DEPARTMENT OF COMPUTER SCIENCE

Segmentation of medical images using a waterflow principle

Tasos Papastylianou

A dissertation submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering

October 2009 | CSMSC-09

Declaration

This dissertation is submitted to the University of Bristol in accordance with the requirements of the degree of Master of Science in the Faculty of Engineering. It has not been submitted for any other degree or diploma of any examining body. Except where specifically acknowledged, it is all the work of the Author.

Tasos Papastylianou, October 2009

Abstract

The image segmentation algorithm based on the paradigm of water flow, proposed by Liu and Nixon in 2007[25][26] was replicated in this project, and evaluated in terms of performance and efficiency, and critically analysed. The algorithm was confirmed to achieve good segmentation accuracy in a variety of images, comparing favourably with other state of the art models, such as the MAC[18], Waterflow[19] and SIOX[21] models. The basic model was optimised, greatly increasing its efficiency and usability, refined and extended, leading to increased segmentation accuracy and flexibility in segmenting a wider variety of image types. The algorithm is tested on medical images demonstrating its suitability for medical image segmentation.

Contents

1. Introduction

When we look at images, our brain effortlessly and instantly divides those images into simple and complex shapes, discrete objects and regions of interest, and categorizes these entities according to their perceived significance and semantic context [1].

Figure 1: Man walking his dog in the park

Looking at the picture on the right, for instance, we would make instant decisions as to whether a shape is a circle, a corner or a straight line, and within milliseconds we make mental associations as to what type of objects this compilation of shapes describes, and whether those objects themselves are components of a larger, more abstract and perhaps more meaningful discrete object category (such as a man, a dog, or a tree), rather than just the sum of their parts. Furthermore, a judgement is made as to whether those objects are worthy of our focus and attention in the overall image presented before us; the man and his dog are clearly the objects of interest in this picture, whereas the trees, the grass and the road are part of the background.

It is often desirable for digital images to be treated by the computer in a similar manner. The process by which we try to partition a digital image into discrete areas, or 'segments', of particular interest to the user is called 'segmentation'.

1.1 Image segmentation

Segmentation in particular is the process of applying a label to each pixel in a digital image, such that the pixels correspond to groups of pixels, or 'segments', sharing common properties; the idea is to divide the image into partitions which are more meaningful to the user, and therefore easier to process or analyse[2]. In the case of trying to partition an image into foreground and background objects, for instance, image segmentation resembles a binary classifier, classifying each pixel in the image as either belonging to the 'foreground' or the 'background' class, given a (minimal) set of instructions by the user*. While the segmented pixels need not necessarily be continuous in nature, the majority of segmentation algorithms deal with extracting discrete objects or coherent structures from an image.

The type of physical cues usually employed as baseline algorithm components are similar to ones our brains utilize themselves [3], such as areas of significantly high or low brightness, abrupt changes in brightness corresponding to edges, texture patterns, basic recognizable shapes etc. Our brain, however, has the ability to extrapolate objects by associating the partial information derived from these cues with prior knowledge and expectations about these objects. This is the reason we sometimes fail to spot an object in an image, but once we do, it becomes very difficult to 'unsee' it [4]. (cf. Pictures 2 on right, and Picture 3 on the next page)

Figure 2: An image, processed in such a way that the original object is difficult to identify without prior knowledge

* Note this refers to segmentation of standard 3-channel images. There is a somewhat different process, called 'matting', which also deals with the degree of transparency attributed to the pixels, and can classify a pixel as being both foreground and background to an extent, but this is beyond the scope of this project and classical segmentation algorithms in general.

Figure 3: The original image from which Figure 2 was derived. Knowledge of this image suddenly makes interpretation of the previous image a trivial task for the brain.

This difference between how our brain works and how segmentation is to performed by a computer, leads to a few important points regarding segmentation in general:

- Segmentation relies on user input, even if minimal, to account for the context and 'prior knowledge' part.
- Accurate segmentation is not always possible, regardless of the algorithm used, as some images may contain little in the way of physical cues, and rely predominantly on 'prior knowledge'
- Different segmentation results can be obtained with the same algorithm on the same image, according to the user initialisation used.

There are certain properties that are desirable for any segmentation tool [5]:

- Simplicity: The algorithm should rely on as little user input as possible. In particular the user should not be expected to know the intricate workings of the algorithm in order to use the tool effectively
- Efficiency: The user should not wait for an unreasonable amount of time before visible results are obtained.
- Intuitiveness and initialisation invariance: Minimal user input is required for the desired segmentation result. In particular, the result should not rely too much on the specific initialisation used, other than in roughly indicating the object to be segmented.
- Predictability: The user needs to be able to predict the tool's response to a certain input, even if unfamiliar with the underlying algorithm.
- Robustness and Accuracy: The algorithm identifies an object's boundaries as accurately as possible, and is relatively immune to image noise, incomplete information, or artefacts.
- Generality: The tool should be applicable to a variety of different image types.

Segmentation techniques have been employed in a variety of settings. Popular applications include things like face and fingerprint recognition, machine vision, motion tracking, road detection from satellite images etc. The main focus of this paper, however, is segmentation in medical images, which forms a rather distinct area of research.

1.2 Segmentation and medical imaging

There are some particular issues related to segmentation in images of a medical nature, which make the task somewhat distinct to segmentation in any other context. The term 'Medical Imaging' encompasses a variety of different modalities of digital imaging, used in a clinical context. While plain photographic images are used in certain contexts (such as dermatology atlases, retinal and endoscopic studies), these are only the tip of the iceberg. Below are some of the commoner modalities, along with particular issues potentially complicating segmentation:

- Plain Radiographic Films (X-Rays) [6]:

Images are formed by the interaction of X-Rays on photographic film (nowadays digitized). The rays penetrate the objects of interest and get absorbed depending on tissue density. This gives a 2D projection of a 3D object on film, meaning structures are superimposed. Any segmentation relying on edge information or texture needs to somehow differentiate between the superimposed structures. Furthermore, as with most other modalities, there is no colour information involved, which limits to an extent the information available for computation.

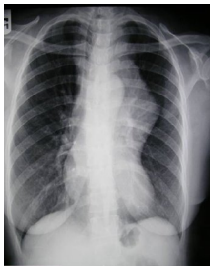


Figure 4: X-Ray film.

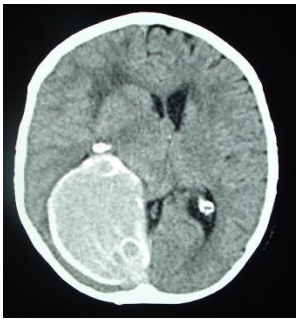


Figure 5: CT head with intravenous contrast showing a large brain tumour

- Computed Axial Tomography scans [7]:

Images are obtained using the same physical properties of plain X-Ray films, but from a variety of input directions. The outputs are then integrated electronically, to produce sections along the body. Series of sections can be further integrated and interpolated as necessary to obtain a 3D representation. Good for differentiating tissues of different density, or abnormal blood supply with the use of intravenous dyes. Images often tend to be grainy, and non-bony tissues have a limited brightness / contrast window, as they are usually of similar density, which can hinder some segmentation algorithms

- Magnetic Resonance Images [8]:

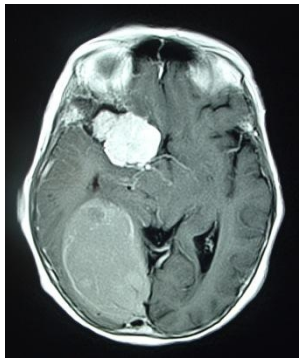


Figure 6: MRI head of the same patient as Figure 5. Note the second tumour which wasn't visible on CT

MRI uses magnetic properties of hydrogen molecules, to assess the extent to which water diffuses into the tissues. While CT is still better for visualising very dense tissue like bone, or air-rich structures like the lungs, MRI is extremely useful in visualising soft tissue anatomy in the body. The parameters of the specific examination determine the type of image produced, hence the examination is usually tailored to the tissue visualised. The image slices produced are therefore usually greyscale images of good resolution, and good tissue differentiation. For this reason, MRI naturally lends itself as a good substrate for segmentation algorithms.

- Nuclear imaging [9]:

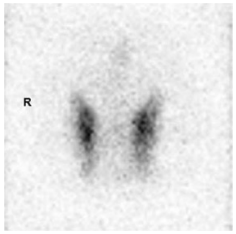


Figure 7: Thyroid scan.

Nuclear imaging typically involves the ingestion of a radioactive isotope. As the isotope decays, gamma radiation is emitted, and this can be detected using a 'gamma camera'. The images produced using this method are usually very grainy, low in resolution and with a low signal-to-noise ratio compared to most other medical imaging modalities. As there is usually little in the way of appropriate edge information, nuclear images can pose problems for algorithms relying on this feature for accurate segmentation.

- Ultrasound Scanning [10]

Figure 8: Ultrasound scan of a foetus.

This technique utilises sound waves above 20 kHz, and the property of sound to easily penetrate fluids, but not solids. The sound is directed through fluid filled cavities onto solid organs, where the sound bounces (echoes) off and is picked up by a sensor. Digital images produced using ultrasound tend to be grainy and mid to low resolution, with reasonably preserved edge information, but often exhibit artifacts, which depending on the extent to which they're present, may effect a segmentation algorithm's ability for accurate segmentation of the objects in question.

Popular applications in any of the above modalities include[11]:

- Locating tumours and other pathologies, measuring and monitoring their growth, change, improvement, etc.
- Computer-guided surgery
- As aids in the diagnosis, prognosis and management planning in a spectrum of conditions.

1.3 Types of segmentation algorithms

There are many approaches to segmentation. Simpler algorithms require assumptions to be made about the images and the nature of the objects that are to be segmented. Simple histogram-based methods, for instance, assume foreground object pixels to have colour or brightness (or some other scalar field), which is different to that of the background pixels (such as dark objects against a light background), and simply segment by choosing a suitable threshold of the measured scalar field[12]. However, methods with such constraints are of limited value in most segmentation problems of non-trivial images, and algorithms with more general applicability are preferred.

The most popular segmentation algorithms can be divided into two broad categories: algorithms that divide the image into partitions based on statistical analysis of the image data, and algorithms that converge to objects by applying physical principles over particular image attributes – naturally methods which combine the two approaches exist.

One of the most useful and commonly used attributes is edges. An edge can be defined as "a set of connected pixels that lie on the boundary between two regions"[13]. Many edge detection algorithms exist, but in general they rely on identifying areas where there are abrupt changes in intensity (i.e. a large gradient); these changes usually correspond well to edges in the image. While not all edges correspond to object boundaries (edges may also be formed by noise, artifacts, shadows etc), and not all objects of interest may have clearly defined boundaries in terms of edge information, deriving an edge map modifies the segmentation problem into trying to create a contour (i.e. a closed curve) that adheres as accurately as possible to the edges of interest, while maintaining a smooth shape, and thus delineating the objects' boundaries as accurately as possible.

Such deformable contours are called 'active contours' – or 'snakes' as an allusion to the way the contours seem to 'slither' while converging into the appropriate shape. The basic concept was introduced in 1987 by Kass et. al[14]. Essentially they are parameterized curves, initialised by the user, usually outside or inside the object of interest. The contour dynamically adjusts its shape in an attempt to minimize certain 'energy functionals', which are designed in such a way as to tend towards a minimum value as the contour converges to the shape which best delineates the object. The original framework by Kass et al. described a combination of two functionals: an external energy which is minimal when the snake is at the object boundary position, and an internal energy which is minimal when the snake has an appropriate smooth shape (mimicking elasticity or expansion, for instance, depending on the initialisation used). The snake is therefore said to be 'active', as at every stage of the algorithm it acts in a dynamic manner according to these energy functionals; the algorithm stops when the sum of these two energy functionals converges to a minimum value.

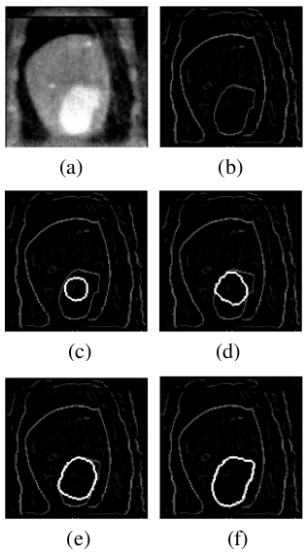


Figure 9: Classic snake.
a. Original Image b. Edge map c. Initialisation d. and e. propagation stages f. final snake.

Snakes form the basis for many popular segmentation algorithms, which attempt to enhance it by applying further physical principles to it. The Geodesic snake proposed in 1997 by Caselles et al.[15] for instance, monotonically shrinks or expands the contour towards the desired boundary, with each point moving in the direction of the normal to the contour, and at a speed determined by the image forces at that point. Gradient Vector Flow based snakes introduced by Xu and Prince[16][17] use an external force functional derived from the gradient vectors of a gray-level or binary edge map obtained from the image (i.e. vectors that always point towards the edges), making them more effective with objects containing boundary concavities, compared to traditional snakes.

More recently, the Magnetostatic Active Contour (MAC) model proposed by Xie and Mirmehdi[18] demonstrated significant improvements compared to the above models, by using an external force field based on magnetostatics and hypothesized magnetic interactions between active contour and object boundaries. This algorithm will be discussed in some more detail in the next section, and used as a comparison model representing snake-based approaches.

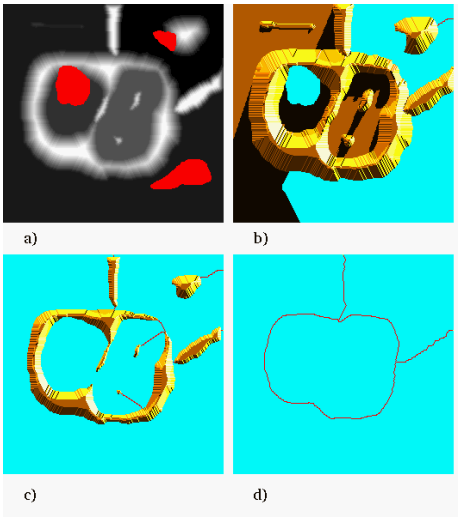


Figure 10: The Watershed Transformation.
a) Initial picture with markers b) Rising water over a topological map c) Formation of catchment basins d) Final segmentation

Algorithms also exist which, like snakes, borrow ideas from physical principles, but do not make use of the snake paradigm. One such popular algorithm is the 'Watershed' transformation[19], which is an example of a morphological approach to segmentation. This algorithm treats the image's gradient map as a topological surface, and simulates water flowing over this surface from a set of markers, and forming catchment basins, which then define the segmentation. This algorithm will be also be discussed in more detail and used for comparisons in the next section, representing morphology-based approaches.

Statistical approaches usually either involve processing the image data and partitioning it into clusters with similar statistics, or region-growing methods, where a seed-region is initialised and grows in such a way that each step leads to an expanded region with statistics that conform to the initial region. Theoretically these approaches would fare better in objects differentiated better by texture profiles rather than clear boundary information. A common clustering approach is the K-means algorithm, which partitions an image into K clusters[20]. The algorithm is initialised by selecting K cluster centres (either randomly or based on a heuristic), and assigning all the pixels in the image to one of these centres, in a way that the variance of a given measure of interest (such as intensity) is minimized. The algorithm is repeated until convergence, with new centres defined for each iteration by averaging the pixels in a cluster.

The Simple Interactive Object Extraction (SIOX) model[21], which has recently been integrated into the popular open source GNU Image Manipulation Program (GIMP)[22], uses a clustering approach based on 'k-dimensional trees'[23] to perform segmentation of images into foreground objects of interest and background objects, based on their 'colour signatures'. This algorithm, will be discussed in some more detail in the next section, and used as a comparison model representing statistical segmentation approaches.

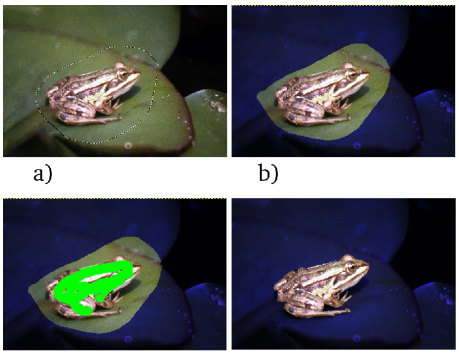


Figure 11: SIOX tool in the GIMP.
a) Outlining region of interest in initial image
b) Outlined region shown
c) Representative foreground pixels marked
d) Segmented object.

1.4 Aims and objectives

Liu and Nixon proposed in 2006 a framework for image segmentation, inspired by the paradigm of waterflow[24]. The algorithm was refined further and they published papers in 2007 demonstrating the algorithm's performance on traditional images[25] and medical images[26]. The algorithm borrows concepts from all three categories of segmentation algorithms mentioned above: a 'seed', or 'flowpoint' is initialised, which then expands like a single-layer 'waterfront' flowing over the image, segmenting objects of interest by using a combination of forces acting on the waterfront. These are derived from interactions with edge information, image statistics and the waterfront itself. The authors reported good results with their algorithm, and concluded it to be a suitable technique for medical image

processing[26], overcoming some of the theoretical limitations of current algorithms, such as obscure parameters (the waterflow algorithm parameters are intuitive as it relies on physical principles), relative initialisation invariance and segmentation of objects with holes or gaps within them.

However, when Liu and Nixon published these papers, they omitted several points that are key to assessing the true value of the algorithm:

- They chose to show only the final segmentation results, without showing how the waterfront propagates, what type of problems it may encounter, how responsive or intelligent it is with regards to the parameters used, how many steps it would typically take a user to arrive to the right parameters, etc.
- They neglected to mention any implementation details. Had the algorithm been presented as a mere framework, that would not have been entirely unreasonable. The authors, however, did in fact design and demonstrate the algorithm in their papers, but offered very little insight as to how their mathematical models were implemented in practice to arrive at the working algorithm, and whether there were any particular issues flagged during the implementation and design, such as useful optimizations, software limitations etc.
- They made no comments on the algorithm's complexity and usability, extent of reliance on human input in terms of initialisation and parameter selection, and how these affect the algorithm's performance.
- No comparisons were made to other algorithms, and images used seem to be tailored to demonstrate the particular algorithm's features. An analysis of strengths and weaknesses on a variety of pictures as compared with other algorithms would have been desirable.
- The medical images used were good quality images, with good contrast windows, from medical modalities that usually lend themselves well to segmentation, such as MRI and retinal photography. It would have been useful to see how the algorithm fares against medical images with significant noise or artefacts, which are quite common in medicine, and also against more challenging modalities such as nuclear scans and ultrasound, which are considered less easy to interpret clinically, hence successful segmentation would be of greater clinical significance.

