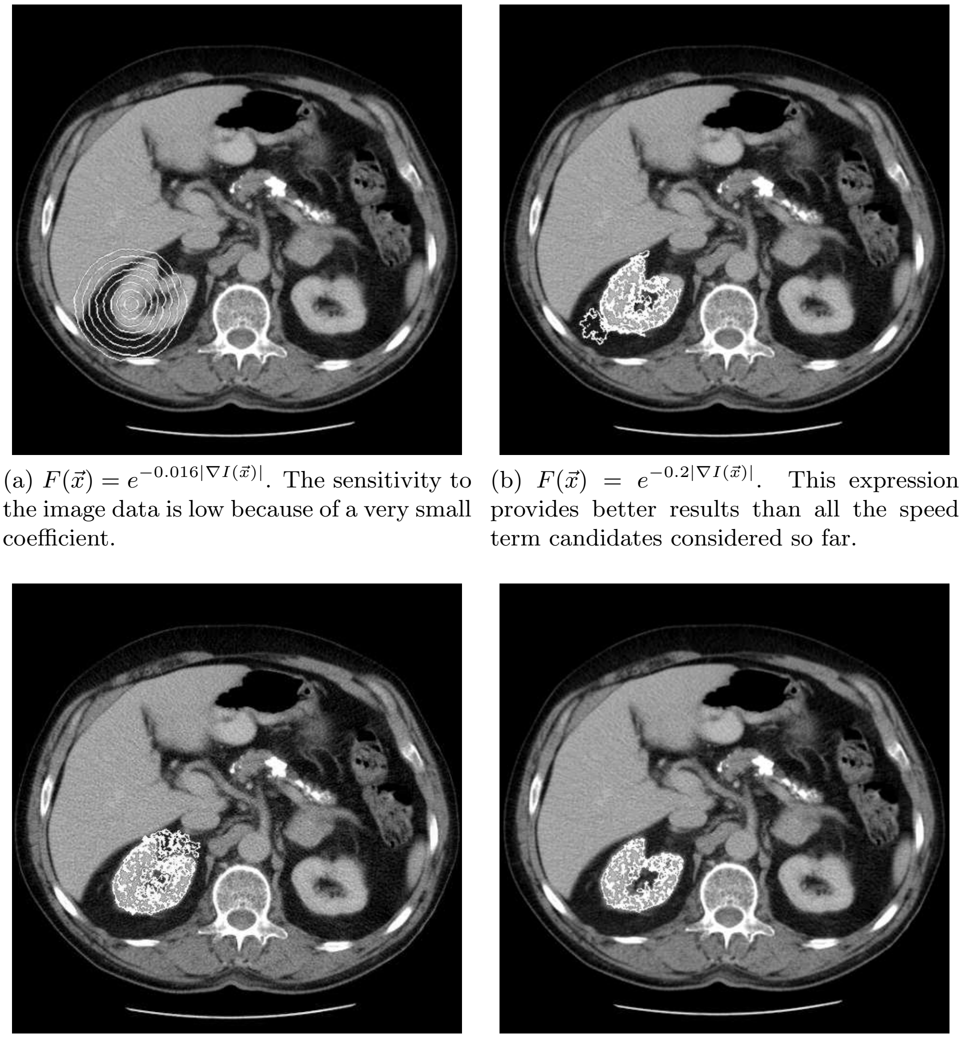
The results, illustrated in Figure 6, show that these speed expressions work better than the inverse of the gradient in Figure 5. First, the interface develops within the right kidney for some time. When it reaches the border, the high gradient magnitude of the HU value prohibits it from crossing the border for several iterations. The indicator for this is the thick white layer on some part of the border of the organ. However, at some point the propagating front breaks the chain and flows out of the region corresponding to the kidney. It is worth noticing that, by the time this happens, a part of the kidney is still not within the expanding region. The conclusion is that these expressions are better than the first one but do not yet solve the problem.

4.3.2 Base image: exponential gradient magnitude speed terms

In order to emphasize the image data even more in the development of the front, we then consider expressions exponential to the gradient magnitude of the image HU value from Equation (3). If the constant C is too small, like in Figure 7a ($C = 0.016$), then the sensitivity of the interface to image data is again very low. A larger value $C = 0.2$ provides a considerably better performance for the method (see Figure 7b), but it is still not sufficiently good for the exact task. The interface still goes beyond the area of the right kidney leaving a part of it uncovered.

Figure 7d confirms that larger values of the coefficient C prove more useful for the task. Moreover, if we stop the development of the front at some specific time with an exponential speed term and large enough coefficient, it is easy to notice that it follows the contour of the kidney. To illustrate this, we consider $F(\vec{x}) = e^{-0.5|\nabla I(\vec{x})|}$ and cease the interface expansion at some time point. The outcome is shown in Figure 8.

Despite being quite promising, this expression for the speed term in the eikonal equation (1) still has some problems. A further consideration of Figure 8 reveals that the interface has a large number of holes — some parts of the kidney are not covered by the front. The main reason for this is the



(a) $F(\vec{x}) = e^{-0.016|\nabla I(\vec{x})|}$. The sensitivity to the image data is low because of a very small coefficient.

(b) $F(\vec{x}) = e^{-0.2|\nabla I(\vec{x})|}$. This expression provides better results than all the speed term candidates considered so far.

(c) $F(\vec{x}) = e^{-0.5|\nabla I(\vec{x})|}$. The front does not cross the kidney boundary at the lower part any more.

(d) $F(\vec{x}) = e^{-0.9|\nabla I(\vec{x})|}$. The expanding region approximates the kidney but there are a lot of holes in it.

Figure 7: Fast Marching with speed term $F(\vec{x}) = e^{-C|\nabla I(\vec{x})|}$, $C > 0$.

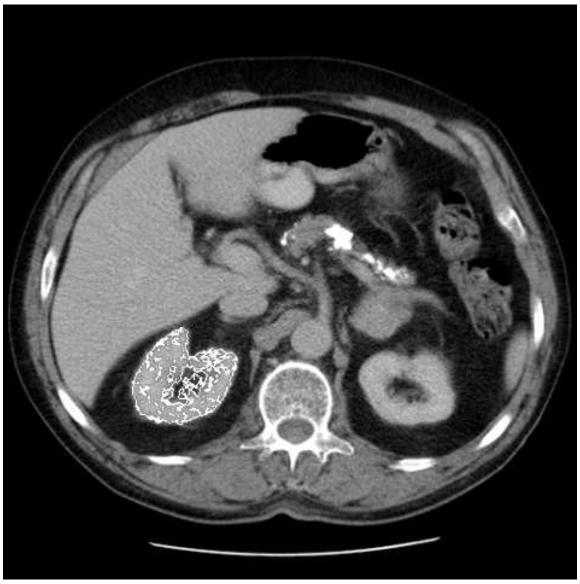


Figure 8: The interface at some time point with $F(\vec{x}) = e^{-0.5|\nabla I(\vec{x})|}$. It covers the kidney but contains too many holes.