The aims of this project were therefore to:

- Explore the framework in detail and identify particular strengths and weaknesses, as well as areas for improvement.
- Provide a useful software library which would enable this technique to be used in future research work, while highlighting any pertinent design and implementation issues

In order to fulfil the above aims, the following objectives were set:

i. *Literature review:* Study the relevant literature on image segmentation, and focus on a small number of some of the more promising and representative algorithms to analyse and draw ideas and insights from. Then study the waterflow image segmentation framework in detail as described by Liu and Nixon in [24], [25] and [26], and critically analyse it from a theoretical point of view.
ii. *Replicate the algorithm:* C++ was selected as the language of choice. A compiled language like C++ with the appropriate optimizations seems to fare better in terms of computational efficiency for the type of tasks typically involved in image processing programs[18][27]. Furthermore, the choice of C++ allowed for the OpenCV library to be used, which is a powerful, yet easy-to-use framework for image processing tasks.
iii. *Evaluation of the algorithm:* Evaluation of the algorithm's performance in terms of accuracy, efficiency and usability was carried out in a qualitative and quantitative manner where appropriate, by comparing, both visually and numerically, against the ground truth and 3 other well-respected segmentation models – MAC, Watershed and SIOX – on a variety of artificial images, designed to test particular issues and elicit strengths and weaknesses.
iv. *Evaluation on medical images:* As above, but testing on a variety of medical imaging modalities to explore strengths and weaknesses in this context.

v. *Explore potential for further improvement:* Attempt to refine and extend the framework further, apply optimizations to the algorithm, and discuss theoretical ways in which the framework could be modified to potentially yield better segmentation results. The findings may be submitted for publication if sufficient novel progress is made.

2. Theoretical background

Here we briefly review the three models introduced in the previous section, as state-of-the-art representatives of three broad categories in image segmentation approaches (i.e. snake-based, morphology-based and regional-statistics-based), followed by the Waterflow framework in detail.

2.1 The MAC model[18]

The MAC model determines the evolution of a contour, under the influence of forces applied to it from the image, based on the physical concept of magnetic fields. If we consider a pair of points P and Q having charges q_P and q_Q and velocity vectors $\vec{u}_P$ and $\vec{u}_Q$ respectively, the following relations apply according to the physics of electromagnetics[28]:

$$\vec{F}_{QP} = q_P \vec{u}_P \times \vec{B}_P, \qquad \vec{B}_P = \mu_0 q_Q \vec{u}_Q \times \frac{\hat{R}_{QP}}{4\pi R_{QP}^2}$$

where $\vec{F}_{QP}$ is the magnetostatic force exerted on q_P due to q_Q, $\vec{B}_P$ is the magnetic flux density at point P due to the point charge at Q, μ_0 is the permeability constant, R_{QP} is the distance between the two charges, $\hat{R}_{QP}$ is their unit distance vector, and $\times$ denotes the cross product.
If currents are considered instead of point charges, then given that:

$$dq\vec{u} = dq\frac{d\vec{s}}{dt} = \frac{dq}{dt}d\vec{s} = Id\vec{s}$$

we obtain the following versions of the above formulae:

$$d\vec{F}_P = I_P d\vec{s}_P \times \vec{B}_P, \qquad \vec{B}_P = \frac{\mu_0}{4\pi} I_Q \oint_{C_Q} d\vec{s}_Q \times \frac{\hat{R}_{QP}}{R_{QP}^2}$$

where, P and Q are points on the active contour and object boundary respectively, I_P and I_Q are the corresponding currents, ds_P and ds_Q are vectors corresponding to infinitesimally small segments at P and Q and C_Q is the boundary curve.

Therefore, if a constant current is applied through the boundary curves and the active contour, the contour will experience a force perpendicular to its direction, which dynamically changes in magnitude along with the contour's shape, due to its changing position within the magnetic field.

If the contour curve C_P is treated as a Level Set, then the current direction at each position of the curve can be easily found as a vector perpendicular to the gradient. Similarly for the boundaries, the intensity gradient is locally perpendicular to image edges, so a vector perpendicular to this gradient vector gives the current direction in the boundary curves. Both directions can be examined separately in either case, therefore the contour can evolve bidirectionally, segmenting both inwards and outwards.

To account for pixel quantization, the above equations are modified slightly:

$$\vec{F}(c) = I_c \vec{Y}(c) \times \vec{B}(c), \qquad \vec{B}(c) = \frac{\mu_0}{4\pi} \sum_{i \in b} I_b \Gamma_i \times \frac{\hat{R}_{bc}}{R_{bc}^2}$$

where Γ is a vector in the direction of the current, having magnitude equal to the gradient, I_b and I_c are the currents applied to the boundary and active contour respectively, and Y denotes the electric current vector, which is a unit vector in the direction of the current along the contour.

Finally, a standard stopping function is added in a weighted manner against this external force in the final model, to account for an internal force component providing elasticity, similar to GVF snakes.

The MAC model has been shown to be able to handle complex geometries and arbitrary initialisations favourably compared to previous snakes. However, as most snakes of its type, it relies primarily on boundary information rather than a judgement of region similarity, hence performance may be limited in objects relying more on texture differentiation rather than boundary information.

2.2 The Watershed transformation[19][29]

Any 2D image can be considered as a topographic surface; this means that a virtual three-dimensional landscape can be produced from the flat 2D image, such that for each pixel, the height over that pixel position corresponds to a scalar field relevant to the pixel, such as its intensity.

If one imagines flooding this surface from its troughs (or minima) and waters coming from different sources are prevented from merging (i.e. as if a dam was built at that location to prevent merging), then the landscape and hence the image is partitioned into 'catchment basins', outlined by the so-called 'watershed lines' representing the dam walls.

Since areas where the image gradient is highest correspond well to image edges, the watershed algorithm uses the gradient as the scalar field used in the topographic transformation, which leads to homogeneous areas becoming catchment basins, and watershed lines forming over edges. In order to limit the segmentation to regions of interest to the user, the user places markers within the regions of interest. Flooding then is simulated only from these markers, which ensures that the image will be partitioned to a number of segments equal to the number of markers used.

Software tools using the watershed transformation for segmentation typically also provide the ability to introduce new markers after segmentation has taken place, or to 'weld' segmented regions together, giving the user more flexibility in getting the desired result.

A different type of segmentation based on the watershed transformation, called 'hierarchical' segmentation, relies on setting thresholds (such as volume or depth) instead of using markers. The image is flooded from the minima, as per the basic algorithm, which produces all the watershed segmentations possible, and then the catchment basins are clustered together hierarchically, according to the thresholds used. While this reduces the number of partitions produced, the user has little control (other than welding regions together post-segmentation) over what objects or regions get segmented in preference to others. We will therefore be drawing our comparisons using only the marker-initialised version of the algorithm, which is more intuitive to the user.

2.3 The SIOX method[21]

The SIOX method makes use of the concept of 'colour signatures', described in 2000 by Rubner et al[30]. To explain the basic concept, imagine if we wanted to reduce a greyscale image (which

typically ranges from values of 0 to 255), to just 5 colours, conventional histogram methods might divide this range into 5 'bins' of 51 colours each, and group all colours in this way, and replace all colours of that group in the picture, with the single colour that corresponds to the centre of the bin. This method however, would make very little sense in a picture where some intensities are unequally represented in the image. So, the intensity frequency density function is divided instead into 5 clusters, that correspond better to the colours (one way to do it would be using quintiles, for instance). Each cluster could then be represented by a single point (the cluster centre) and a weight which denotes the size of that cluster. A signature for that image can then be derived as a set of the main clusters formed. In a similar manner to clustering over intensity which is a one-dimensional quantity, clusters and therefore signatures can be obtained for 3-dimensions, such as a colour map.

The SIOX algorithm requires the user to roughly outline the object of interest, making sure no 'foreground' pixels are included. This splits the image initially into an area of 'definite background' and an area which contains both background and foreground pixels. The background data is divided into clusters using a k-dimensional tree approach, which are then used to derive a colour signature for the background region. An optional rough selection of 'definite' foreground also creates a 'foreground' colour signature. The algorithm then simply classifies the pixels in the image as those belonging to the background signature, and those not belonging to it, by traversing the tree and checking whether the pixel belongs to one of the known background clusters. If foreground has also been selected, then the pixels are also checked against the foreground tree. If the pixel does not belong to either tree, the cluster with the minimum distance is selected for classification.

A post processing step is applied to the end result, to eliminate background colours that appear in the foreground. This involves identifying all spatially connected regions that were classified as foreground, and the largest, which is assumed to be the one of user interest, is kept, discarding the rest.

One of the main weaknesses of the algorithm is that it relies on appropriate initialisation; failing to outline the object of interest properly (i.e. ensuring the outlined region includes the whole object), can give unusable results. A particular strength is that it can segment multiple similar objects simultaneously in the same picture, by initialising it on just one.

2.4 The Waterflow framework[24][25][26]

The waterflow framework assumes a thin layer of water flowing from within an object in an image, and over the image in such a way that it expands in area and stops when this layer completely covers the object in question. The image itself is treated as a hybrid between a flat 2D surface and a 3D topographic surface, in that the water encounters physical obstacles that stop its flow, but at the same time, if the waterfront has accumulated enough velocity, it can overcome those obstacles as a single thin layer of water molecules (as if seeping through), without necessarily simulating a rising level of water, like in the watershed transformation.

The algorithm simulates this in the following manner:
1. Every contour pixel experiences attractive or repellent forces from surrounding elements
2. The total force acted on a contour pixel conveys an outward velocity to it (*acceleration phase*)
3. Image forces act against the accelerated pixels, depending on their position and direction.
4. If the acquired velocity is enough to overcome the image forces, flooding of neighbouring pixels takes place in the appropriate direction. (*exterior movement phase*)
5. Steps 1 to 4 are repeated until there is no more expansion.

Steps 1-4 are dealt with separately in detail, in the subsections that follow.

2.4.1 Water contour forces

The main force causing expansion of the water layer is the water 'pressure' originating from a 'seed' or 'flowpoint(s)' initialised by the user, from which the water expands outwards. This is translated as a constant repulsive force between all water molecules in the layer; since inner molecules are surrounded by other water molecules, the repellent forces cancel each other out. On the waterfront (i.e. contour), however, the repellent forces from inner water molecules, push the contour outwards, making the water layer expand.

A second 'internal' force, acts on the expanding waterfront, which the authors termed 'surface tension'. Outside its physical context, the term may be a bit misleading, as it is not actually acting on the surface of the expanding water layer, but rather on the expanding contour. Like surface tension, however, it enables the waterfront to maintain as smooth a shape as possible. This is done by applying an attractive force between all water contour elements.

Finally, an 'adhesive' force is described, alluding to water's tendency to stick to a container's walls. This is translated as an attractive force emanating from image edges, which is proportional to the particular edge strength, and acts on the expanding contour.

All the forces above are described using an inverse-square law, in a manner similar to particles in a Gaussian force field. While this does not necessarily conform to a truly physical framework of water and its related forces as they would occur in nature, it provides consistency among the three forces, making their respective contributions easier to analyse.

In formal mathematical language, this is expressed as follows:

$$\vec{F}(\vec{r}_i) = \sum_{j \in W,\, j \neq i} L(\vec{r}_j) \frac{\vec{r}_j - \vec{r}_i}{|\vec{r}_j - \vec{r}_i|^3} \tag{1}$$

where $\vec{r}_i$ and $\vec{r}_j$ are vectors corresponding to two pixels *i* and *j* in the image, $\vec{F}(\vec{r}_i)$ is the attractive force experienced by *i* due to *j, W* denotes the entire image area, where these forces apply, and $L(\vec{r}_j)$ is the weight attributed to pixel *j* in terms of its force exerting potential. If *j* is a non-flooded pixel, then *L* takes values from 0 to 1, reflecting the edge strength at that point. If it is a water contour element, *L* takes the value of 1, whereas if it is an inner water element, it takes the value of -1 to act as a repellent force.

2.4.2 Acceleration phase

The total force acted on a pixel, can be used to calculate a velocity achieved by that pixel as a result, using flow principles. The rate of water flow can be defined as the number of water molecules passing through a given surface per unit time. Flow can be related to the pressure difference at the point of interest through the following equation:

$$f_r = \frac{\Delta P}{R} \tag{2}$$

where f_r is the flow rate, ΔP is the pressure difference in the direction of flow, and *R* is the resistance (due to viscosity, the fluid channel, temperature, etc). Flow is also related to the velocity of the water molecules in the direction of the flow through the following equation:

$$f_r = A V \tag{3}$$

Combining equations (2) and (3) we get:

$$V = \frac{\Delta P}{A R} \tag{4}$$

Since both V and ΔP are in the direction of flow, and Pressure / Area = Force, (4) can be rewritten as:

$$\vec{V}(\vec{r}_i) = \frac{\vec{F}(\vec{r}_i)}{R} \tag{5}$$

In other words, the water element achieves a velocity V in the direction of the total force F acted on it. The resistance variable, can be related to particular image attributes, which would curtail the evolution of the expanding water layer in a manner that benefits segmentation. The authors decided to define resistance as follows:

$$R = e^{kE} \tag{6}$$

where E corresponds to the edge strength at the particular pixel in question, and k is a user adjustable parameter controlling the rise of the exponential curve. Areas of the image with a high edge response would therefore limit the speed acquired by the expanding waterfront at that region.

2.4.3 Image forces

Image forces represent the 'obstacles' to the flowing water. The total image force acted on the moving pixel is a combination of two force functionals: an edge-based, and a region-based functional.

The edge-based functional is defined as the gradient of the edge map. This leads to a vector field which is always pointing towards image edges, and whose magnitude is dependent on the edge strength. It is defined as:

$$\vec{F}_E = \nabla E(x_t, y_t) \tag{7}$$

where ∇E is the gradient of the edge map, and x_t and y_t are the coordinates of the flow target, because the force is presumed to act on the moving contour element along its course. Since more precise (i.e. thin) edges produce a clearer edge gradient, they should also prove more useful in terms of providing a significant and well localized force, preventing an expanding contour from 'escaping' an edge.

The region-based functional is defined as a vector, in the same direction to that of the expanding contour pixel, and magnitude determined by the statistics of the flooded and unflooded regions in the area near the pixel in question, as follows:

$$F_R = -\frac{n_{int}}{n_{int}+1}(I(x_t, y_t) - \mu_{int})^2 + \frac{n_{ext}}{n_{ext}-1}(I(x_t, y_t) - \mu_{ext})^2 \tag{8}$$

where $I(x_t,y_t)$ is the intensity of the target pixel, subscripts *"int"* and *"ext"* denote flooded and unflooded regions respectively, n denotes the number of pixels in the corresponding region, and μ denotes the mean intensity of the corresponding region. The function of equation (8) is clearer if we consider the 3 extreme cases:

- If $\mu_{int} \approx \mu_{ext}$, then $F_R \approx n_{int} + n_{ext}$, therefore $F_R \geq 0$, i.e. the forces pushes the pixel in the direction of flooding.
- If $I(x_t, y_t) = \mu_{int}$, then again $F_R \geq 0$.
- If $I(x_t, y_t) = \mu_{ext}$, then $F_R \leq 0$, i.e. in the opposite direction to flooding.

In general, the more the target pixel's intensity is compatible with the intensity of the pixels in the flooded region, the more likely it is to get flooded, otherwise flooding is opposed, to the extent that the pixel is more compatible with the unflooded region compared to the flooded one.

2.4.4 Exterior movement phase

The accelerated contour pixel needs to overcome the image forces, if it is to flood the neighbouring pixel. According to kinetics, if a constant force F_0 is applied to a body, causing a displacement S_0, the work W_0 produced is:

$$W_0 = F_0 S_0 \tag{9}$$

A moving body A of mass m_A and velocity v_0 is said to possess a kinetic energy E_k, where:

$$E_k = \frac{1}{2} m_A v_0^2 \tag{10}$$

If F_0 when acted on *A,* had brought it to a complete halt after a displacement equal to S_0, then the work W_0 done would have been equal to the moving body's initial kinetic energy E_k, i.e.

$$\frac{1}{2} m_A v_0^2 - F_0 S_0 = 0 \tag{11}$$

We can apply this result directly to the water contour pixels. F_0 corresponds to the total image force F_T, which is a weighted combination of the edge-based force F_E and region-based force F_R, via a user-defined parameter, α, taking values in the range (0,1):

$$F_T = \alpha F_E + (1-\alpha) F_R \tag{12}$$

Finally, from (11) and (12) we consider the following expression:

$$\lambda v^2 + F_T \geq 0 \tag{13}$$

where F_T is a scalar, which is positive if the direction of the force is the same as the velocity of the contour pixel, and negative otherwise, and λ is a regularization constant, chosen by the user, as a means of controlling the trade-off between the two energy terms. In all cases where the above expression is true, flooding of the target pixel occurs; conversely, if F_T is in the opposite direction to the expansion, and the velocity is not adequate to satisfy expression (13), then no flooding occurs.

3. Basic framework implementation and assessment

3.1 Implementation

As mentioned earlier, the algorithm was implemented in C++, using the OpenCV image processing library[31]. In this section some of the main design decisions are explored.

3.1.1 General user interface.

The OpenCV library provides a simple graphical interface which allows displaying images on screen and basic event handling. All other menu options are implemented in the console. The software provides the ability to simulate a particular number of steps, so that the evolving contour can be traced visually, as well as the ability to reset the image (while preserving the parameters), freeze the current segmentation (allowing for consecutive segmentations to occur using different parameters), toggle between the normal image and the edgemap, save particular frames, change contour colours etc. If specified at the command-line, frames can be exported to a folder, so that an animation of the evolving contour and resulting segmentation can be produced.

The user initialises the algorithm by drawing with the mouse directly on the image or on the visible edgemap if preferable. While single flowpoints are usually enough for initialisation, the user is free to draw any curve they feel might result in a better or faster segmentation.