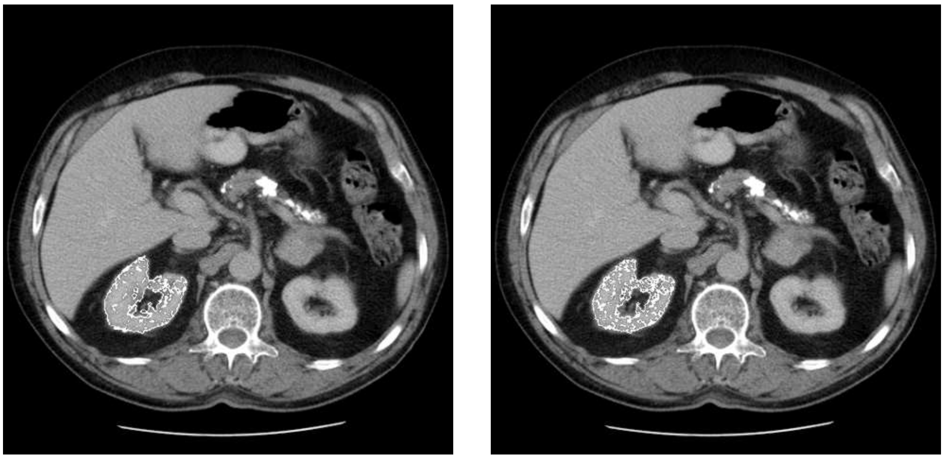
considerable level of noise in CT scans.

4.3.3 Windowed image: exponential gradient magnitude

Our first step to tackle this issue is not to use the base HU values for the image but to introduce *windowing* (for details see [31]). As a result of windowing the scale of the voxel values narrows to the greyscale value range — from 0 to 255 — and, hence, minor deviations are smoothed out.

The advantage of using windowed images over base unsmoothed images is easily seen from the comparison of the output of the method on the base image and Figure 9 where the windowed image is used. The proportion of holes in the interface in Figure 9a is much less than in Figure 8. Also, in both images in Figure 9 the front interfaces manage to cover the whole boundary of the kidney without including most of the renal pelvis. (In these images, the renal pelvis occupies the inner darker part within a kidney.) While the interface on the base image also covers a substantial part of the renal pelvis before having included the whole kidney.

However, this technique is enough to smooth out only minor noise. We introduce further smoothing techniques to tackle this issue in a more generic fashion.



(a) $F(\vec{x}) = e^{-0.5|\nabla W(\vec{x})|}$. The number of holes is considerably less compared to the raw image (without windowing).

(b) $F(\vec{x}) = e^{-|\nabla W(\vec{x})|}$. The whole kidney is covered without crossing the border inwards and including the renal pelvis.

Figure 9: The interface at some time point on the windowed image with exponential speed term. Note the coefficients 0.5 and 1.

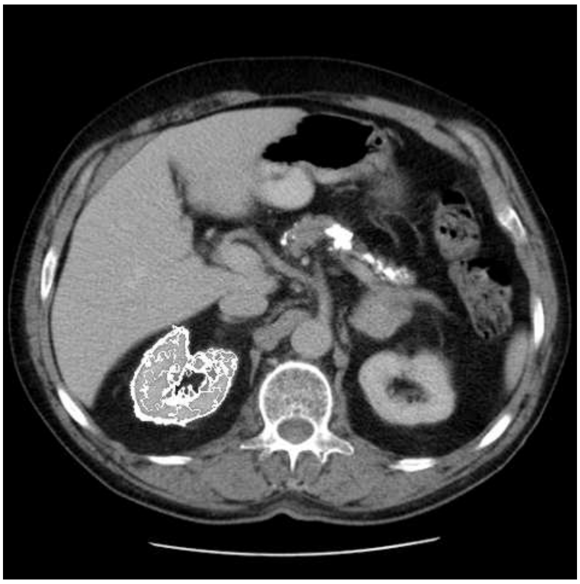


Figure 10: Fast Marching with speed term $F(\vec{x}) = \frac{1}{1+|\nabla ADW(\vec{x})|^2}$. The improvement over the unsmoothed image is huge, see Figure 6a for comparison.

4.3.4 ADF smoothed image: inversed gradient magnitude

Papers such as [16], [29] advise to smooth the input image with some edge preserving technique before using it in the Fast Marching method. In our experiments we use anisotropic diffusion filtering (first introduced by Perona and Malik [23]). We start with windowing the image to project the interesting part of the scale onto a smaller one. Then we make several iterations of anisotropic diffusion on the windowed image. And only after that do we use the image data to build the speed term expression $F(\vec{x})$ for the eikonal equation (1).

Interestingly, the application of anisotropic diffusion filtering (ADF) largely improves the performance of the Fast Marching method even for the simpler speed term expressions. For instance, Figure 10 depicts the development of the interface with speed term

$$F(\vec{x}) = \frac{1}{1 + |\nabla ADW(\vec{x})|^2}.$$

We denote the image data resulting from windowing and anisotropic diffusion by $ADW(\vec{x})$. The huge improvement in the performance of the method caused by the introduction of the smoothing techniques becomes evident if we compare this result with Figure 6a.

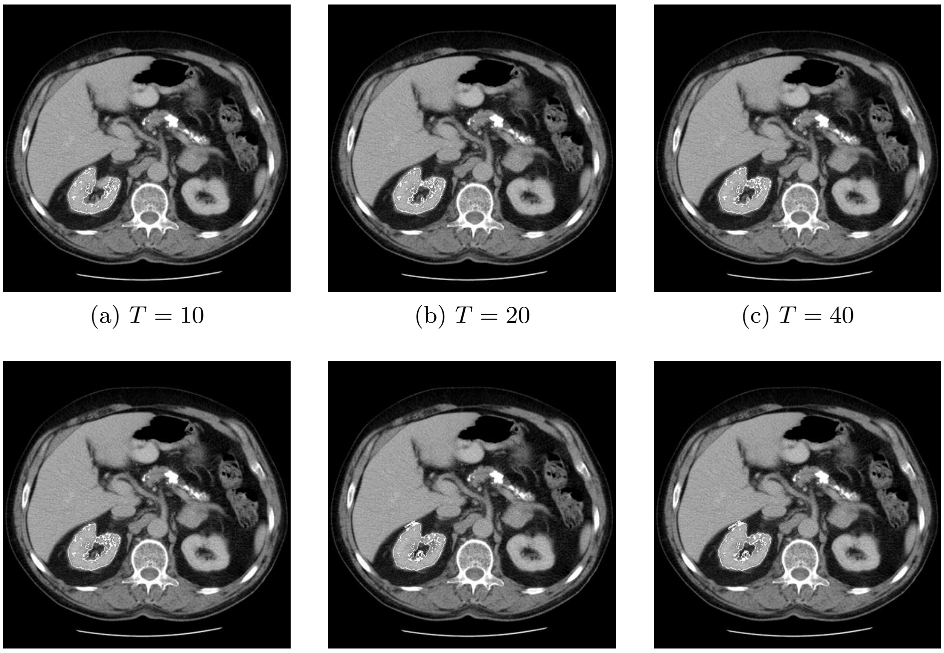
4.3.5 ADF smoothed image: exponential gradient magnitude

Now let us analyse the growth of the interface on an ADF smoothed image when we use the speed term that worked best for the unsmoothed data — exponential expression. Two different remarks about the experiment results in Figure 11: the interface undergoes extremely slow expansion from one image to another; also, most holes on the interface close as it develops.

The use of the ADF smoothing has a tremendous impact on the image data. Within the area corresponding to the organ it smooths out the differences in the values of neighbouring voxels making that area relatively uniform. Meanwhile, due to its edge-preserving quality, ADF does not have detrimental influence on the borders on the organ. As a result, the gradient magnitude values are low within the organ and much higher closer to the borders.

Because of the above, the expansion of the wave is very fast within the organ. By time point $T = 10$ it has covered most of the area of the right kidney. At the same time, the growth of the interface becomes much slower after that, since it has reached the organ borders. Subsequently, even until time point $T = 360$ the main movement is towards the local deviations and not across the borders.

Figure 11 shows the expansion of the wave front near the borders of the right kidney. The number of white marks within the interface decreases from



(a) $T = 10$ (b) $T = 20$ (c) $T = 40$

(d) $T = 50$ (e) $T = 360$ (f) $T = 370$

Figure 11: The expansion of the interface on smooth image with speed term $F(\vec{x}) = e^{-|\nabla ADW(\vec{x})|}$. Notice how slow the growth of the region is and how the holes close as it develops.

one time point to the next (only a handful of time points are illustrated). This trend is explained by the fact that the exponential speed expression is strong enough to hold back the growth of the interface across the border of the organ for a sufficiently long time. By contrast, the smoothing technique makes the inner area of the organ less diverse and enabling the interface to break the resistance and include local high-gradient-magnitude zones.

Thus, a good smoothing technique always increases the efficiency of the method. The speed term expression still remains task dependent and has to be carefully chosen for each job. We choose Equation (3) with $C = 1$ based on the results of our experiments.

4.4 Stopping criteria and morphological closing

Stopping criteria describe when and how the development of the propagating front is stopped. Existing approaches to stopping criteria include ceasing the development of the front at a fixed time [18] or histogram-based methods [4]. The development of the front in Fast Marching slows down at points with high gradient magnitude values (in particular, in the neighbourhood of organ boundaries). We devise an algorithm for capturing the moment of slow development of the front and ceasing the expansion process there.

In case of our favoured speed term function

$$F(\vec{x}) = e^{-|\nabla ADW(\vec{x})|}$$

its value decreases rapidly for high values of the gradient magnitude. This means that the development of the propagating front slows down at points in the image with high gradient magnitude values. Figure 11 shows this fact in an evident way. Expanding from a single point the front reaches the kidney border in about 40 time units. But even after another 220 units the increase in the area it occupies is negligible.

However, even if the expansion speed becomes too small in the vicinity of feature borders, the development of the front does not stop there. After a long delay, the interface eventually "overshoots" the boundary and enters the surrounding areas. This trend is illustrated in Figure 11f. Another reason for this "overshooting" is that the organ boundaries are not always very distinctive, i.e. the gradient magnitude is not high along the whole border, but undergoes variations. Then the front crosses the border where the image gradient magnitude values are low (Figure 12 illustrates the segmentation of the aorta, which is physically connected to other blood vessels).

Taking into account everything stated, we considered two options for the stopping criteria for the Fast Marching:

- stop the expansion of the front at a fixed point in time, or
- estimate the time when the development is sufficiently slow and stop it then.

Using a fixed point in time T_0 as a stopping criterion for the propagating front in Fast Marching suggests the following procedure:

1. Identify the seed points.

2. Run Fast Marching for the image volume from the initial seed points.

3. Make a set U of the points for which the time function value is less than or equal to T_0,

$$\vec{x} \in U \iff T(\vec{x}) \leq T_0. \tag{4}$$

4. Label the set U as the considered feature.

Several remarks about stopping the expansion at a fixed time point are worth making. This approach is evidently preferable for organs without very distinct boundaries. One such organ is the abdominal aorta. It has numerous connections with other smaller blood vessels. The image gradient magnitude at the connection points is usually not high and the wave expands to the other vessels easily.

An obvious drawback of the stopping criterion with a fixed time point is its inconsistency with the variable volume of each feature. If the parameter T_0 works well on only one slice of an image, the shape of the feature may still be well underestimated for that parameter in case of, say, ten slices. And, vice versa, if the organ is outlined well by Fast Marching with a fixed time stop on ten slices, it may overshoot on a single slice.

Fortunately, our scheme for seed points choice mitigates this effect. The seed points chosen from the feature approximations from the IPF are more or less uniformly distributed over the whole area of the feature and their number is proportional to the size of the feature. Then the propagating front reaches the border for different feature volumes approximately at the same time. This provides an alternative stopping criterion.