The parameters used in the basic framework are λ, k and α, as specified above. Parameters can be changed before initialisation, or in the middle of the contour evolution, even though that is not recommended as a segmentation strategy (it can be useful, however, if segmentation is slow, or terminates early, in which case altering the parameters might help the contour evolve better.)

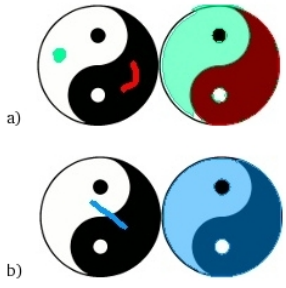
Segmentation results are saved in the form of a human-readable (ascii) portable bitmap (.pbm) format[32]. Separate utilities were designed to manipulate these (such as invert, add or subtract as part of a complex segmentation strategy) and analyse them against the ground truth or other segmentation results (i.e. from other algorithms), producing useful statistics. They can also be used directly in more sophisticated programs like The GIMP, to easily specify a selection mask over the original image, therefore enabling side-by-side use of the software as part of a broader image manipulation task.

3.1.2 Basic structures

The software only deals with one image at a time, which is loaded at execution through command-line arguments. The decision was therefore made to treat the associated OpenCV Image Objects as a globally available structure, accessible to all the different functions in the program. Two such objects are made available, one for displaying the main image, and one for the edge map. These objects are only used to provide visual information to the user, and are not involved in any of the algorithm's calculations.

The structure of the OpenCV Image Object is fairly rigid, so instead of attempting to extend it, a second structure was made globally available, called the pixel array. A pixel, in this context, is a class, which holds the information used for the main algorithm's calculations, such as a pixel's initial colour and edge strength, or whether it is a background element, an inner water element, a contour water element, or a frozen element (i.e. appearing in the final segmentation result, but considered as a normally acting pixel during the current segmentation attempt). It also holds information on whether a pixel has been altered during what is termed a 'watercycle' (i.e. a single iteration of the algorithm's 4 main steps described above), which helps track any changes to be displayed on screen (avoiding unnecessary updating of every pixel in the image) between cycles, and is also used to signal termination of the algorithm when no more alterations are taking place. An array of such 'pixel' objects is built for all the pixels in the image.

3.1.3 Initialisation



The user initialises the algorithm by drawing 'flowpoints' on the image using the mouse. These can be drawn anywhere on the object, as long as they don't cross any boundaries that the user would otherwise have liked to be segmented. An alternative strategy is to initialise the algorithm outside the object of interest, essentially surrounding the object of interest with water; the user can then invert the segmentation result using the utilities mentioned above. Initialising the algorithm across boundaries can be used as a segmentation strategy, where the user wants to make sure certain boundaries are intentionally not segmented.

Figure 12:
a) Separate initialisation preserves the two halves.
b) Cross boundary initialisation excludes the boundary from segmentation

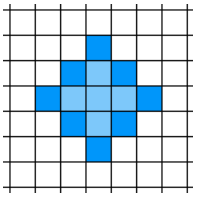


Fig. 13: Flowpoint

When a flowpoint is added, a diamond consisting of 13 pixels is drawn, rather than a single pixel. This is to ensure that even a single point will have enough initial 'pressure' to enable it to expand autonomously. The contour elements are drawn using solid colour, whereas the inner water elements are semi-transparent, a convention which is kept during the evolution of the expanding water layer as well. Further flowpoints can be added at any time, regardless of any existing evolving contours.

3.1.4 Edge strength map

As mentioned above, accurate and precise edge detection should lead to better localised, well defined edge-forces acting on the expanding contour. A simple 'intensity-gradient-magnitude' map, while adequate for clearly defined images, often gives thicker edges in non-trivial images, depending on the degree of smoothing applied, if any, and how sharp the transitions are at object boundaries in the original image. Algorithms giving precise, thin edges, on the other hand, are often prone to small gaps (usually where the gradient magnitude falls below a certain threshold), particularly in noisy images. The authors postulated that such small gaps in thin edges should be overcome by the water contour's surface tension. The canny edge-detection algorithm was therefore used to build the edge strength map, as one of the most respected edge-detection algorithms, known for its accurate, single-pixel wide edge detection[33]. In order to obtain an edge strength map, smoothing was applied to minimize noise, and then the canny algorithm was iterated over several thresholds and the values added together to produce a single map, leading to stronger edges having a higher value (highest possible being 255, to correspond to the typical greyscale range for easy visualisation).

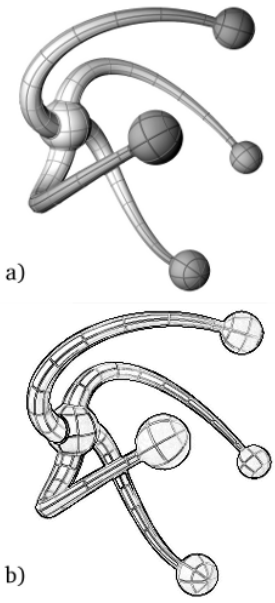


Figure 14: Edge strength map example.
a) Original Image
b) Corresponding Edge strength map (inverted) based on the Canny edge-detection algorithm.

3.1.5 Water forces

To ease calculations, all vector fields in the algorithm were split into their horizontal and vertical vector components. This enables manipulation of the vectors as simple scalar fields, with a +ve or -ve sign denoting the appropriate direction in the respective axis. The water contour forces, were therefore calculated for each pixel by transforming equation (1) as follows:

$$F_i = \sum L_{(x,y)} \frac{\Delta x}{(S_{xy})^3} \quad \text{and} \quad F_j = \sum L_{(x,y)} \frac{\Delta y}{(S_{xy})^3} \quad (S_{xy} \neq 0) \tag{14}$$

where F_i and F_j are the horizontal and vertical components of the resulting force on the contour pixel, Δx and Δy are the horizontal and vertical distances respectively between the contour pixel and each force-exerting pixel, S_{xy} is $\sqrt{\Delta x^2 + \Delta y^2}$ (i.e. the overall distance between the two pixels), and L is the edge strength of the force-exerting pixel divided by 255 to convert it to the range $0 \leftrightarrow 1$.

3.1.6 Acceleration phase

The velocity is then obtained by dividing each force component with the resistance as per equation (5), and storing them in a 'Velocity' object containing two member variables corresponding to the horizontal and vertical velocity components. A velocity object array is constructed to account for all the existing contour pixels.

The two steps, described above are naturally condensed to a single step for efficiency reasons. Despite this, however, this still represents one of the biggest 'bottlenecks' in the algorithm, as for each contour pixel, the algorithm needs to iterate over the entire image to calculate an appropriate force. Note that a predetermined 'force-field' map to speed up the process, would not have been possible, as the evolution of the contour redefines the forces exerted by the image (newly flooded areas emerge producing water-to-water forces, and flooded edges cease to have their 'adhesive' effect).

3.1.7 Image Forces

The edge strength gradient force is calculated locally, using a 3×3 pixel area, centred over the *target* pixel (i.e. the pixel to be flooded if expansion is successful). Again, the forces are dealt with in horizontal and vertical components, as in figure 15. Note that the direction of this edge gradient force is independent of the direction of flooding, as it always points towards the stronger edge.

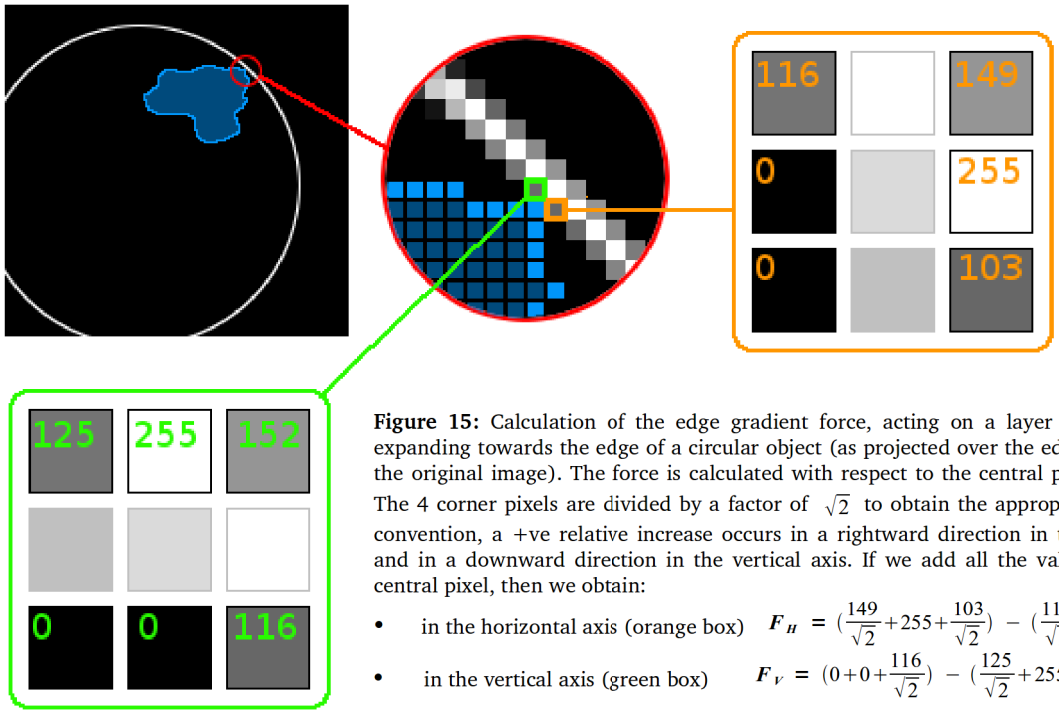


Figure 15: Calculation of the edge gradient force, acting on a layer of water, which is expanding towards the edge of a circular object (as projected over the edge strength map of the original image). The force is calculated with respect to the central pixel in a 3×3 grid. The 4 corner pixels are divided by a factor of $\sqrt{2}$ to obtain the appropriate projection. By convention, a +ve relative increase occurs in a rightward direction in the horizontal axis, and in a downward direction in the vertical axis. If we add all the values relative to the central pixel, then we obtain:

- in the horizontal axis (orange box) $F_H = (\frac{149}{\sqrt{2}}+255+\frac{103}{\sqrt{2}}) - (\frac{116}{\sqrt{2}}+0+0) = \mathbf{351}$
- in the vertical axis (green box) $F_V = (0+0+\frac{116}{\sqrt{2}}) - (\frac{125}{\sqrt{2}}+255+\frac{152}{\sqrt{2}}) = \mathbf{-369}$

The regional-statistics force is calculated as per equation (8), but the result is multiplied by -1 if the flooding direction is leftwards in the horizontal component, or upwards in the vertical component. This is done to ensure the force is acting in the direction of the velocity component, as intended. Note that this step also involves iteration over the entire image, representing the 2nd important bottleneck in the algorithm.

3.1.8 Exterior movement phase

Expression (13) defines the velocity as a positive scalar field, and the total force F_T as positive if it occurs in the same direction as the velocity vector, and negative otherwise. In this implementation, in order to account for the use of horizontal and vertical components with fixed signs corresponding to absolute rather than relative directions, the exterior movement step is modified slightly:

- if the velocity component is ***negative,*** then movement will only occur if $F_T - \lambda v^2 \leq 0$
- if the velocity component is ***positive***, then movement will only occur if $F_T + \lambda v^2 \geq 0$

where $F_T = \alpha F_{edge} + (1-\alpha) F_{regional}$ as per equation (12).

3.2 Assessment and preliminary results.

Liu and Nixon used custom made images designed to illustrate some of the abilities of their waterflow model[24][25][26]. These are recreated in this section as faithfully as possible, to assess the extent to which our implementation conforms to their findings. Both qualitative and quantitative evaluation is performed where appropriate; Liu and Nixon used a Mean Squared Error (MSE) statistic for quantitative evaluation, which compares the distances between contour points and their nearest ideal ground truth contour point. While a MSE result of zero is clear enough to interpret, a positive value is generally not very intuitive in itself and gives no information whether the error results from over or under-segmentation. Since this kind of segmentation is essentially a binary classification problem, a confusion matrix based analysis is used instead, which produces intuitive and easy-to-visualise results. In all the subsequent analyses, both qualitative and quantitative, entire object segmentation, rather than just contour points, is being considered (so for example, a true positive instance corresponds to a pixel, where this has been classified as belonging to the object of interest, and according to the ground truth this is indeed the case).

		Ground Truth		
		TRUE	FALSE	
Segment-ation	Positive	True Positive	False Positive	→Precision
	Negative	False Negative	True Negative	
		↓ Sensitivity	↓ Specificity	↘ Accuracy

- Accuracy = TP + TN / (TP + TN + FP + FN)
- Precision = TP / (TP + FP)
- Sensitivity = TP / (TP + FN)
- Specificity = TN / (FP + TN)

Figure 16: Example of a Confusion Matrix.

The qualities Liu and Nixon focused on specifically are multi-object detection, ability to handle complex geometries, weak-contrasted edges, and immunity to noise.

3.2.1 Multi-object detection

Figure 17 demonstrates the first concept, where three distinct objects are placed in the target image. Conventional snakes often encounter difficulties in segmenting distinct multiple objects, resulting in a single contour that includes all objects within it (i.e. the classification results in many false positive results). Recent snakes, such as the geodesic model[15], overcome this issue by treating the evolving contour as a Level Set of a 3-dimensional closed shape, but this translation of the problem to a higher dimension greatly increases the computational complexity of the algorithm[24].

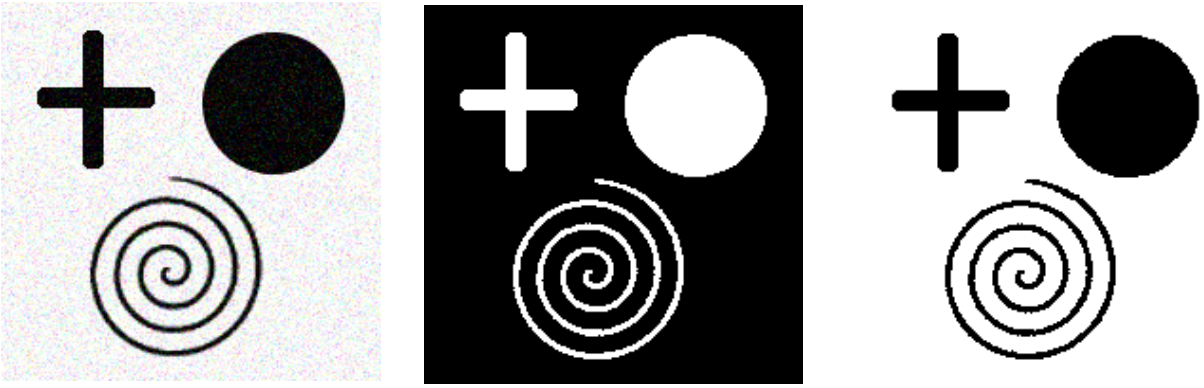


Figure 17: Multi-object detection. *Left:* 200×200 image, corrupted with 10% Gaussian noise, as per the original paper[24]. *Middle:* Segmentation using a single external 'flowpoint' seed. Parameters used: $\lambda = 10$ (as default value of 1 led to small gaps in the segmented area, due to noise), $k = 1$, $\alpha = 0.5$. *Right:* Segmentation using separate flowpoints per object. Parameters used: $\lambda = 200$ (higher value needed to help the water flow through the noisy thin spiral line), $k = 1$, $\alpha = 0.5$

As mentioned before, the only initialisation restraint of the waterflow model is that the boundaries of any objects we wish to segment cannot be crossed, therefore 'internal' or 'external' flowpoints are

used. The middle image shows the segmentation achieved using a single external flowpoint (black pixels correspond to 'positively' segmented regions, and white pixels correspond to 'negative' pixels). Inversion of the segmentation map is a trivial operation to perform, therefore this is a perfectly valid segmentation strategy. Segmentation obtained using separate flowpoints per object is shown in the image on the right for comparison.

3.2.2 Complex geometries

Another common problem in many segmentation algorithms is the ability to handle complex shapes, and in particular shapes which contain a mixture of thick and thin lines in a variety of orientations[24][18]. The target image below on the left, designed by Liu and Nixon (figure 18) incorporates all these features. Liu and Nixon used an image of size 512×512. Here successful segmentation was achieved on both 600×600 and 400×400 images, using a central flowpoint.

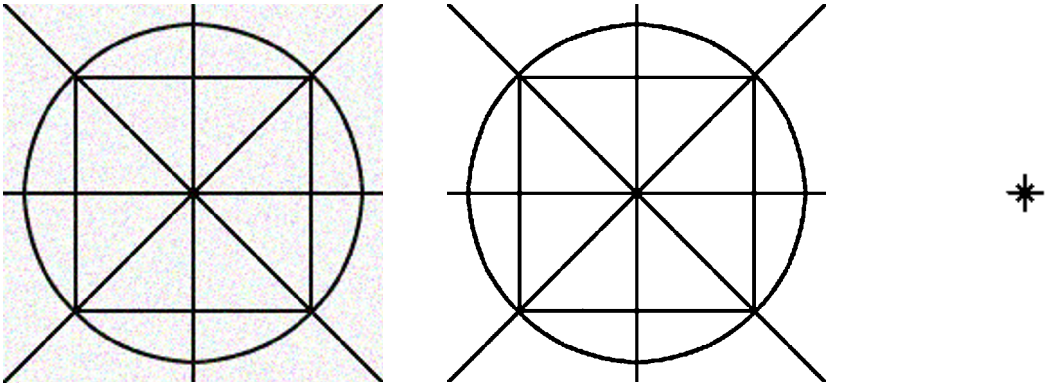


Figure 18: *Left:* The target image, containing thin horizontal, vertical, diagonal, and curved lines. 10% Gaussian noise is applied on this test as well. *Middle:* Successful segmentation in both the 600×600 and 400×400 versions of the target image, using only the default parameters ($\lambda = 1$, $k = 1$, $\alpha = 0.5$). *Right:* Unsuccessful segmentation attempt on a 200×200 target image.