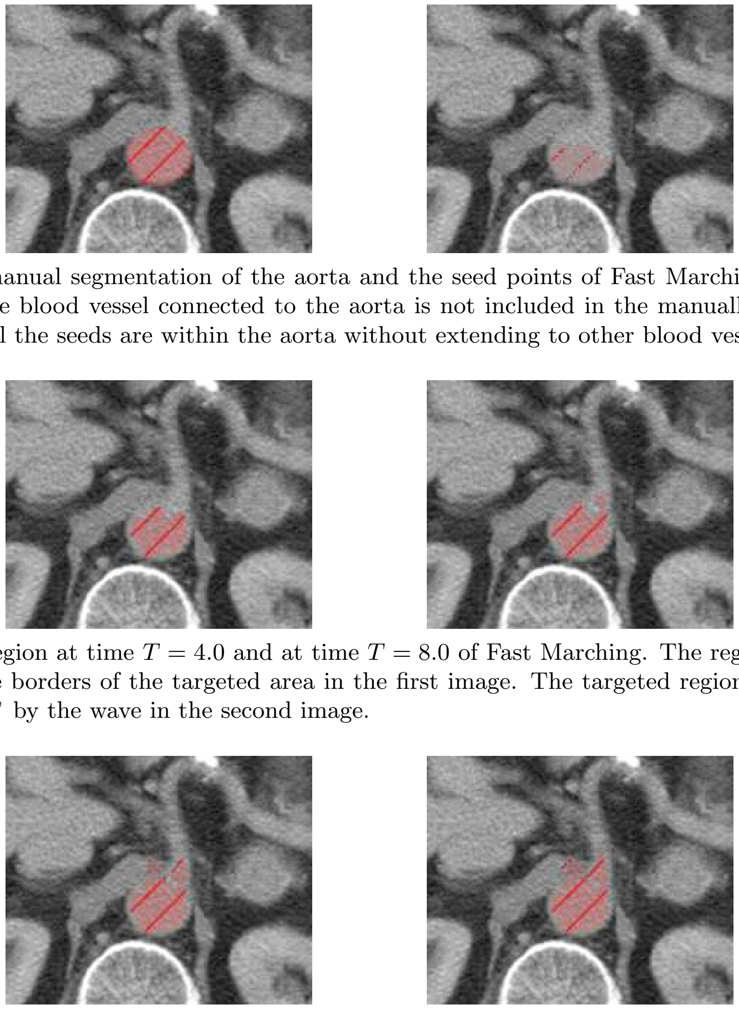
Namely, it is possible to capture slow development of the front and cease the process there. The feature segmentation procedure can be refined as follows:

1. Identify the seed points.

2. Run Fast Marching for the image volume from the seed points.

3. Sort all the lattice points into an array in increasing order of the time function values resulting from the Fast Marching method.

4. Set two parameters: a small time step ΔT and a percentage $P \in [0, 100]$.



(a) The manual segmentation of the aorta and the seed points of Fast Marching for the aorta. The blood vessel connected to the aorta is not included in the manually selected region. All the seeds are within the aorta without extending to other blood vessels.

(b) The region at time $T = 4.0$ and at time $T = 8.0$ of Fast Marching. The region is still within the borders of the targeted area in the first image. The targeted region has been "overshot" by the wave in the second image.

(c) The region at time $T = 12.0$ and at time $T = 16.0$. The region continues developing in the neighbouring blood vessel. By time $T = 16.0$ the region has already shaped the aorta and the blood vessel connected to it.

Figure 12: The development of Fast Marching for the aorta.

5. Start at the beginning of the sorted array of points and set $T_0 = \Delta T$, $N = 0$, $\Delta N = 0$.

6. While the time value of the current point is $\leq T_0$, move forward along the array, meanwhile increment the counters N, ΔN.

7. If $\Delta N < N * P/100$ or all the points have been considered, then cease the process. Label the set of the points from the beginning of the sorted array to the current position as the considered feature.

8. If $\Delta N \geq N * P/100$, then increase T_0 by ΔT and set $\Delta N = 0$. Return to Step 6.

The key idea is to consider equal intervals of points of time and count the number of positions first reached by the front for each time interval. The first such time interval with a sufficiently low number of new positions for the front in comparison with the the number of all positions already reached is selected and the propagation is brought to a stop right after that time interval.

This approach is obviously better for internal organs with well expressed boundaries. The slowdown in the movement of the interface for such features is adequate and the scheme manages to capture it. These features include the kidneys, the vertebra, etc. Our segmentation procedures for considered organs are based on this approach.

The regions output by the Fast Marching are then finalised with several iterations of morphological closing [13] in order to remove spurious holes and to smooth out the boundaries.

5 Unsupervised Segmentation

Putting all the above elements together, we construct the following *novel unsupervised segmentation procedure*:

1. Pre-process the data with windowing and anisotropic diffusion filtering (as per §2)

2. Partition the data and construct the IPF (§3)

3. For each feature of interest, filter the resulting IPF for candidate regions based on anatomical knowledge (§3)

4. Within candidate regions, choose seed points which are in an expected healthy-organ greyscale range (§4.2)

5. Choose hypersurface front propagation and its parameters, and seed it from Step 4 (§4.3)

6. Advance the front until it makes little or no progress (§4.4)

7. Finalise the labelled regions with several iterations of morphological closing (§4.4).

Steps 2–4 of the unsupervised method are illustrated on a slice in Fig. 4, and some final results on another slice in 13(a).