Segmentation of a 200×200 version of this image, however, was unsuccessful with the basic model for two reasons:

- The 'pressure' force is very limited due to the narrow spaces, and is therefore not enough to overcome the surrounding edges causing 'adhesion' forces. Note that increasing the regularization constant λ does not have the desired effect, as both pressure and adhesion forces increase to the same extent. If λ is increased even more, eventually it causes the water to seep through the object boundary entirely. While this can be curtailed by increasing the resistance modifier k, water still leaks out from areas with edge gaps, where the resistance effect does not occur.
- Edges formed in lesser resolution images may produce unwanted artefacts due to quantization, since they need to be at least one pixel wide. Such artefacts are shown in figure 19 (taken from the 200×200 version). The converse is also true: A useful segmentation strategy in low resolution images where an edge effect is desirable, is to scale the image to a larger resolution; even though the image quality itself doesn't improve with the scaling, the edges will correspond more accurately to the object boundaries.

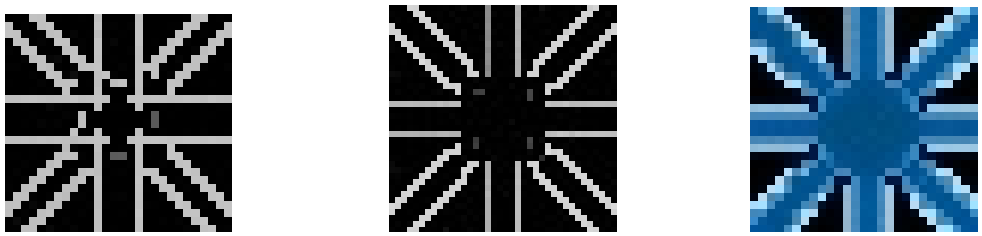


Figure 19: *Left:* Edge strength map derived from the 200×200 image version from figure 18. Note the edge artefacts, effectively blocking the way to the diagonal branches. *Middle:* The equivalent edge strength map from the 400×400 target image version. Note how this corresponds a lot better to the actual object boundaries. Also note there are some small gaps where the edges join into corners. *Right:* the segmentation outcome over the edge map in the middle. Surface tension prevents water from leaking out of the small gaps.

3.2.3 Weak contrasted edges

Figure 20 shows a 200×200 image of a white circle, cut out from a 15% Gaussian noise field. Liu and Nixon used this type of image to test the ability of the algorithm to segment structures with weak contrasted boundaries, essentially making the segmentation rely on the region-statistics image force. The region-statistics force however, is applied on a per-expanding-pixel basis, rather than on a cluster. So, a 'white' pixel in the noisy background, would still be classified as an object pixel. The authors postulated that the surface tension should be enough to prevent the contour from 'leaking out'. They even demonstrated a successful segmentation of a similar image in their paper.

In our test, some leakage was indeed observed, which leads to the conclusion that the amount of surface tension isn't quite enough to prevent the formation of the pseudopod-like structures seen in figure 20. Perhaps Liu and Nixon used a tweaked version with increased surface tension to obtain this result, or it may be a serendipitous result of the particular image they used. Manually increased surface tension does indeed produce the expected result, as we shall see in the next section.

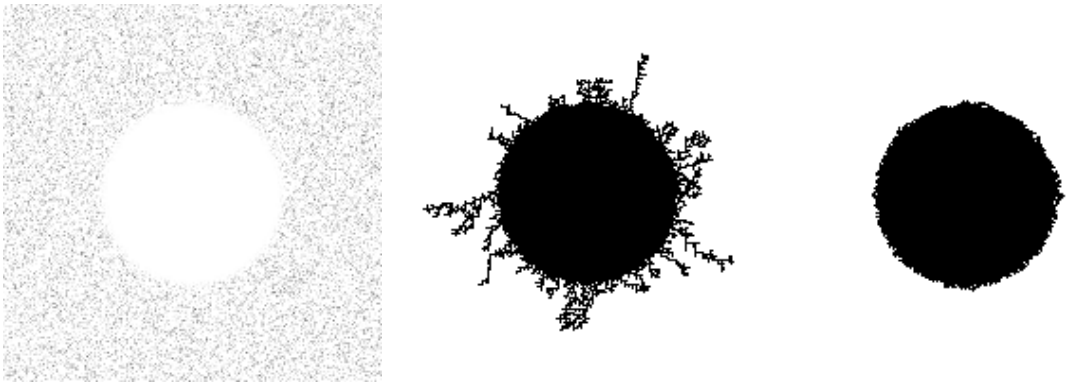


Figure 20: *Left:* 200×200 image of a white circle cut out of a 15% Gaussian noise field, therefore producing poorly defined boundaries. *Middle:* The segmentation achieved using the basic framework with the parameters set at $\lambda = 0$, k = N/A (can be anything since overall effect of speed will be zero due to λ) and $\alpha = 0$ (i.e. only region-based image forces are used). *Right:* Improved result, using increased surface tension. (see refinements section)

The fact that the surface tension in the basic model described by Liu and Nixon isn't ideal, can also be seen from the way the water layer expands. If surface tension was ideal, one would expect the expanding water layer to assume a more circular shape, just like real water droplets do. Both in the demonstrations shown in Liu and Nixon's papers, and in our implementation of their basic model, a diamond-shaped expanding contour is observed instead (figure 21). The segmentation result of figure 20 is therefore not surprising, and just reflects the need for refinements to the algorithm.

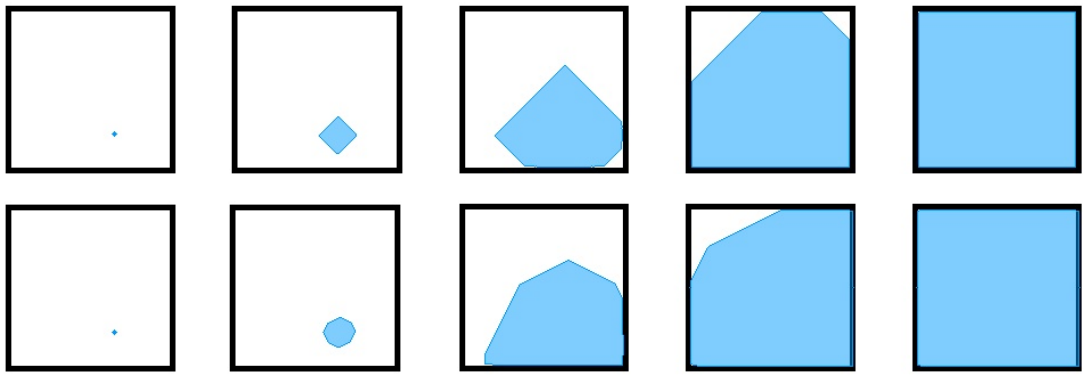


Figure 21:
Top row: Evolution of a water layer using the basic framework's definition of surface tension
Bottom Row: Evolution using disproportionately increased surface tension (see refinements section)

3.2.4 Immunity to Noise

Performance against noise was measured by evaluating the model against varying levels of two types of noise: Gaussian noise and Impulsive noise.

In Gaussian noise, each pixel in the image has an added noise value (+ve or -ve) over the intended signal. The added noise follows a normal distribution, meaning that few pixels will have too much noise applied, and most pixels will have noise levels close to zero. In this context, 50% noise was calculated as follows: Increasingly noisy images were classified against the pure signal. The noisy image which achieved 50% (i.e. random) accuracy, was selected to represent 100% noise. If it took 8.00cmκ iterations to reach the 100% image, then 50% noise corresponds to the $\kappa/2^{th}$ image, 25% to the $\kappa/4^{th}$ image, etc.

In Impulsive noise a predetermined percentage of pixels are chosen randomly in an image, and their value is changed to a random colour. In this context 100% means all the pixels are affected, 50% half are affected, etc. This usually results in noise which looks 'harsher' compared to Gaussian noise.

The results are shown in the figure below.

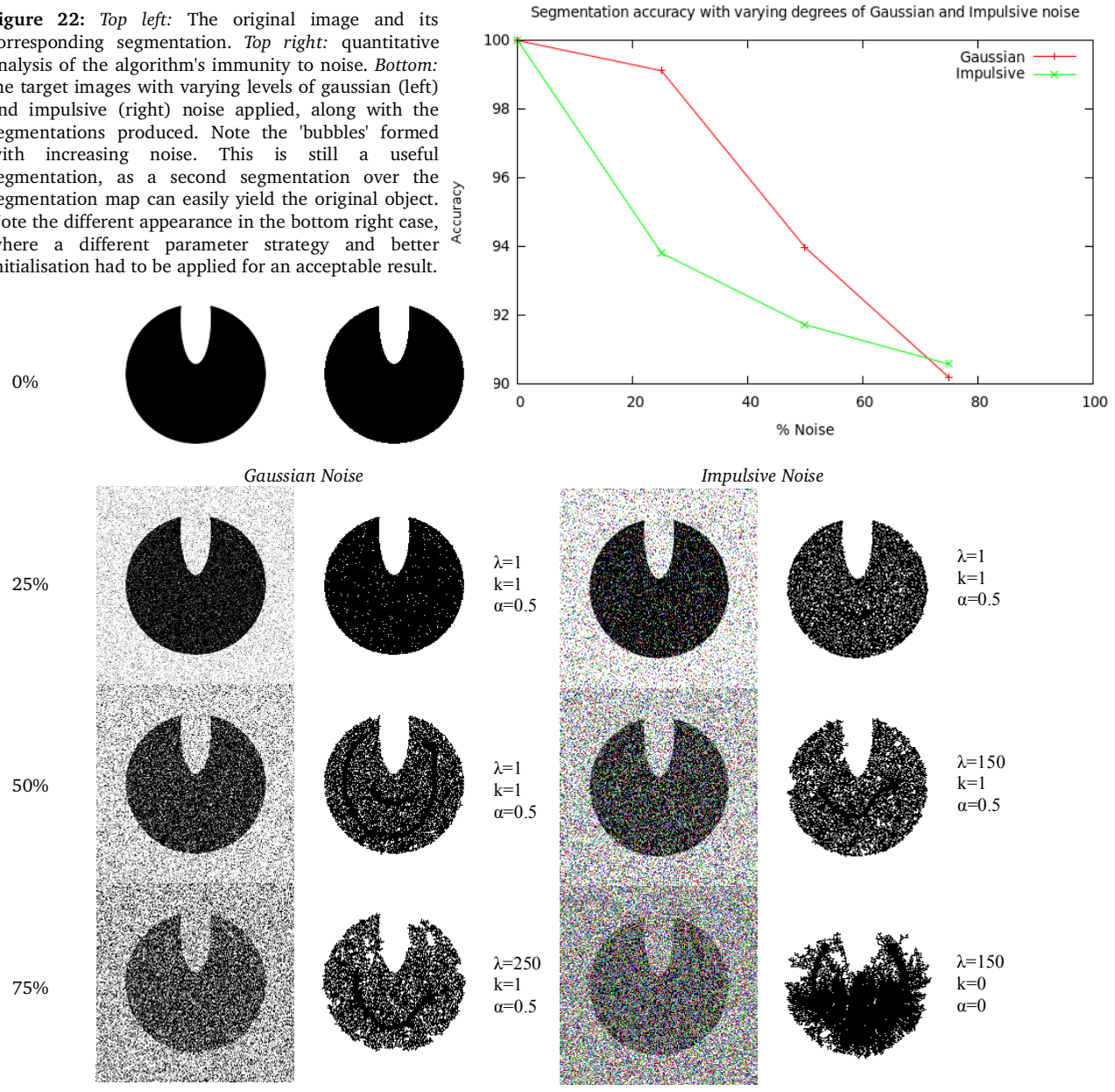


Figure 22: *Top left:* The original image and its corresponding segmentation. *Top right:* quantitative analysis of the algorithm's immunity to noise. *Bottom:* the target images with varying levels of gaussian (left) and impulsive (right) noise applied, along with the segmentations produced. Note the 'bubbles' formed with increasing noise. This is still a useful segmentation, as a second segmentation over the segmentation map can easily yield the original object. Note the different appearance in the bottom right case, where a different parameter strategy and better initialisation had to be applied for an acceptable result.

4. Improvements on the basic model.

The basic waterflow model has many positive features. It is simple and intuitive to use, as it is based on the physical principle of water flowing over an object. To a large extent, it is predictable in its output, and demonstrates good accuracy for simple and complex objects, and relative robustness to noise. It is flexible, as it makes use of both region-based statistics, and edge-based information, which enables it to be adapted to the particular case, making it a good tool for segmentation in a variety of image types.

However, there are a few points of concern. The most important one is efficiency. Liu and Nixon did not make any mention of the algorithm's efficiency in their papers, but this is clearly a very important omission, as segmentation of even simple objects can take several minutes with the basic model. The reason for this is that for each iteration, all contour pixels need to go through the entire image data twice – once for the acceleration phase, and once for the region-statistics step. The low efficiency becomes particularly evident in images that force the contour surface to become convoluted, such as grainy images, increasing the contour area exponentially.

Edge information can sometimes be unhelpful, unrefined or inappropriate. While the regional statistics element of the algorithm helps with this to an extent, appropriate initialisation with an adequate and informative region is required in pictures where the region-based force is the dominant component; and even then, leakage can occur through single noisy pixels. The model would benefit from the ability to manipulate the edge information, or specify other intuitive and dominant characteristics in the image, such as brightness or colour information, to appropriately affect the behaviour of the expanding water layer.

Another issue that we have touched on briefly in the previous section, is the rigidity of the forces involved. In a physical sense, this is equivalent to restricting the model to water, and its particular fixed ratio of pressure, surface tension and adhesion forces. It would be useful to vary these independently. In a physical sense, this would be like having the ability to choose between water and, say, an oil droplet, which shows far greater surface tension, and less adhesion for the same pressure, therefore allowing more flexibility in choosing a segmentation strategy.

In this section, we will discuss some optimisations, refinements and extensions implemented over the base algorithm to deal the above shortcomings.

4.1 Optimisations

As mentioned earlier, the largest obstacle to the framework's speed is the need to iterate over the entire image for each pixel in the expanding contour. This is done twice*, per contour pixel, for two different reasons. The first, is the need to calculate the gaussian force field acting on each contour pixel on the expanding water layer as accurately as possible. The second, is to obtain a representative statistic based on the characteristics of the segmented and unsegmented regions, to produce an appropriate region-based image force.

We would argue that the benefit gained in terms of a small increase in accuracy by including distant pixels in this calculation, on either of the two operations, is not justified by the exponential rise in computational cost. According to equation (1), it takes 100 pixels for pixels at just a 10-pixel distance, to cumulatively exert the same force as a single neighbouring pixel. Furthermore, from the

*Note that even though the two iterations could have been reduced to just one, the gain in speed in terms of computational complexity isn't very significant. Therefore, the two steps were kept separate for design clarity reasons.

point of view of emulating a physiological model, all three relevant forces (adhesion, pressure and surface tension) are extremely short range forces, therefore the Gaussian Force field is perhaps not the most efficient way to model them.

In terms of the regional-statistics force, again, if the object is fairly homogeneous (which is usually the case in distinct objects) then the region-based force should act correctly, even using a fraction of the image for statistical comparison.

A user-specified parameter W has therefore been introduced, taking values in the range 0-100%, specifying the percentage of the overall image size, which is to be used as the area of iteration around the contour pixels. This small change leads to an exponential reduction in speed.

Furthermore, W can be used as part of a segmentation strategy. For instance, if W is set to 0 (which corresponds to a 3×3 pixel area), pressure causing expansion is the dominant force, unless the contour is touching a boundary. This creates a contour that expands unimpeded by thick or irrelevant edges. Figure 23 shows an example of such a case; if the wider area was effective in terms of the Gaussian force field, then the water pressure wouldn't be enough to enable the contour to propagate through this 4-pixel wide channel. Furthermore, the water might have leaked out from the weak-edged passage ahead. By limiting the force-relevant area to a 3×3 area, the water can propagate unimpeded. This kind of strategy could prove useful in segmenting thin structures, like capillaries in an angiogram.

Figure 23: Evolving waterfront, as seen over the edge strength map (white = strong edge)

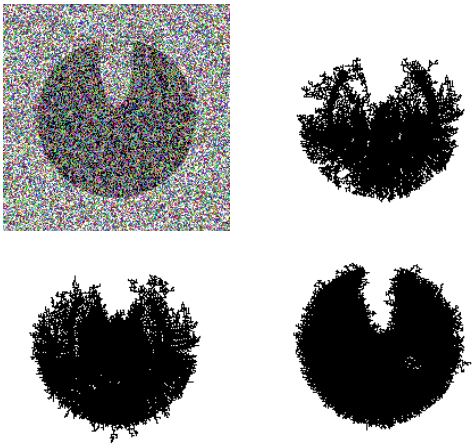


Figure 24: *Top left:* The 75% Impulsive noise infused image used in figure 22. *Top right:* The segmentation result from the basic algorithm (W=100%). *Bottom left:* Segmentation using W=50%. *Bottom right:* Segmentation using W=10%

It is difficult to talk about any generic effects on accuracy with lowering W, as the effect isn't necessarily always a negative one, as explained in the example above. Changing W doesn't simply *degrade* a segmentation, rather it *changes* it, in a manner dependent on the image type. Another example of this can be seen in figure 24, where the segmentation outcome actually improves by lowering W. An explanation for this might be that, by decreasing the area of the Gaussian force field, there is in fact less interference to be had on the expanding contour, from the noisy image.

In general, a relatively large value for W, such as >50%, usually produces results consistent with the basic model, while still conferring a significant reduction in complexity. Low values, such as 5-10%, have not been found to generally affect segmentation in a negative way. Rather than introducing two separate parameters for each of the two steps that require it, W was implemented as a single parameter applied equally to both the water forces and the image forces. This was done mainly for simplicity, as it avoids introducing more parameters to the user than necessary, while still achieving the intended optimisation effect.

4.2 Refinements

The refinements presented here are essentially changes to the basic framework that allow the user greater freedom in adopting appropriate segmentation strategies, through enhanced configurability of some of the framework's more rigid features, while not detracting from its intuitive nature.

4.2.1 Independence of water forces

The basic framework defines the relative weight for each of the three water forces (pressure, surface tension and adhesion), via a variable L as per equation (1), where $L=1$ if the pixel is a water element, and L is in the range '0 ↔1' corresponding to the edge strength otherwise. While the effect of the acquired velocity can be controlled as a whole through parameter λ, the proportion in which these three forces act in the basic framework is rigid. As discussed in the previous section, it is sometimes desirable to independently manipulate one of these three forces, in an analogy to changing the type of liquid used for segmentation.