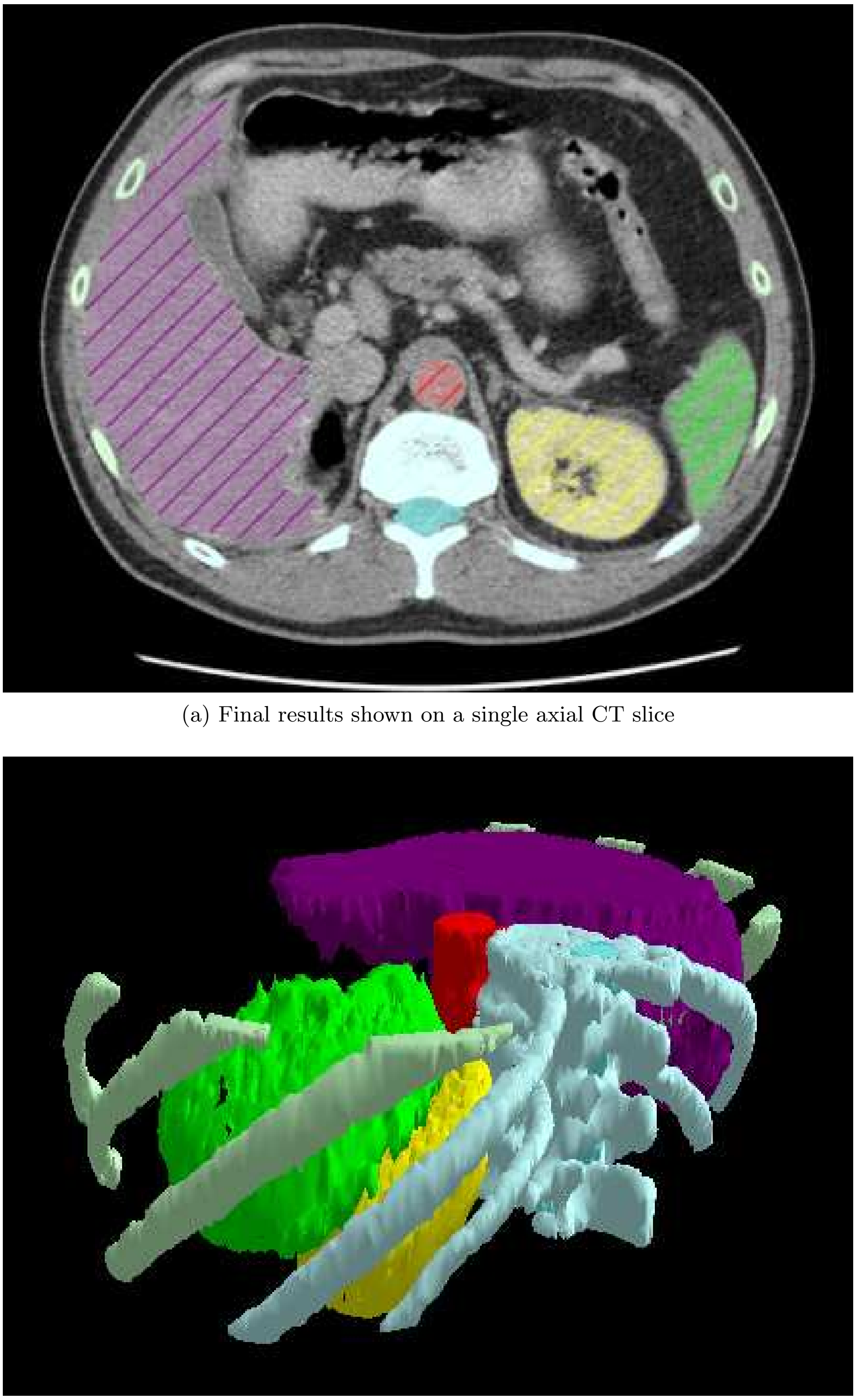
6 Results and Validation

We have illustrated our findings on a variety of single slice volumes, so as to make it possible to print each image. We have carried out extensive experiments on 3D volumes of up to 50 CT slices, allowing for both partitioning and Fast Marching to extend in all three dimensions. We varied the algorithm parameters systematically, and the overall results were consistent with what is summarised here.

6.1 Kidney segmentation

Our first quantitative evaluation is based on 7 CT volumes of 5–38 slices each. A knowledgeable operator traced contours manually for kidneys with a typical inter-operator variability of 93–94% (Dice coefficient). Table 1 shows the comparison between the manually traced gold standard and the output of our unsupervised segmentation algorithm, quantified using *Dice similarity coefficient (DSC)* [8], *Jaccard similarity coefficient (JSC)*, *true positive volume fraction (TPVF)*, *true negative volume fraction (TNVF)*, *precision (Prec)* and *relative volume difference (RVD)* [32, 6]. The Dice coefficient evaluation is split per kidney in Table 2 and is close to the inter-observer variability. The evaluation outcome can readily be compared to kidney segmentation results previously reported in the literature [7, 14].

Our dataset contains images of low resolution (5mm slice thickness, 0.68–0.78mm pixel resolution), acquired with or without contrast agent administration and covers both healthy and diseased kidneys. However, we have excluded cases with kidneys severely distorted with huge tumours, necrosis and shifted within the abdomen, as depicted in Figure 14. For such diseased organs with prominent malignant tissue, the Dice coefficient indicated a lower match of around 30%, but these represent only a tiny fraction of the overall cases studied. There is no agreed convention on whether to include tumourous tissue into the segmentation, and the decision is mostly task-dependent. In our case only healthy tissue had been labelled in the gold standard, yet the machine crossed the relatively weak boundaries between the tumour and the healthy tissue, and went on to label both, thus mis-segmenting the organ.



(a) Final results shown on a single axial CT slice

(b) 3D reconstruction of labelled features using *marching cubes*

Figure 13: 3D abdominal feature labelling

Sim M	Mean		SD
DSC	90.37	±	1.69
JSC	82.47	±	2.8
1-RVD	92.57	±	5.98
TPVF	92.81	±	4.8
TNVF	99.86	±	0.07
Prec	88.29	±	2.47

Table 1: Match of our segmentation against gold standards with different similarity measures

Kidney	DSC
A	93.016
B	91.528
C	91.394
D	91.025
E	90.824
F	90.360
G	90.024
H	87.752
I	87.396

Table 2: Dice similarity coefficient of our segmentation and gold standards per kidney

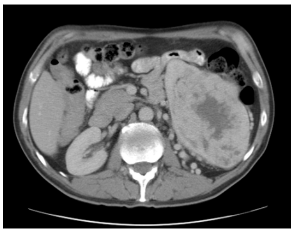


Figure 14: Extremely tumourous left kidney on which our algorithm yields only 30% DSC

Feature	Dice Coefficient (% match)		
	Mean		Std dev
Aorta	88.40	±	1.90
Spinal Cord	89.54	±	4.14
Spleen	84.14	±	5.63
Vertebra	85.70	±	7.57
Liver	81.44	±	17.73
Right Kidney	74.65	±	23.68
Left Kidney	75.70	±	23.92

Table 3: Match of our segmentation against gold standards

6.2 Multi-organ abdominal segmentation

Our second quantitative evaluation is based on 5 patient image volumes of 1–21 slices each. A consultant radiologist traced contours manually for multiple abdominal organ, slice by slice. This amounted to many hours of work for the several dozen slices.

Average DSC of the manually traced gold standard and the output of our unsupervised segmentation algorithm for each organ is reported in Table 3. For a segmented region S, Dice normalises the volume of its correctly segmented parts ($G \cap S$) over the average of S with its corresponding gold standard G thus: Dice = $\frac{2|G \cap S|}{|G|+|S|}$.

For particular images with healthy organs of expected anatomical shape and radiodensity, the algorithm performed well, with the Dice coefficient around 90%. However, for diseased organs with prominent malignant tissue, the Dice coefficient indicated a lower match (cases like the one depicted in Figure 14 were included in the dataset thus lowering the overal performance score). The overall mean in Table 3 is brought down by pathological cases (such as the one illustrated in Figure 14 where there is a disagreement over the inclusion of tumourous tissue between machine and manual segmentation results.