A group of parameters was therefore introduced, to enable independent manipulation of the relative weight attributed to each of the three forces. Using the L notation to conform with the equations above, the parameters are L_e, L_c and L_w, corresponding to forces exerted from edge (i.e. adhesion), contour (i.e. surface tension) and inner water (i.e. pressure) pixels. The default values are kept the same as that of the basic framework for consistency.

The effect of independently increasing the surface tension has been demonstrated in the previous section (figures 20 & 21).

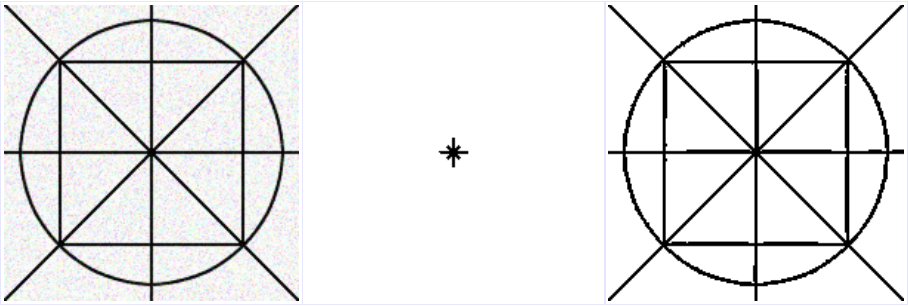


Figure 25: *Left:* The 200×200 version of the image in figure 18. *Middle:* Segmentation result obtained using the basic model.. *Right:* Segmentation result obtained by setting $L_w=10$

Figure 25 shows the an example where increasing the water pressure can be effective. Compare this to the results in figure 18. A small increase in water pressure allows the water layer to overcome edge artefacts and disruptive adhesion forces, and flow through extremely narrow passages, while still not being large enough to cause leakage through edge gaps. Note that the previous strategy described for flow through narrow passages (setting W to 0) would not be effective here, because of the consistent noise and edge artefacts, which creates 'blockages' within the passages.

Figure 26 shows an example using the adhesion force as the sole acting force, making it behave a bit more like a conventional edge-seeking snake. While this is a rather trivial example, it demonstrates the adaptability this refinement offers.

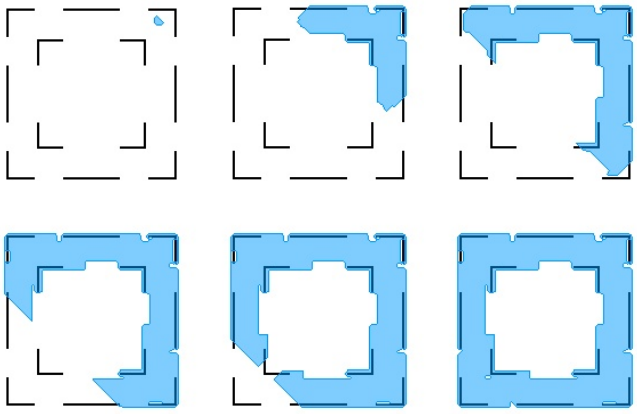


Figure 26: Segmentation using adhesion as the driving force: (Parameters used: $\lambda=1$, k=1, $\alpha=1$, $W=20\%$, $L_e=100$, $L_C=0$, $L_w=0$)

4.2.2 Image smoothing

Image smoothing involves averaging the values of pixels with those of their immediate neighbourhood. This produces an image where transitions at object boundaries become more graded, and lose their sharpness as a result. Smoothing is often applied by default in edge-detection algorithms, as a method of dampening down unwanted edges created by noise, and preserving a continuous edge over small artefactual gaps in the original image[34]. A default smoothing over a 5×5 pixel

neighbourhood has been applied by default in our implementation, as it was judged to produce the most appropriate amount of noise reduction, without significant loss in edge information. However, the quality of the edge information largely depends on the type of image. No smoothing on an image with clear boundaries and no noise, results in both accurately and precisely delineated object boundaries. In noisy images, increased smoothing results in a reduced effect on the edge strength map derived from noise, but at the expense of thicker edges, or in the case of the canny algorithm, opposing edges on either side of the boundary, with increasing distance between them and formation of gaps. On using the segmentation software, the ability to visualise the edge information becomes a very useful tool in predicting the behaviour of the expanding water layer. The ability to select a custom amount of smoothing therefore helps the user select the edge information that would be most appropriate for a particular segmentation strategy, as shown in figure 27. Our implementation allows a choice of smoothing over a 1×1 area (i.e. no smoothing) up to a maximum of 19×19, as further smoothing was not deemed to be useful in the majority of images.

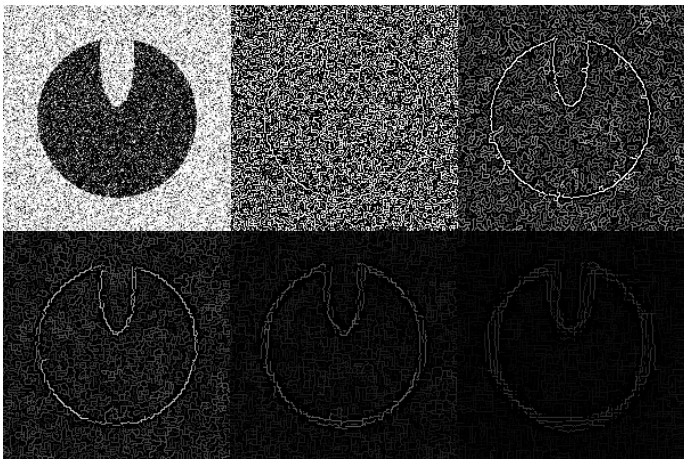


Figure 27:
Top Row: Left: Original image corrupted with 50% Gaussian noise. *Middle:* Canny edge with no smoothing (i.e. S=1×1). *Right:* Same with smoothing over a 3×3 area. The edge is now thin and more defined, but a lot of noisy edges still remain.
Bottom Row: Left: The default 5×5 smoothing used in the base algorithm. *Middle:* 9×9 smoothing. Few noisy edges remain, but the quality of the object edge starts deteriorating. *Right:* 15×15 smoothing. Note there is practically no noisy edges, but the object edge is now weaker, less precise and with many gaps present

4.3 Extensions to the basic framework

The extensions presented here are changes that either enhance the basic framework's segmentation ability in general, or provide tools that allow for easy segmentation strategies in particular image types where the basic algorithm would encounter difficulties.

4.3.1 Resistance

Resistance in the basic framework is implemented as a reduction in the expanding water contour's velocity, according to the edge strength it is found on. While this is useful in images with good boundary information, it is less useful in images with weak edges due to poorly defined boundaries, or inappropriate edges, such as in noisy images. We defined here 3 more classes of resistance, to help with these cases.

In the first type, resistance simply increases or decreases, according to how distant the brightness of the image pixel the contour finds itself on is, compared to a brightness specified by the user (by clicking on a target pixel on the image). This is useful if the object is in general darker or lighter (or *both*) compared to it's *immediate* surroundings, regardless of how well defined its boundary might be. Note that this strategy may succeed where variation in brightness might fail for the region-statistics method, if the transition is too gradual.

The second type, takes into account colour distance, rather than just brightness. Differentiating pixels by colour yields a lot more information than brightness, as several different colours can appear to have the same brightness. Colour distance is implemented as the sum of the absolute distance in each of the three channels (red, green, blue), compared to that of the target pixel. This is useful in segmenting objects that are fairly homogeneous in colour, the extent of which can be

determined by altering the resistance modifier appropriately.

The last type, alters the resistance experienced according to how different a pixel's hue is, compared to the target pixel. This is useful in segmenting objects which are relatively homogeneous in hue. Note here the difference between 'absolute colour' and hue. A blue object with a shadow cast on it, will still consist of the same blue hue throughout. However, in terms of absolute colour, the shadowed area has a different colour than the rest of the object (i.e. dark blue vs light blue).

In many cases where a brightness or colour based segmentation strategy may be useful as above, the existing image forces may not provide enough information to contain the contour, as the resistance doesn't reverse the velocity's direction, it only minimizes its value. In order to effectively terminate the algorithm in these cases, on the basis of the changes in resistance, a constant inhibitory force F_c was added. The idea is that, on areas where the resistance causes a significant drop in velocity, a small inhibitory force will contain the contour. The α parameter is therefore converted into an α and β pair, which controls the total image force acted against the velocity. Equation (12) is therefore modified as follows:

$$F_T = \alpha F_{edge} + \beta(F_{region}) + (1-\alpha-\beta)F_{const}$$

Figure 28 summarizes and demonstrates the effect of the different types of Resistance.

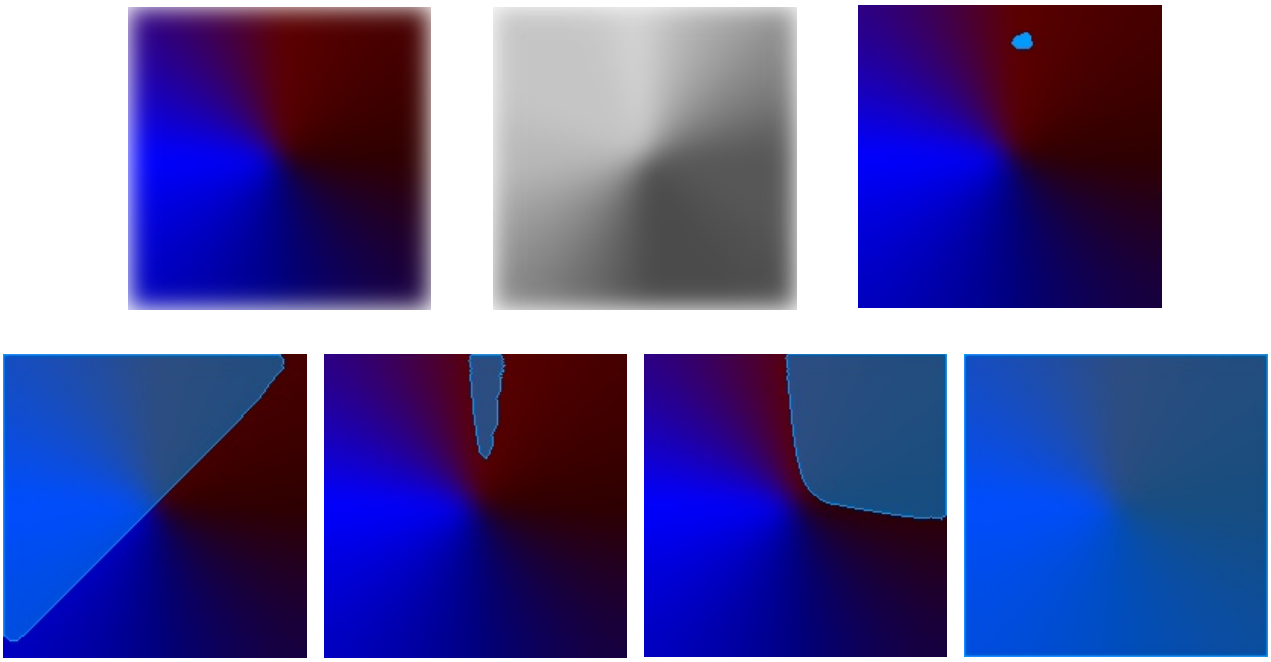


Figure 28:
Top Row from left to right: a) Image constructed such that the colours rotate through hue and brightness. At the extremes, the top right corner is of a red hue, and the bottom left corner is of a blue hue, both at the same brightness level. b) The greyscale equivalent of the image, demonstrating that the two colours at the bottom left and top right corner have the same brightness. c) the target pixel chosen and also initialisation used for all the segmentations in the next row. Parameters used were: λ=100, k=50, α=0, β=0, Γ=100
Bottom Row from left to right: a) Resistance increases away from target pixel's brightness. b) Resistance increases away from target pixel's colour. c) Resistance increases away from target pixel's hue. d) Basic framework resistance to edges. Since this image has smooth transitions throughout, the edge strength map is effectively blank. Regional forces prove ineffective as well, as the transition is smooth enough, to gradually shift the segmented region's statistics toward the unsegmented area, as the contour expands.

4.3.2 Edges in colour images

As has been pointed out in the previous subsection, in the case of colour images, there is a lot more information that can be obtained from a pixel than just it's brightness. Traditional edge-detection

techniques, including the canny algorithm, derive their edge information from the way pixel intensity varies across the image (i.e. by detecting areas of large intensity gradient). This in effect means that the information utilized in the image is reduced to that of its greyscale equivalent. This doesn't always yield the desired outcome, however, as in colour images, different colours can have similar intensities, therefore producing no change in the intensity gradient, at the boundaries between such colours.

An edge-detection algorithm was therefore designed to take full advantage of the concept of absolute colour differences described in the previous subsection. For each pixel in the image, the intensity differences were calculated between the horizontal and vertical neighbouring pixels, for each of the colour channels in the image. Then instead of adding the obtained values (which would lead to a result similar to an intensity gradient map) the absolute values are added up. This means that, if two channels changed in opposite directions, leading to a similar intensity, by adding the absolute values, the absolute deviation per colour can be quantified instead. Once all the pixels in the image have an 'absolute colour difference' value associated with them in the vertical and horizontal directions, the two are combined by keeping the highest value if the two differ. The maximum absolute colour difference found is used to then scale all the values in the range $0 \leftrightarrow 255$, which are then used as the intensity of the newly created edge strength map. The result can be seen in figures 29 and 30. Note that when applied to a greyscale image, the edge strength map produced is essentially the same as an intensity gradient map.

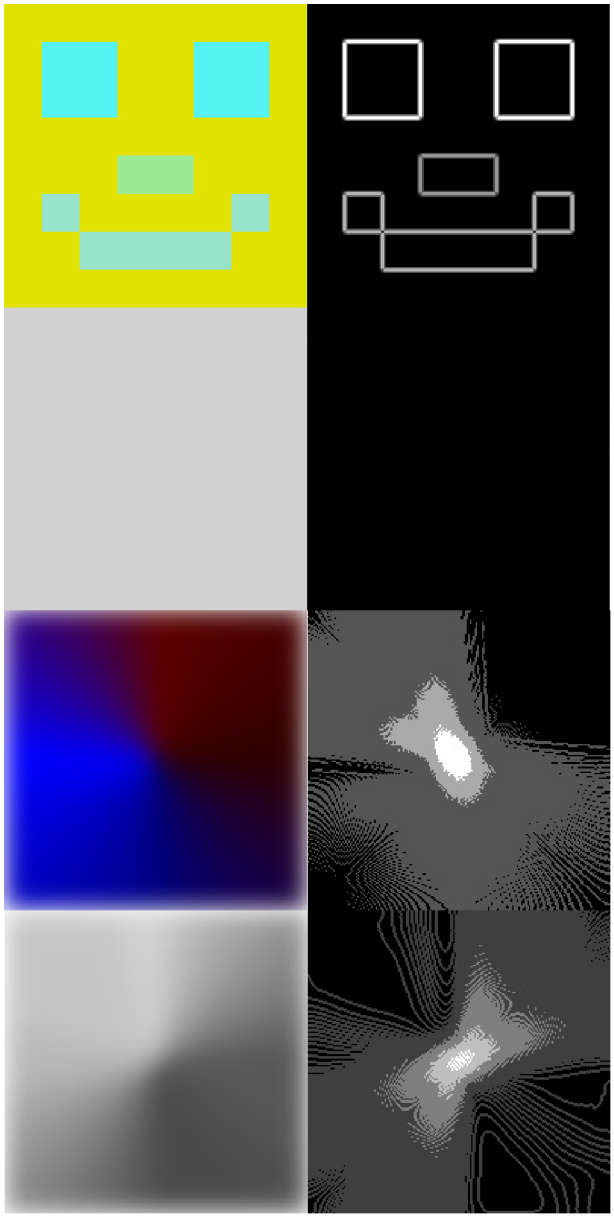


Figure 29: *Top Row:* Example of an image featuring colours of equal intensity, and its absolute colour difference based edge strength map. Segmentation using this edge map is trivially easy. *Second Row:* The greyscale image produced from the previous picture. The information previously provided by colour is lost . As is expected, any edge strength map based on this image is empty. *Third Row:* The colour image from figure 28 and its associated absolute colour difference edge strength map. *Fourth Row:* The greyscale equivalent of the previous picture and its edge map. Note the difference in the two edge map results.

4.3.3 Edge thresholding

Edge thresholding is a simple yet effective technique to further eliminate inappropriate edge information. The assumption is that important edges will be stronger than unimportant edges. If this is the case then edges below a certain threshold can be removed from the edge map completely, leaving only the most appropriate edges behind to be used for the segmentation. Noise can be easily eliminated in this manner, without the need for excessive smoothing. Since the edge map can be visualised directly in the segmentation software, thresholding provides a quick way to manipulate and potentially improve the edge map before segmentation.

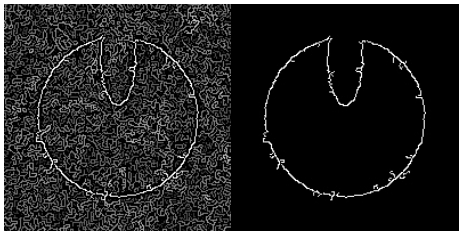


Figure 30: *Left:* The edgemap from figure 27. *Right:* Same, thresholded at a strength of 200.

4.3.4 Edge connection

As discussed earlier, our brain often 'fills in the gaps' when information is missing, to obtain a meaningful object it can relate to by experience. A special form of this, is the ability to infer object boundaries, known as illusory contours[35], where edge information is only partially present in the original image. Kanizsa's triangle and Kanizsa's square are probably the most famous examples of this type of image (figure 31). Successful segmentation of such simple illusory contour diagrams has been demonstrated using specialised Level-Set snakes[36]. An attempt was made to provide this functionality to the waterflow framework as well, by implementing a 'loose' edge connection algorithm, where a 'loose' edge is defined as either a single detached edge point, an edge point at the end of a line of edge points, or one that forms the tip of a sharp corner in the edge map. The algorithm simply tries to join up these loose edges in optimal pairs sharing compatible features such as position and orientation (i.e. more compatible if facing each other), distance (compatibility decreases with distance), and intervening edges (two loose edges will not be connected if the line path crosses another edge). The algorithm is basic and lacks advanced 'edge directional continuity' features such as those implemented in the canny edge-detection algorithm, and as such is not very robust in non-trivial images where edge information isn't always meaningful, and therefore its use in medical images is probably limited. It works reasonably well in diagrammatic images, however, so it has been kept as a 'proof of concept' feature in the software implementation. Figure 31 shows an example of the algorithm at work.

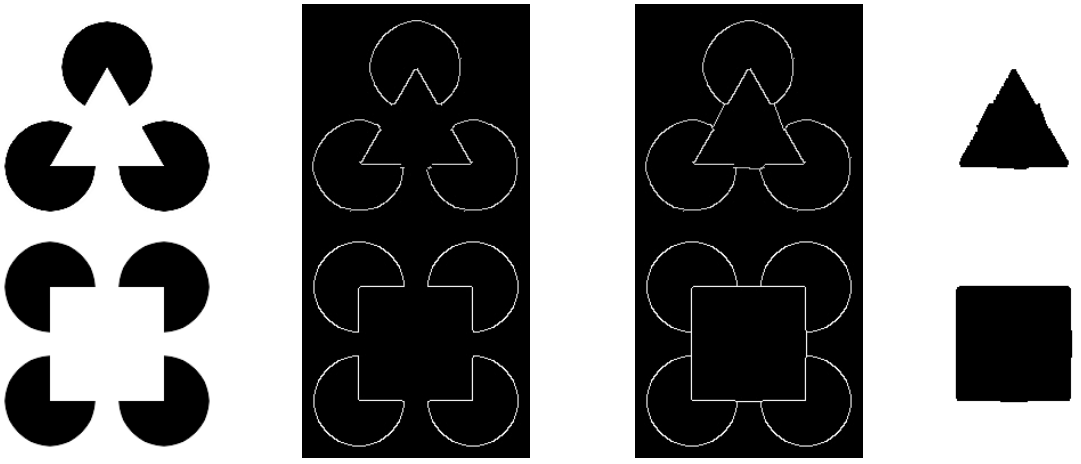


Figure 31: Illusory contour segmentation using an edge connection approach. *From left to right:* a) The illusory contour (kanizsa traingle and kanizsa square). b) The resulting edge strength map. c) The edge strength map after loose edge connection. d) The obtained segmentation result.

4.3.5 Modified behaviour of water forces

In certain cases, it may be beneficial to modify the behaviour of the water forces, to benefit from particular image features. For instance, objects with blurred boundaries, appear to have weak, thick edges on the intensity gradient map; these edges are easily overcome by the expanding water layer. Modifying the behaviour of flooded edges to retain their adhesive effect, prevents the water layer from 'escaping' such thick weak edges, enabling the algorithm to segment blurry objects more effectively. However, despite a locally improved result, caution is needed as it can lead to unpredictable results in the overall segmentation, making its use less intuitive and subject to experimentation. Figures 32 & 33 show an example of a case where preserved adhesion might be useful.

Figure 32: An example of an object giving a weak thick edge

Figure 33: Effect of retaining flooded edges' adhesion effect. (Note: the region-based force is off to demonstrate the effect.)
Top Row: Progression of the evolving contour as per the basic framework, as seen over the edge strength map of the image in figure 32.
Bottom Row: Progression of the evolving contour in the same image, with flooded edges preserving their adhesion effect.

5. Experimental Results

In order to objectively assess the performance of both the basic framework and the enhanced model, a series of comparisons are made with the three models described earlier; the MAC, Watershed and SIOX models. Firstly we compare the performance of all five models against the ground truth, using synthetic images, designed to elicit strengths and weaknesses, or involving objects typically known to pose particular problems in many segmentation algorithms. Secondly, we qualitatively compare segmentation results obtained on medical images across the modalities described in the introductory section.

The MAC model software used was obtained from the authors of the original paper (available online at the University of Bristol website's 'Vision' section[*]), and is developed in java. The Watershed implementation used was the open source 'SegmentIt' tool by Bruno Klava (available online at sourceforge[**]) also developed in java. The SIOX implementation used is the one integrated in the open source GNU Image Manipulation Program (The GIMP[***]). The Ground truth was obtained either from direct conversion of the object into a binary map if appropriate, or via manual delineation of the objects of interest.

The Watershed and SIOX tools do not provide details on how the segmentation evolves over time in the same way the Waterflow model's water layer and MAC snake evolution can be traced from initialisation to final segmentation result. Therefore only initialisations and final segmentation results will be shown, except in cases were the evolution was deemed not to be straightforward, or a particular or non-intuitive strategy had to be used (such as if multiple steps were needed for an optimal result), or if tracing the contour provides specific insight into the algorithm's operation.

Performance for each test is analysed in terms of a confusion matrix and efficiency. Comments on any particular algorithm's strengths, weaknesses or usability are made where relevant. The parameters used for each segmentation are only explicitly shown where different from the default parameters in each model, shown in Table 1.

* http://www.cs.bris.ac.uk/Research/Vision/AC/MAC/index.html
** http://watershed.sourceforge.net/
***http://www.gimp.org

Model		Parameter name	Function	Default Value
Magnetostatic Active Contour (MAC) model		Gaussian Sigma	Controls image smoothing	1
		Current Orientation	Controls direction of electrical current in the model	0
		Magnetic Diffuse K	Alters nature of magnetic field	0
		Curvature Force	Controls snake smoothness	1
		Gradient Threshold	Eliminates very small edge gradients	0
Watershed Transformation (WST) model		Connectivity	Choose available number of directions for pixel propagation [options: 4 and 8]	4
Simple Interactive Object Extraction (SIOX) model		Smoothing	Amount of image smoothing as a preprocessing step	3
		Colour sensitivity – brightness component (L)	Determines the L component axis cluster size	64
		Colour sensitivity – reg/green component (a)	Determines the a component axis cluster size	128
		Colour sensitivity – yellow/blue component (b)	Determines the b component axis cluster size	256
Waterflow model	Basic (WFB)	Velocity modifier λ	Linearly increases the effect of velocity against the image forces	1
		Resistance Modifier k	Exponentially increases the effect of resistance	1
		Image force balance constants α and β	Controls how the Total Image Force is calculated, where $F_T = \alpha F_{edge} + \beta F_{region} + (1 - \alpha - \beta)\,\Gamma$ ($\Gamma = 0$ in the basic model)	0.5
	Extended (WFE)	Constant External force Γ	A constant force always acting in the opposite direction to the expanding contour	0
		Actively processed area W	Percentage of the overall image used in velocity and region statistics calculations	100.00%
		Water force weight ratio L	Ratio of adhesion : pressure : surface tension.	01:01:01
		Resistance type R	Options: Resistance to Edges or increasing / decreasing Intensity, Colour or Hue	Resistance to Edge
		Smoothing area S	Area over which smoothing is applied on obtaining the edge strength map	5×5
		Edge strength map technique	Options: Map based on Canny or Absolute Colour Gradients	Canny
		Edge thresholding below θ	Threshold below which weaker edges will be discarded from the edge strength map	0
		Edge connection	Attempt to artificially connect 'loose' edges	FALSE
		Water force modification	Preserve adhesive effect of flooded edges	FALSE

Table 1: Default parameters for the comparative study segmentation models

Note that the different algorithms produce their segmentation results in somewhat different formats, and these are re-interpreted as a binary map containing only positive and negative pixels, for accuracy measurements against the ground truth. Efficiency is determined as time taken to obtain the segmentation (preprocessing operations were not taken into account). All software was run on a laptop, running ubuntu linux on an AMD Turion™ 64 Mobile processor.

5.1 Synthetic images

5.1.1 Four disc problem

This problem involves segmentation of multiple objects arranged in a symmetrical manner around a central point. As demonstrated by Xie and Mirmehdi in [18], even in a simple image like this, this type of arrangement causes problems with some snakes, as the central part of the image can lead to the formation of 'saddle points' (i.e. points where the evolving contour receives forces that cancel each other out, and hence stops evolving). External and internal initialisations are demonstrated where applicable, demonstrating the different strategies that can be followed.

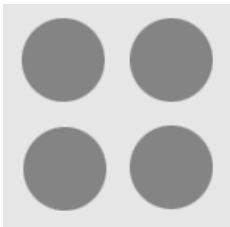


Figure 34: The four-disc problem, adapted from Xie and Mirmehdi[18]

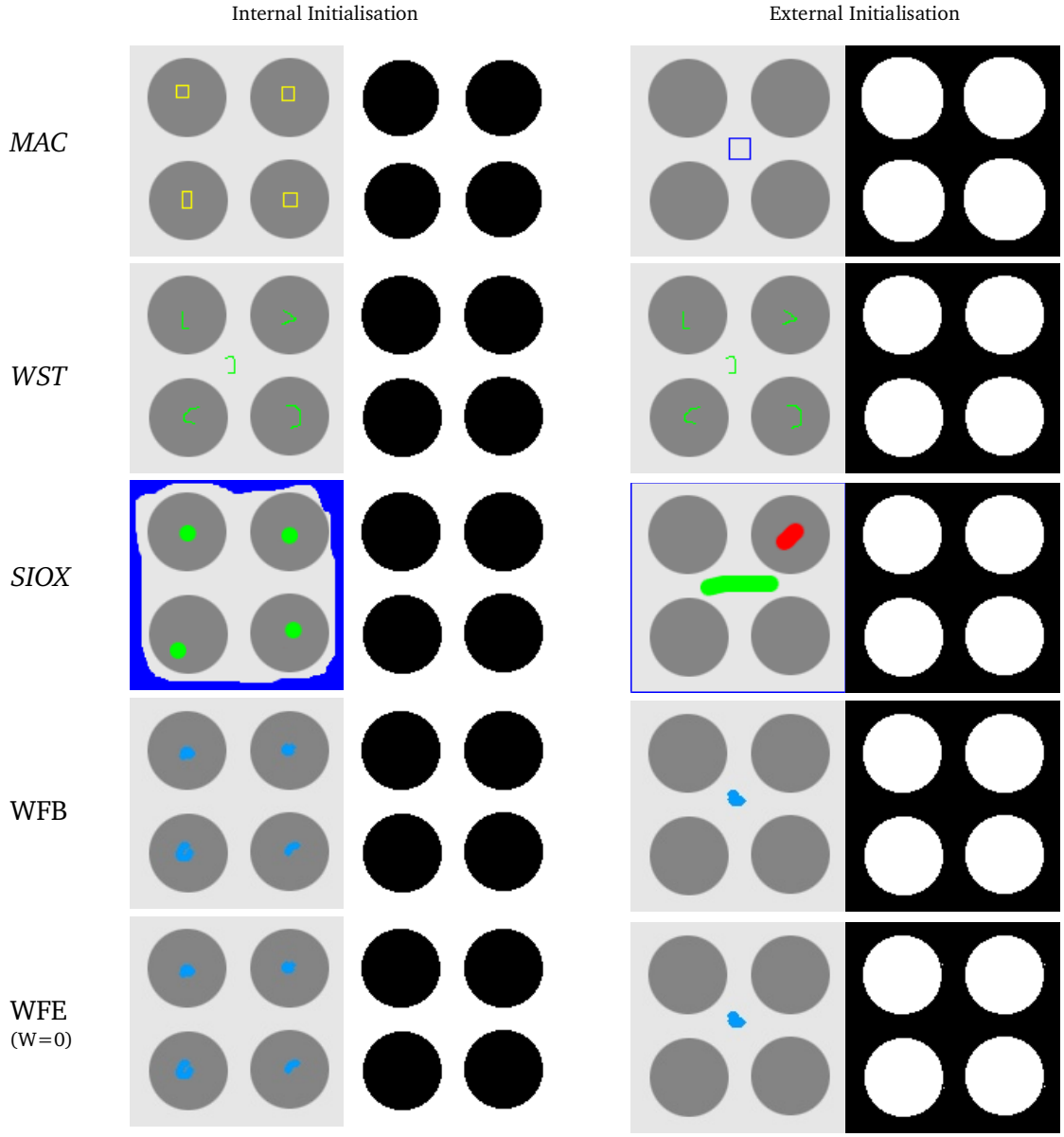


Figure 35: The four disc problem. Internal and External initialisations, with their corresponding segmentations.
Notes: The MAC snakes expand both inwards and outwards from their initialisations. The WST segments into 5 regions with the single initialisation, and the user simply chooses the regions of interest. In the SIOX row, the blue area corresponds to the area initialised as definite background surrounding the object of interest, the green markers are markers denoting definite foreground, and red markers denote extra areas of definite background.

All algorithms achieve good performance on the four-disc problem. Note the difference in efficiency between the basic waterflow model and its extended version. Without the extensions, the basic framework is the slowest algorithm by far (followed by the MAC, due to its use of Level Sets). The extensions make it comparable in speed to the WST and SIOX models.

	Sensitivity	Specificity	Precision	Accuracy	Efficiency
	Internal Initialisation				
MAC	92.91%	100.00%	100.00%	96.95%	9.4s
WST	99.31%	99.62%	99.49%	99.49%	0.6s
SIOX	99.41%	99.56%	99.41%	99.49%	0.7s
WFB	99.41%	99.56%	99.41%	99.49%	17.2s
WFE	99.38%	99.56%	99.42%	99.49%	0.5s
	External Initialisation				
MAC	93.01%	100.00%	100.00%	96.02%	28.6s
WST	99.61%	99.31%	99.48%	99.49%	0.6s
SIOX	99.56%	99.41%	99.56%	99.49%	0.6s
WFB	99.58%	99.39%	99.54%	99.50%	1m 29.6s
WFE	99.57%	99.32%	99.49%	99.47%	1.3s

Table 2: Four disc problem (internal and external approaches)

5.1.2 The Rorschach problem

The image portrayed in figure 36 resembling a Rorschach inkblot test, was designed to test several issues that often cause problems in segmentation algorithms. The image is rich in thin lines, acute concavities (both internal and external) and features a generally complex geometry with a symmetrical component.

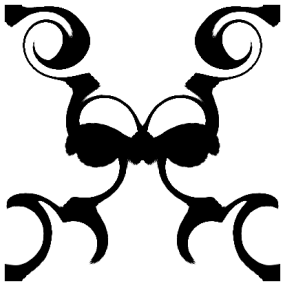


Figure 36: A Rorschach Inkblot-like image

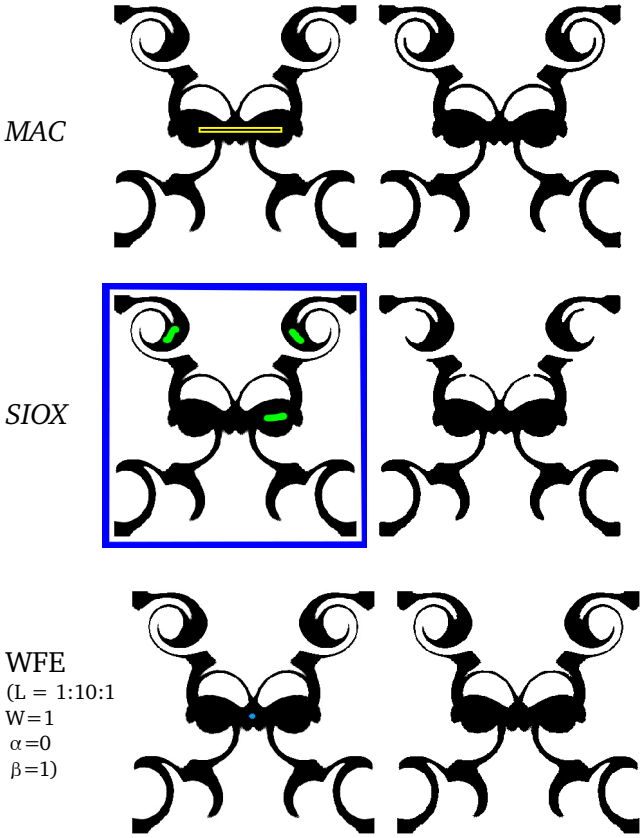


Figure 37: The 'Rorschah' problem. Note the extra initialisations needed by the WST and SIOX models for effective segmentation, the missing lines in these two models, and the thicker lines in the MAC model (which explains the lower accuracy found numerically)

	Sensitivity	Specificity	Precision	Accuracy	Efficiency
MAC	99.71%	99.00%	96.25%	99.14%	14m 13s
WST	97.70%	99.78%	99.13%	99.35%	1.2s
SIOX	96.94%	99.89%	99.55%	99.28%	3.0s
WFB	99.86%	99.89%	99.58%	99.88%	4h 22m
WFE	99.73%	99.91%	99.66%	99.87%	15.6s

Table 3: The Rorschach Problem

The basic waterflow model achieves the best results in this particular problem, however this comes at the unacceptable time of over 4 hours due to the increased surface area associated with the shape complexity. This would clearly make the basic framework unusable in real-life situations. The extended model required slight experimentation with the 'water pressure' and 'image forces', but achieved a comparable result in only a few seconds. The results obtained for the WST and SIOX models where also the best possible in a number of parameters and initialisations tested. Note that while quantitatively the MAC model appears to be the least accurate, this is due to a thicker segmentation of the thin lines in the image, and in fact qualitatively it is better than the WST and SIOX models, which are missing important parts from the end result. In general, accurate segmentation in terms of object thickness may be an issue if the smoothing used during edge-detection causes the edges to thicken, a problem which the waterflow model can potentially be susceptible to as well. In general, less smoothing leads to more accurate boundary detection, at the expense, however, of less noise reduction, which clearly in this type of crisp image isn't an issue.

5.1.3 Broken edges

Figure 38 shows an object which poses certain challenges to segmentation. Firstly, the edge of the object contains small gaps. Most snakes will pass through these gaps, rather than treat the broken edge as a continuous boundary. Naturally, depending on the situation, this may be the desired result, but if one attempts to segment the delineated object instead, it is usually very difficult to control the algorithms so as to display enough elasticity to prevent the snake from escaping at those particular points, but allow it to evolve everywhere else. The same should be the case for the waterflow model. The other issue is that the main body of the object also happens to be of the same intensity as its background, which makes statistical methods also unlikely to segment the object correctly. The authors of the MAC paper showed successful segmentation of a similar shape in their paper, but we were unable to reproduce the effect on this particular image (perhaps the gaps in their image were smaller). We successfully segment this object with the extended waterflow model, by using the edge connection feature, as well as some edge manipulation (intensity gradient map used on an unsmoothed image – canny algorithm did not produce the correct connection result) and preserved adhesion of flooded edges to prevent water from escaping from the thin newly-formed connected edges.