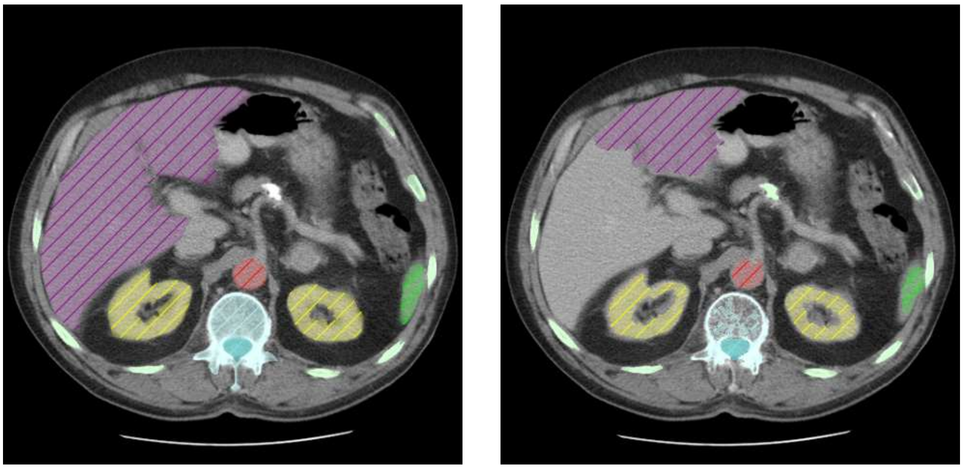
The results of labelling a 40 slice volume have been piped into Marching Cubes [33], yielding the unsupervised 3D reconstruction shown in Figure 13. Once the 3D shapes are available, evaluation techniques (such as volume calculations, evaluation of thickness, or rate of growth) become straightforward to compute, as does preparation for 3D printing.

On a laptop with average configuration our unsupervised segmentation algorithm runs for 20–30 minutes for an image volume of 20 slices. Running it on a parallel super–computer would shorten the overall time, but we are keen to maintain our image analysis software tool to lowest configuration parameters so as to make it more readily accessible to clinicians.

7 Discussion

The proposed segmentation algorithm, being based on two different established techniques, combines strengths and weaknesses of both. In other words, the quality of the end result depends on how well the IPF partition approximates the image structure and the performance of FMM at identifying organ boundaries. We consider errors of the algorithm on several image volumes of different sizes.

7.1 Image 1: A Single Slice



(a) Manually traced reference segmentation

(b) The machine-segmented result

Figure 15: Segmentation results for a single slice Image 1

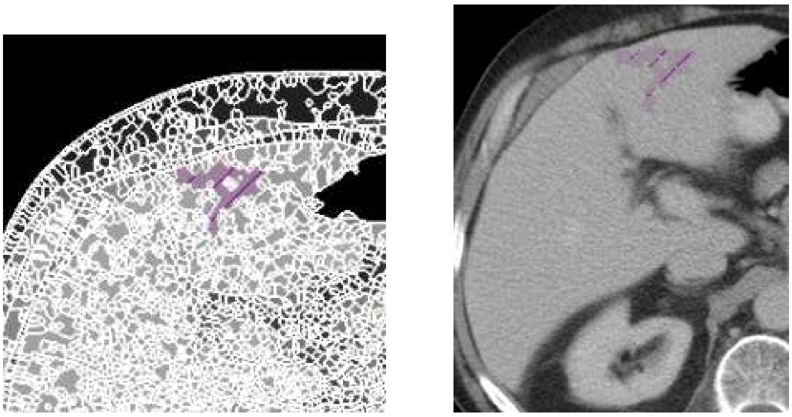


Figure 16: The IPF region filtered for liver (Step 3) and FMM seeds generated from this region (Step 4) in Image 1

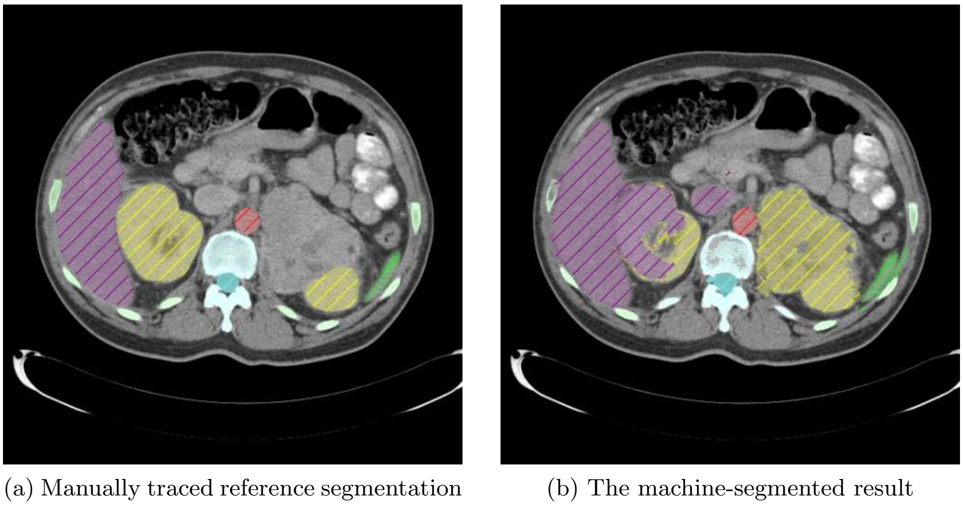
The ground truth segmentation and the result of our approach for a single slice image are presented in Figure 15. It can be seen from the images that FMM has a tendency of leaving out a narrow band at the neightbourhood of organ boundaries (kidneys, aorta and spleen). This is because it slows down the front development as soon as it gets closer to the organ boundaries where the variability of image intensities increases rapidly. We perform morphological operations on the FMM result to partially solve this issue but a more elaborate solution may be needed (see Step 7 of the segmentation procedure).

In case of the liver IPF regions covering it appear smaller than the expected average. As a result the seeds cover a very small proportion of the actual liver region and an early slowdown in the front expansion is falsely identified as arrival at the boundaries by the stopping criterion. Changing a parameter of the stopping criterion fixes this problem for the specific image.

Finally, the highly porous parts of the spine are also badly identified as can be seen from Figure 15. The reason is the local high variability of image intensities in these regions. This trend continues in other images and is the main factor bringing down the performance of our approach for vertebra identification. One solution that will fix this problem is extra smoothing of the image in the pre-processing step but this may cause the loss of important details in other parts of the image.

7.2 Image 2: Twenty Slice Volume

Figure 17 depicts ground truth and our segmentation results for a twenty slice image with pronounced renal tumours. The presence of malignant tissue changes the spatial and intensity characteristics of the image. This makes the task of segmentation harder to perform since the assumptions



(a) Manually traced reference segmentation

(b) The machine-segmented result

Figure 17: Segmentation results for a twenty slice Image 2

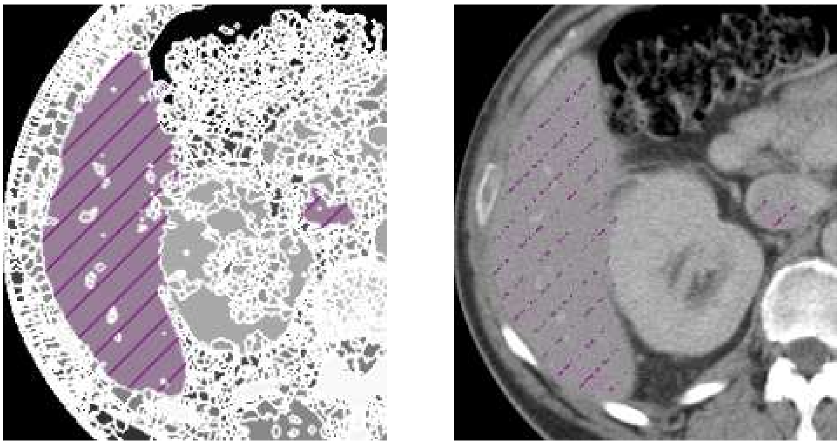


Figure 18: The IPF region filtered for liver (Step 3) and FMM seeds generated from this region (Step 4) in Image 2

from biological knowledge about the image structure may no longer be true. As a result we see poor segmentation outcome.

The region identified as liver covers big chunks of neighbouring tissues including a portion of the right kidney. This is a result of deficient partitioning of the image as shown in Figure 18. The IPF region selected in Step 3 stretches to the right (this corresponds to anatomical left) to include leftmost voxels of the right kidney and inferior vena cava. Since there are FMM seed points for liver both in the right kidney and inferior vena cava, parts of these are contained in the final result. We assume that a more careful selection of seed points should be used in such cases to guarantee

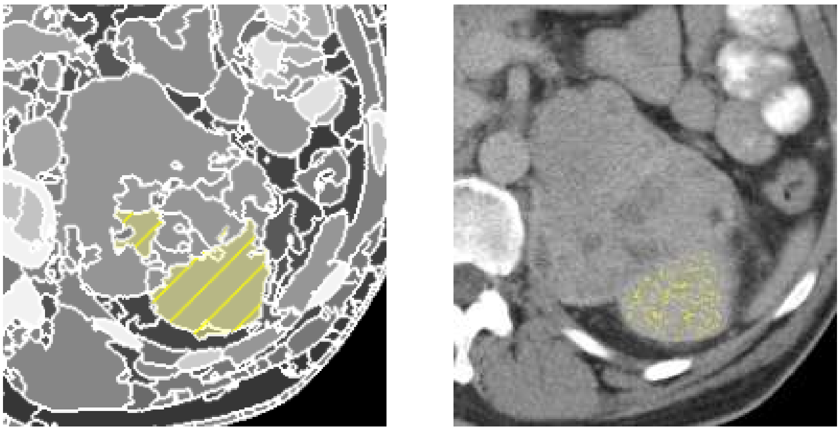


Figure 19: The IPF region filtered for left kidney (Step 3) and FMM seeds generated from this region (Step 4) in Image 2

that all seeds are inside the set of voxels comprising the considered organ.

We have a partially similar situation for the left kidney (see Figure 19). The organ is displaced by a big tumour and, because of overlapping greyscale ranges in the tumour and the kidney, the region selected from the IPF contains samples of both (the two yellow areas depicted in the first image in Figure 19 are actually connected in 3D and form a single 3D region). The filtering for seeds based on grayscale values eliminates most of the voxels in the area of the tumour but even a single seed voxel in it is enough for the FMM front to grow to cover both the left kidney and the malignant formation.

The other difficulty here is that the boundary between the kidney and the tumour is exceptionally weak. This means that an FMM front started within the kidney can easily cross the border and leak into the tumour region. Anyway, this is an instance of the hardest cases of abdominal organ segmentation and we don't expect our approach to handle it in a faultless manner. Semi-automated or manual tools can be used to correct segmentation results in such cases.

8 Conclusions and Future Work

The novelty of seeding a front propagation method automatically from a carefully crafted hierarchical structure makes our segmentation entirely unsupervised. Our current experiments on kidneys show a match between automated segmentation results and manually labelled gold standard of 90%. This is similar to the state-of-the-art algorithms [7, 14], albeit on different data. We are not aware of open access benchmarking data for the abdomen.

All our results have been validated qualitatively by a human judge.

Quantitative comparisons with other approaches will be carried out in future. Since new medical scanners produce data with increasingly higher resolution and accuracy, experiments on a wider class of volumes are needed, including on MRI data (which we have started tackling).

Future experiments will include the possibility of marching towards a feature from the outside inwards. This would be particularly suitable for segmenting bone tissue.

Whilst the procedure is fully unsupervised, in practice users (particularly clinicians) sometimes prefer to override some of the automatic features of the algorithms. For that consideration, all our software tools provide both automatic and manual parameter setting, providing a light level of supervision if desired. The overall time saver for the user is, in any case, the automation of the overall segmentation, leaving them the much less onerous option of fine tuning the final labelled volume as a post–processing task.

Acknowledgements We are grateful to Dr. Zoë Traill, of the Churchill Hospital Oxford, for providing us with abdominal CT scan data and for spending considerable time drawing gold standard contours with our software tool.

References

[1] Mimics$^{\text{TM}}$. `biomedical.materialise.com/mimics`.

[2] Simpleware$^{\text{TM}}$. `simpleware.com/software`.

[3] Serge Beucher. Watershed, hierarchical segmentation and waterfall algorithm. In Jean Serra and Pierre Soille, editors, *Mathematical Morphology and Its Applications to Image Processing*, volume 2 of *Computational Imaging and Vision*, pages 69–76. Springer Netherlands, 1994.

[4] Paola Campadelli, Elena Casiraghi, and Stella Pratissoli. Fully automatic segmentation of abdominal organs from CT images using fast marching methods. In *21st IEEE International Symposium on Computer-Based Medical Systems, 2008. CBMS'08*, pages 554–559. IEEE, June 2008.