Figure 38: Object delineated using a dashed line.

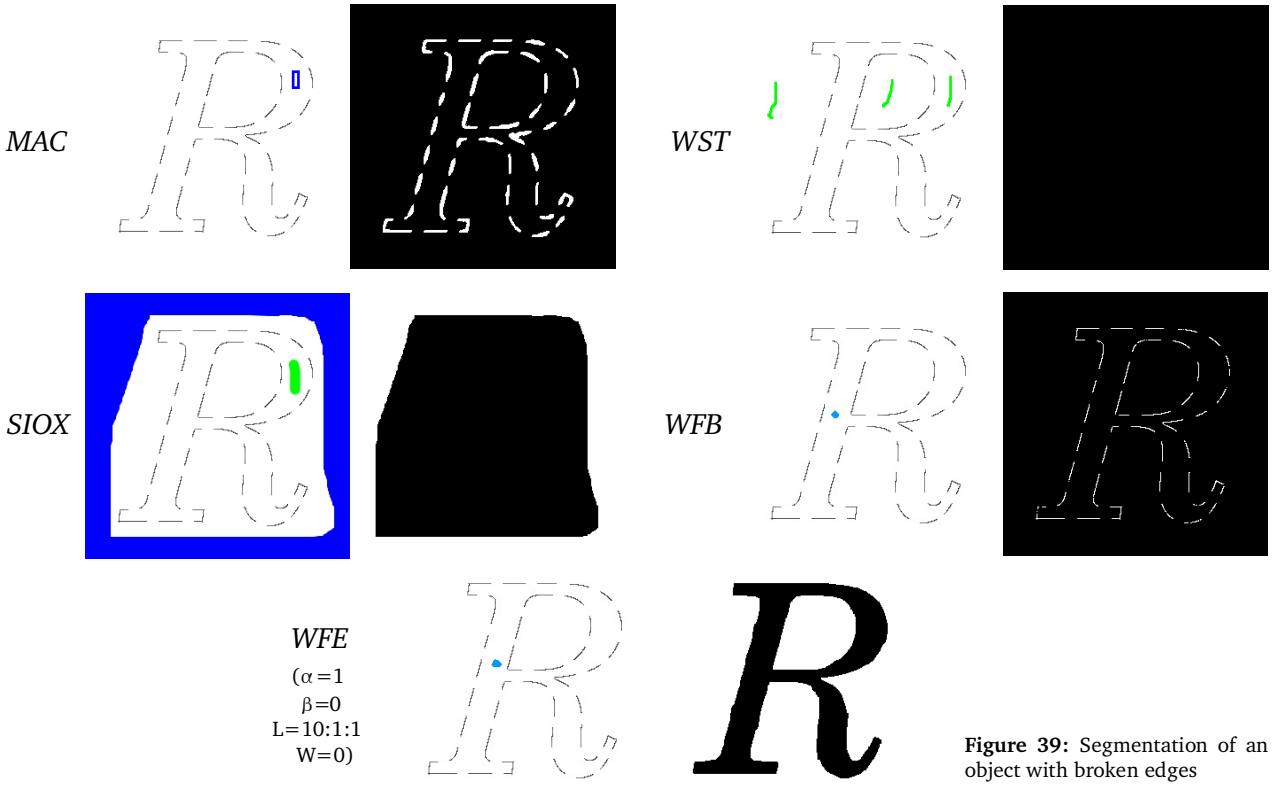


Figure 39: Segmentation of an object with broken edges

As expected, the SIOX model and Watershed models cannot deal with this type of segmentation. The MAC model could not be made to segment the walls of the object successfully despite several parameter combinations tried. Note again the increased thickness produced by the MAC model compared to the waterflow model.

	Sensitivity	Specificity	Precision	Accuracy	Efficiency
MAC	87.53%	2.11%	18.11%	19.05%	6m 33s
WST	100.00%	0.00%	19.82%	19.82%	0.6s
SIOX	100.00%	45.35%	31.15%	56.18%	1.0s
WFB	95.51%	0.40%	19.16%	19.25%	1h 2m
WFE	97.98%	99.65%	98.56%	99.32%	2.7s

Table 4: Object with broken edges.

5.1.4 Weak Edges

Objects with graded or blurred boundaries such as the object in figure 40 can cause problems in a number of segmentation algorithms for two reasons. First, edge information fades to nothing as the graded boundary is approached, so there is no edge to contain the evolving snake or watershed. Secondly, the graded end approaches a colour similar to its background, which means statistical methods will stop segmenting well before the actual implied boundary, and rather at a point where the pixel statistics are at the threshold between that of the rest of the object and that of the background.

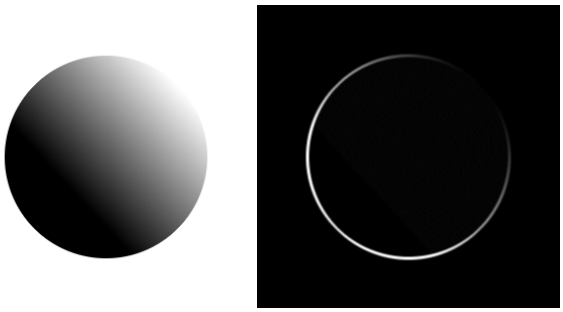


Figure 40: Object with a weak edge and its corresponding intensity gradient map.

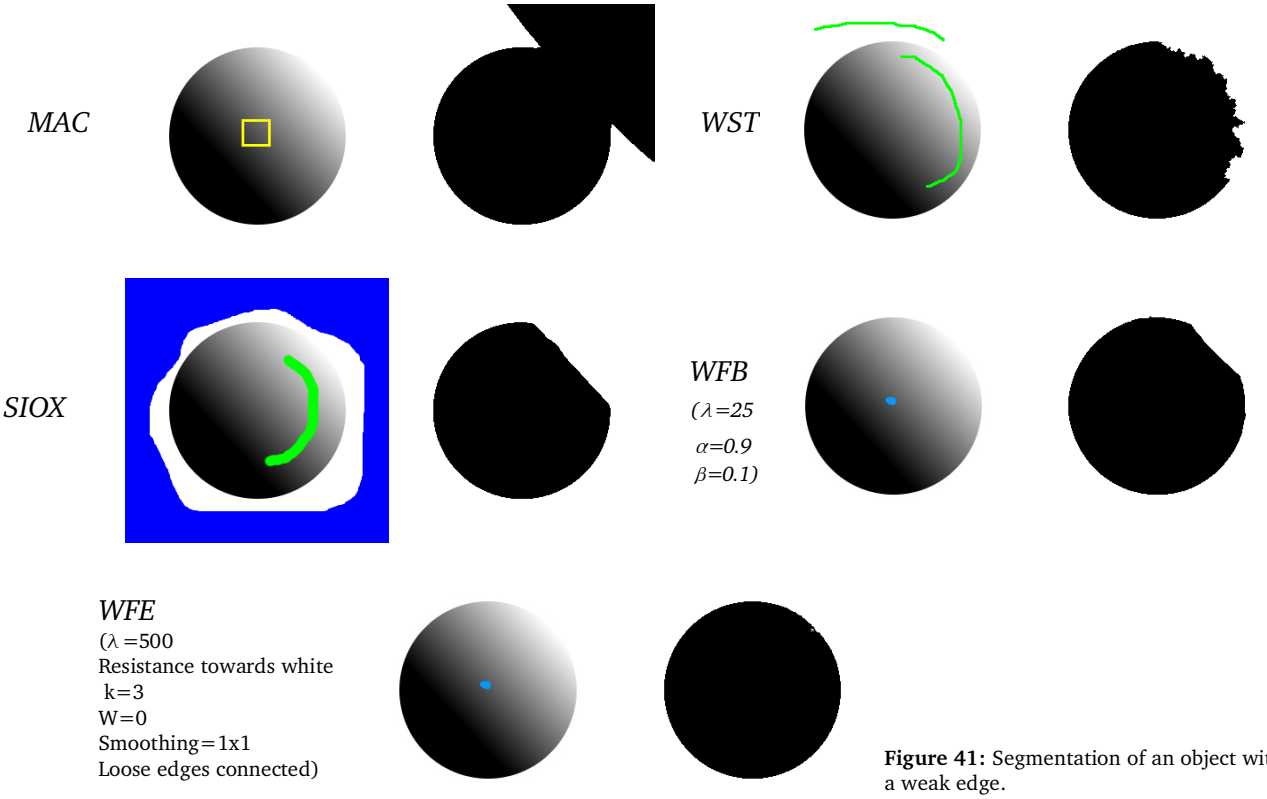


Figure 41: Segmentation of an object with a weak edge.

All models were tried on a variety of their allowable parameters to obtain the best result possible. The MAC model has been previously shown[18] to handle weak edges reasonably, however we could not replicate this on this particular object, and the snake escaped through the weak edge. The watershed model provides a segmentation which is only as good as its initial markers, i.e. closer to the weak boundary and covering a reasonable surface along the sphere. The SIOX model, as expected, stops at a point where the intensity is halfway between the object's average intensity and the background white. A reasonable result was reached with the basic model after some experimentation, by finding a velocity modifier which was enough to overcome some change in regional statistics, but not enough to escape over the weak edge. The extended model achieves the best result, by making use of resistance to pixel intensity and loose edge connection.

	Sensitivity	Specificity	Precision	Accuracy	Efficiency
MAC	99.97%	74.31%	67.67%	83.28%	2m 20s
WST	95.28%	100.00%	100.00%	98.35%	0.8s
SIOX	93.67%	99.92%	99.83%	97.73%	1.0s
WFB	97.87%	99.84%	99.69%	99.15%	3m 33s
WFE	99.82%	99.97%	99.95%	99.92%	2.1s

Table 5: Object with weak edges

5.1.5 Noisy images

One of the most important features in any segmentation algorithm is the ability to handle different types of noise. Figure 43 compares the 5 models using the University of Bristol Computer Science departmental logo shown in figure 42, corrupted with Gaussian and Impulsive noise at the 25% and 75% levels. Initialisations were the same throughout all noise levels per model, as shown over the original images on the left.

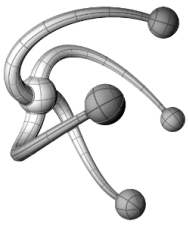


Figure 42: University of Bristol Computer Science Logo

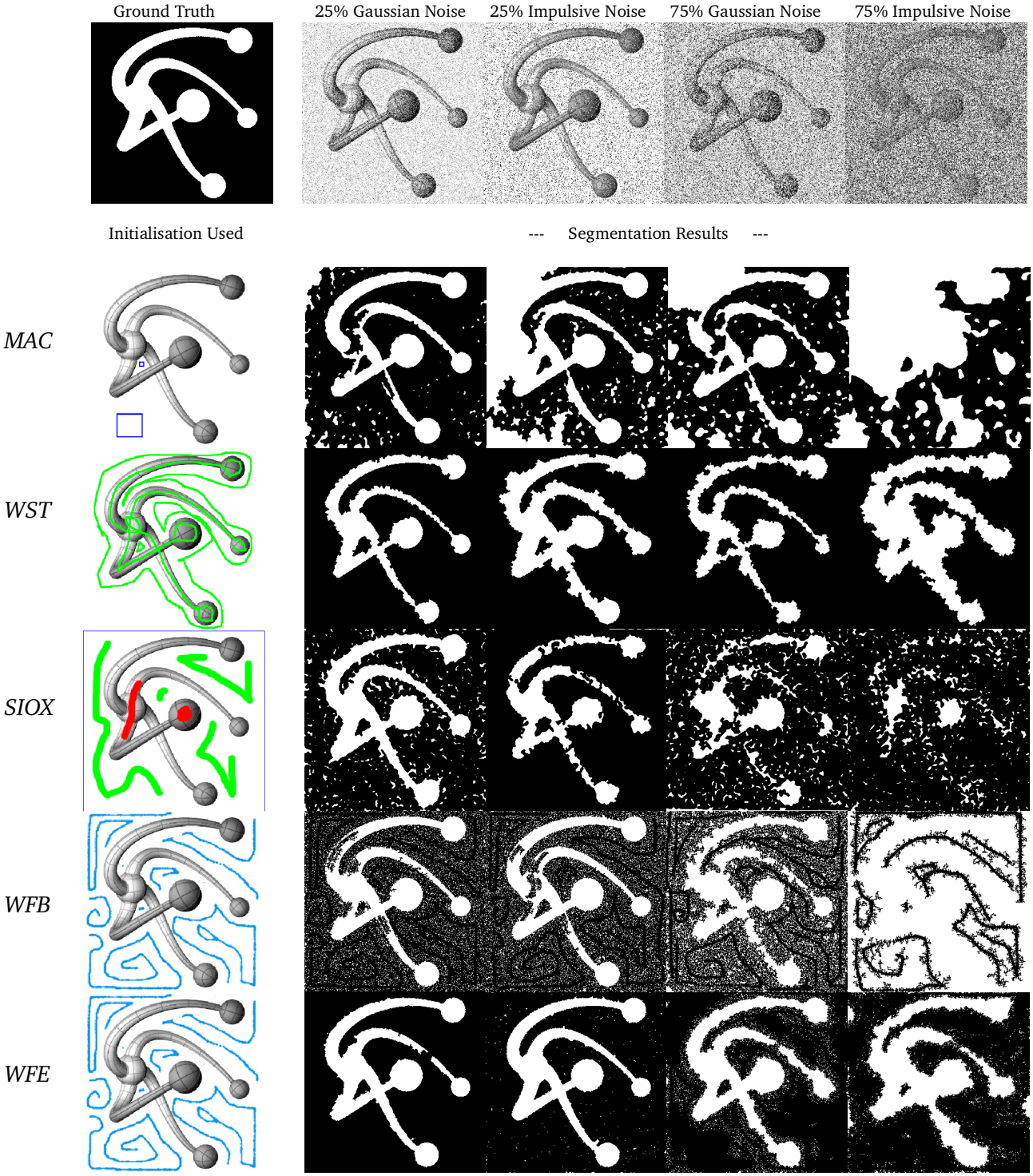


Figure 43: Segmentation of noisy images. Parameters used: *MAC* – Σ=1.5, θ=0.05. *WFE* – 25%(G): λ=1000, k=1000, α=1,β=0,W=1,Smooth=3x3,θ=100 + edge connection and preserved adhesion = True. 25%(I): Resistance to white, k=1000, L=100:1:10000,W=5,Smooth=9x9, ACG edge used. 75%(G): λ=75, k=0, α=0, β=1, L=1:1:10, W=10, θ=255. 75%(I): λ=5000, k=6, α=0, β=1, L=1:1:10, θ=255, W=10, Resistance to middle grey used. WFB 75%(I) failed to segment on a range of parameters.

	Sensitivity (%)				Specificity (%)				Precision (%)				Accuracy (%)				Efficiency (s)			
	25% (G)	25% (I)	75% (G)	75% (I)	25% (G)	25% (I)	75% (G)	75% (I)	25% (G)	25% (I)	75% (G)	75% (I)	25% (G)	25% (I)	75% (G)	75% (I)	25% (G)	25% (I)	75% (G)	75% (I)
MAC	93.8	70.7	78.9	57.5	87.7	93.7	85.7	95.1	96.3	97.5	94.9	97.5	92.4	75.9	80.4	66	N/A*	N/A	N/A	N/A
WST	100	93.8	98.5	86.8	84.2	97.1	76	93.8	95.6	99.1	93.3	97.9	96.4	94.6	93.4	88.4	<1s	<1s	<1s	<1s
SIOX	91.9	99.6	91.5	95.6	97.9	78.6	65.7	33.2	99.3	94.1	90.1	83	93.2	94.8	85.6	81.4	<1s	<1s	<1s	<1s
WFB	87.9	84.9	74.1	29.7	89.4	88.9	90.4	99.7	96.6	96.3	96.3	99.7	88.2	85.8	77.8	45.6	3h	5h	3h	N/A
WFE	99.8	99	92.1	87.5	93.8	94.8	95.6	93.6	98.2	98.5	98.6	97.9	98.4	98	92.9	88.9	4.1s	43.3s	1m20s	46.8s

Table 6: Segmentation of Noisy Images. *MAC model had to be aborted manually as snake would not terminate.

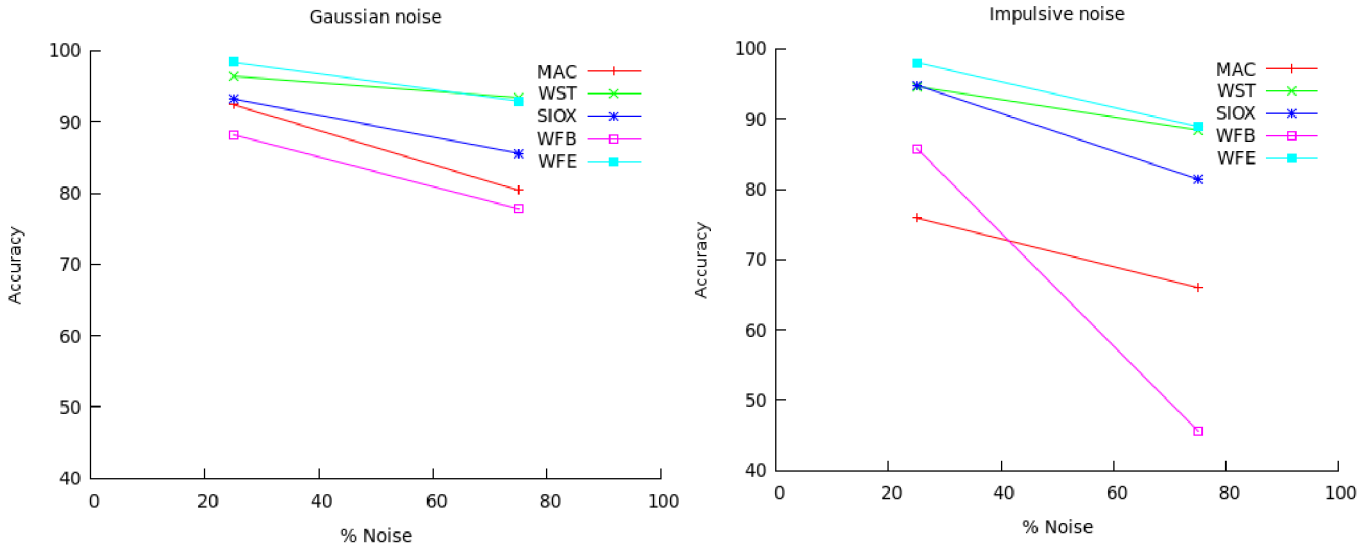


Figure 44: Graphical results of the Gaussian and Impulsive noise tests

Rich initialisations had to be used with all models in order to obtain good results (except the MAC model which didn't produce a better result with richer initialisation). Note that the WFE model permitted a different segmentation strategy to be used at each level, allowing an optimal result. It is important to keep in mind that, in the cases where the WFB model achieves lower accuracy due to the presence of 'bubble' artefacts, in qualitative terms these are not as destructive, as they would be relatively easy to discard using simple post-processing techniques, therefore the degree to which the segmented boundaries are faithful to the object is more important.