[5] Paola Campadelli, Elena Casiraghi, and Stella Pratissoli. A segmentation framework for abdominal organs from CT scans. *Artificial Intelligence in Medicine*, 50(1):3–11, 2010.

[6] Rubén Cárdenes, Rodrigo de Luis-García, and Meritxell Bach-Cuadra. A multidimensional segmentation evaluation for medical image data. *Computer Methods and Programs in Biomedicine*, 96(2):108–124, 2009.

[7] Rémi Cuingnet, Raphael Prevost, David Lesage, Laurent D. Cohen, Benoît Mory, and Roberto Ardon. Automatic detection and segmentation of kidneys in 3D CT images using random forests. In Nicholas Ayache, Hervé Delingette, Polina Golland, and Kensaku Mori, editors, *Medical Image Computing and Computer-Assisted Intervention – MICCAI 2012*, volume 7512 of *Lecture Notes in Computer Science*, pages 66–74. Springer Berlin Heidelberg, 2012.

[8] Lee R. Dice. Measures of the amount of ecologic association between species. *Ecology*, 26(3):297–302, 1945.

[9] Stuart Golodetz, Irina Voiculescu, and Stephen Cameron. A proposed decision-support system for (renal) cancer imaging. In *Proceedings of FBIT '07*, pages 361–366, Jeju City, South Korea, October 2007.

[10] Stuart Golodetz, Irina Voiculescu, and Stephen Cameron. Automatic spine identification in abdominal CT slices using image partition forests. In *Proceedings of 6th International Symposium on Image and Signal Processing and Analysis, 2009. ISPA 2009.*, pages 117–122, Salzburg, Austria, September 2009. IEEE.

[11] Stuart Golodetz, Irina Voiculescu, and Stephen Cameron. Progress on a decision-support system for abdominal CT scans (work-in-progress paper). In Lucia Lo Bello and Giancarlo Iannizzotto, editors, *Proceedings of HSI 2009*, pages 116–119, Catania, Italy, May 2009. Best Paper Award (Telemedicine and e-Health).

[12] Stuart M. Golodetz, Chris Nicholls, Irina D. Voiculescu, and Stephen A. Cameron. Two tree-based methods for the waterfall. *Pattern Recognition*, May 2014.

[13] Raphael C. Gonzalez and Richard E. Woods. *Digital Image Processing.* Prentice Hall, 2008. 3rd edition.

[14] Marius George Linguraru, John A. Pura, Vivek Pamulapati, and Ronald M. Summers. Statistical 4D graphs for multi-organ abdominal segmentation from multiphase CT. *Medical Image Analysis*, 16(4):904–914, 2012.

[15] Ravi Malladi and James A. Sethian. Level set and fast marching methods in image processing and computer vision. In *Proceedings of International Conference on Image Processing*, volume 1, pages 489–492. IEEE, September 1996.

[16] Ravi Malladi and James A. Sethian. An O(N log N) algorithm for shape modeling. *Proceedings of the National Academy of Sciences*, 93(18):9389–9392, 1996.

[17] Ravi Malladi and James A. Sethian. Level set methods for curvature flow, image enchancement, and shape recovery in medical images. In Hans-Christian Hege and Konrad Polthier, editors, *Visualization and Mathematics*, pages 329–345. Springer Berlin Heidelberg, 1997.

[18] Ravi Malladi and James A. Sethian. A real-time algorithm for medical shape recovery. In *Sixth International Conference on Computer Vision*, pages 304–310. IEEE, January 1998.

[19] Ravi Malladi, James A. Sethian, and Baba C. Vemuri. Evolutionary fronts for topology-independent shape modeling and recovery. In Jan-Olof Eklundh, editor, *Computer Vision - ECCV'94*, volume 800 of *Lecture Notes in Computer Science*, pages 1–13. Springer Berlin Heidelberg, 1994.

[20] Ravi 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, February 1995.

[21] Ravikanth Malladi, James A. Sethian, and Baba C. Vemuri. Topology-independent shape modeling scheme. In *SPIE's 1993 International Symposium on Optics, Imaging, and Instrumentation*, pages 246–258. International Society for Optics and Photonics, 1993.

[22] Ravikanth Malladi, James A. Sethian, and Baba C. Vemuri. A fast level set based algorithm for topology-independent shape modeling. *Journal of Mathematical Imaging and Vision*, 6(2-3):269–289, 1996.

[23] Pietro Perona and Jitendra Malik. Scale-space and edge detection using anisotropic diffusion. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 12(7):629–639, 1990.

[24] Christian Rasch and Thomas Satzger. Remarks on the O(N) implementation of the fast marching method. *IMA Journal of Numerical Analysis*, 29(3):806–813, 2009.

[25] James A. Sethian. A fast marching level set method for monotonically advancing fronts. *Proceedings of the National Academy of Sciences*, 93(4):1591–1595, 1996.

[26] James A. Sethian. Theory, algorithms, and applications of level set methods for propagating interfaces. *Acta Numerica*, 5:309–395, 1996.

[27] James A. Sethian. Adaptive fast marching and level set methods for propagating interfaces. *Acta Mathematica Universitatis Comenianae*, 67(1):3–15, 1998.

[28] James A. Sethian. Fast marching methods. *SIAM Review*, 41(2):199–235, 1999.

[29] James A. Sethian. *Level Set Methods and Fast Marching Methods: Evolving Interfaces in Computational Geometry, Fluid Mechanics, Computer Vision, and Materials Science.* Cambridge Monographs on Applied and Computational Mathematics. Cambridge University Press, 1999.

[30] James A. Sethian. Evolution, implementation, and application of level set and fast marching methods for advancing fronts. *Journal of Computational Physics*, 169(2):503–555, 2001.

[31] Amy S. Tidwell. Advanced imaging concepts: A pictorial glossary of CT and MRI technology. *Clinical Techniques in Small Animal Practice*, 14(2):65–111, 1999.

[32] Jayaram K. Udupa, Vicki R. LeBlanc, Ying Zhuge, Celina Imielinska, Hilary Schmidt, Leanne M. Currie, Bruce E. Hirsch, and James Woodburn. A framework for evaluating image segmentation algorithms. *Computerized Medical Imaging and Graphics*, 30(2):75–87, 2006.

[33] Ziji Wu and John M. Sullivan. Multiple material marching cubes algorithm. *International Journal for Numerical Methods in Engineering*, 58(2):189–207, 2003.

[34] Liron Yatziv, Alberto Bartesaghi, and Guillermo Sapiro. O(N) implementation of the fast marching algorithm. *Journal of Computational Physics*, 212(2):393–399, 2006.