5.1.6 Colour Images

Figure 45 shows what is known as an Ishihara plate, commonly used to test colour blindness, as the two colour shades used correspond roughly to the same intensity, therefore a colour blind person cannot accurately rely on intensity alone to distinguish between the two regions, and must use colour receptors instead. We would expect a similar difficulty encountered from the algorithms tested here, with the exception of the SIOX model, since they are optimised to detect changes in intensity, rather than changes in colour.

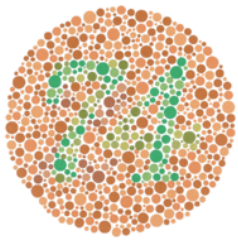


Figure 45: Ishihara test No 74

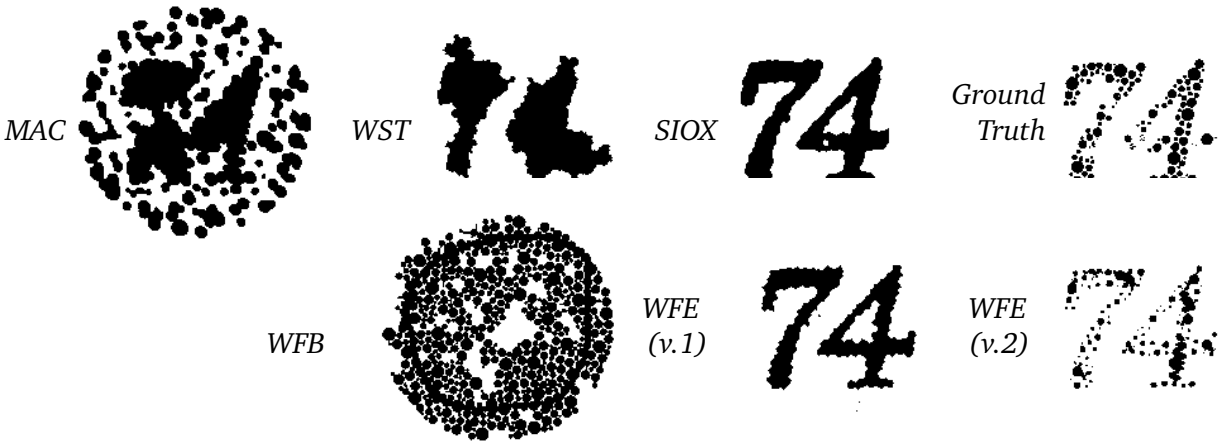


Figure 46: Ishihara plate segmentation results. Parameters used for WFE:
Version 1: λ=1000, Resistance to red, k=100, α=0, β=0, Γ=10, W=0, θ=255.
Version 2: λ=9000, Resistance to green, k=3, α=0, β=0, Γ=10, W=0, θ=255.
Note in the WF and MAC models the segmentation is an inverted 'external' one.

As expected, the SIOX model and the extended Waterflow model making use of colour resistance produce more meaningful results. The Waterflow model, however, obtained a result even closer to the actual ground truth via an alternate strategy, whereas the SIOX could not. Note the diameter of the circles in the second version is thinner by 1 pixel throughout the segmentation, as the resistance only kicks in 'after' it has encountered a pixel. Since this is a consistent 'error', it is a trivial to remedy post-processing, by 'growing' the selection outwards by 1 pixel to obtain a more accurate result (this can be done easily using the GIMP suite).

	Sensitivity (%)	Specificity (%)	Precision (%)	Accuracy (%)	Efficiency
MAC	99.7	78.3	17.8	79.3	6m
WST	86.9	90.8	30.7	90.6	<1s
SIOX	99.7	95	48.2	95.2	<1s
WFB	93.2	74.4	14.6	75.2	2h
WFE (v1)	97.8	95.9	53.1	96	3s
WFE (v2)	55.9	99.3	80	97.4	3s

Table 7: Segmentation of colour images with uniform intensity

5.2 Medical images

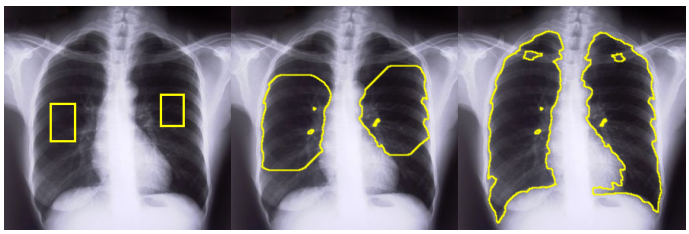
In this section, the extended model is applied onto images of a medical nature, alongside the MAC, Watershed and SIOX models. Due to the ambiguous nature of establishing the ground truth in medical images (in fact this is one of the main uses for segmentation algorithms) results are only discussed qualitatively.

5.2.1 Chest Radiographs

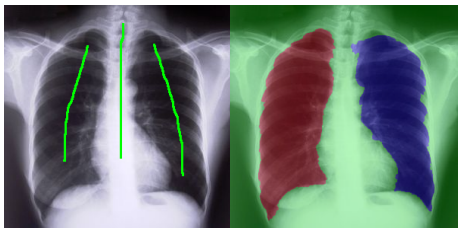
Figure 47: Normal Plain Chest Radiograph in a female patient

Figure 47 shows a normal chest radiograph. The area covered by the lungs in the plain chest radiograph correlates well with the lung's functional residual capacity[37], therefore segmentation of the lung fields is a good way to monitor lung health, and enable numerical measurements to be made, which are easier to observe for trends than just visualising the radiograph itself. Segmentation was obtained using a strategy based on a combination of resistance to intensity and the regional statistics force, which adhered to the heart, ribcage and

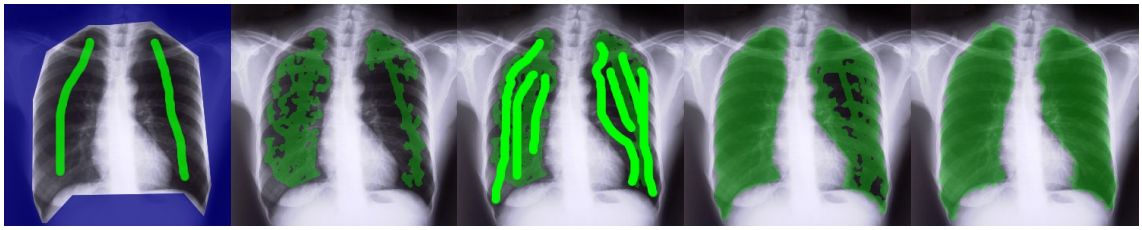
diaphragmatic boundaries well. Initial residual holes and 'bubble' artefacts were eliminated by overriding them with more flowpoints, and resuming segmentation, as shown in figure 48. The SIOX model performed similarly, and exhibited the same behaviour, needing several 'filling' strokes before the final result, but produced a consistent segmentation, although slightly underestimating the heart size, possibly due to the slightly graded edge. The Watershed tool was very straightforward and produced a good result, but with slightly more jagged edges. The MAC model was also very straightforward to use and achieved a good result except for a small number of holes and some jagged edges (but less so than the Watershed model). The hole artefacts were eliminated as a post-processing step (using a simple flood fill over the segmentation map), as the software does not provide the ability to combine segmentations, or resume a snake following further initialisation, like the other models do (in this case the whole boundary was detected so it was not a problem, but this is a major drawback if the boundary is only partially segmented, as there is no easy way to extend it). The MAC and Waterflow models were also the only models that correctly segmented the narrow area between the heart and the diaphragm. In this example, the Waterflow model segmentation is clinically the most accurate one in terms of area and relevant structures delineated (the aortic knuckle is very clearly defined compared to the other models), and the tool was felt to be the most flexible one to use (the SIOX model, was felt to be less flexible, as it required more 'filling' strokes and responded less predictably to them in comparison).



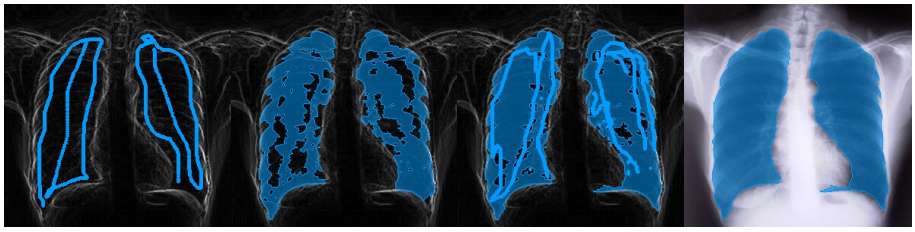
MAC model: initialisation, propagation and final curve



WST: initial markers and resulting catchment basins



SIOX model: Initialisation, first result, further initialisation, example of halfway-through segmentation, and eventual outcome.



Waterflow model: Initialisation (as seen over the edge map), initial ourcome, further initialisation and eventual outcome.

Segmentation maps: MAC Watershed SIOX Waterflow

Figure 48: Plain chest radiograph segmentation. Note the smoother shapes, and well defined structures, (such as the small round notch above the heart, known as the aortic knuckle, where the aorta arches over and behind the heart) in the Waterflow segmentation, compared to the other models.

5.2.2 Digital subtraction angiography

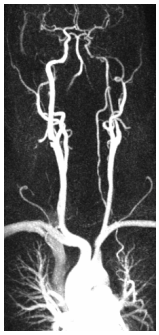
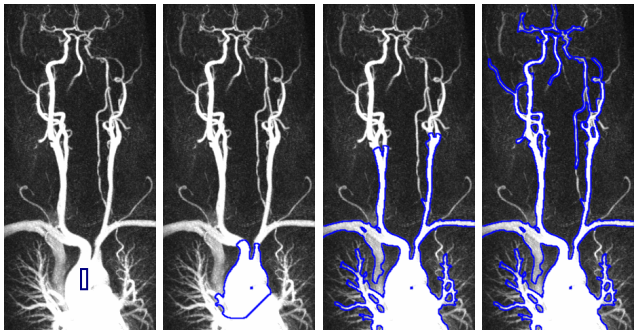
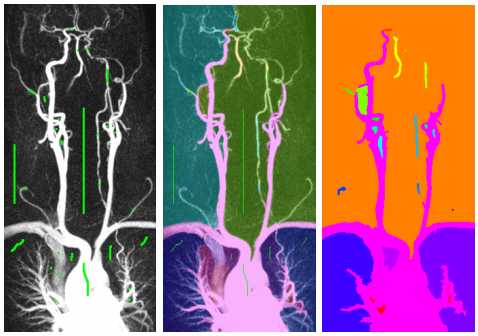


Figure 49: Digital Subtraction Angiography of the carotid tree

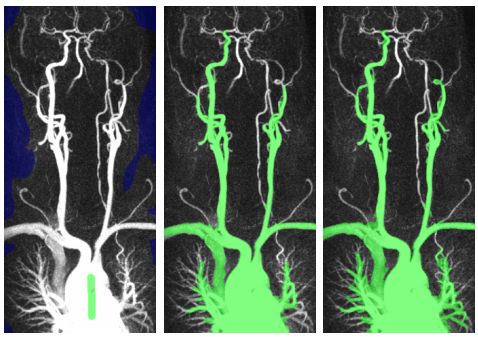
Digital subtraction angiography is an imaging technique used to visualise blood vessels. It involves digitally subtracting the intensity values of two radiographic films: A film obtained during injection into the bloodstream of a contrast medium such as a radio-opaque dye, and a normal pre-contrast film. The theoretical result should be an isolated arteriovenous tree; in practice though, motion artifacts and noise are common, but usually the arteriovenous tree can be clearly visualised in the resulting image[38]. One of the diagnoses sought through an angiogram is areas where vessels might have potentially become narrower, impeding normal blood flow. Good segmentation therefore entails the ability to accurately detect such thin structures.



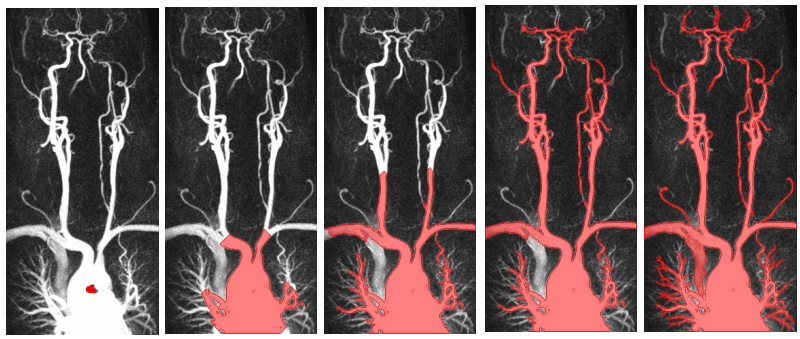
MAC: initialisation, evolution of the MAC snak, and final curve



WST: Initial markers, Pre-welding & final catchment basins



SIOX: Initialisation and result pre and post-corrections



WF: Initialisation, evolution & final result post secondary initialisations at final frame

Results: *MAC* *Watershed* *SIOX* *Waterflow*

Figure 50: Segmentation of a digital subtraction angiography image. The Waterflow model achieves the most complete tree, followed by the MAC model. Note the increased vessel thickness represented by the MAC. Thickness is mostly accurate in the Waterflow model as the segmentation results from a combination of edge (at no smoothing) and regional statistics forces, rather than purely resistance to black.

5.2.3.1 Abdominal CT

CT is an excellent modality to visualise organs of the abdominal cavity, as they have good variation in terms of Xray absorption profiles, and can be differentiated radiologically relatively easily. We test our models in segmenting a pair of kidneys. Such segmentation could be used to monitor the development of hydronephrosis (i.e. kidney distension due to obstruction), or simply segment the kidney at various levels to enable a 3D model to be made.

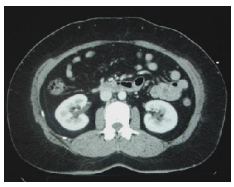


Figure 51: Abdominal CT scan at kidney level

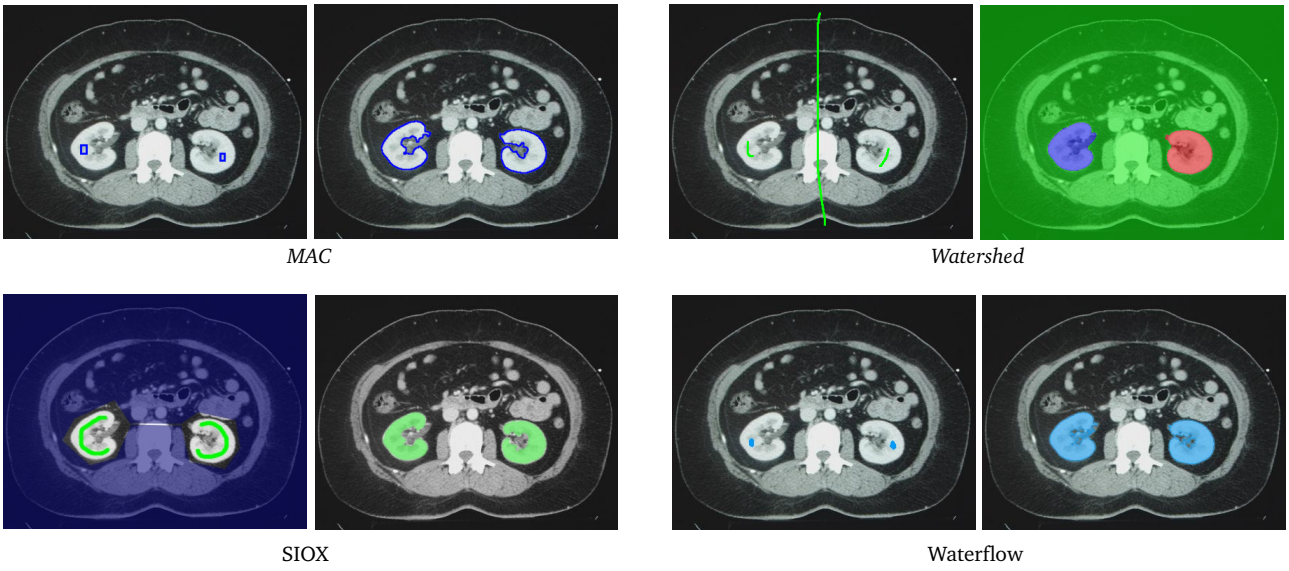


Figure 52: Segmentation of kidneys on abdominal CT. The waterflow model accurately segments the renal pelvis (i.e. the stalk entering the kidney), as well as the calyces (the main body of the kidney)

The basic model's parameters were used for this segmentation, as the kidneys in this context have both good edge information, and significantly different intensity profile to their immediate surroundings. Furthermore, it is a small and non-convoluted area, so the basic model is reasonably fast in this case. Note that the waterflow model is the only one that accurately detected the stalk. (if we wanted to exclude it, higher resistance could have been used, for instance). Accurate segmentation of the stalk was not possible with the other models in this context.

5.2.3.2 Head CT

While not as well defined as MRI, CT of the brain can be very useful in detecting specific pathology. Increased pressure of the cerebrospinal fluid in the brain, for instance, can cause enlargement of the ventricles, which is a fluid filled space at the centre of the brain. Segmentation could be used to monitor the ventricles at specific brain levels.

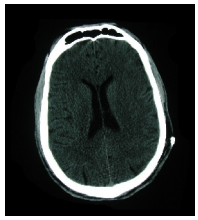


Figure 53: Head CT

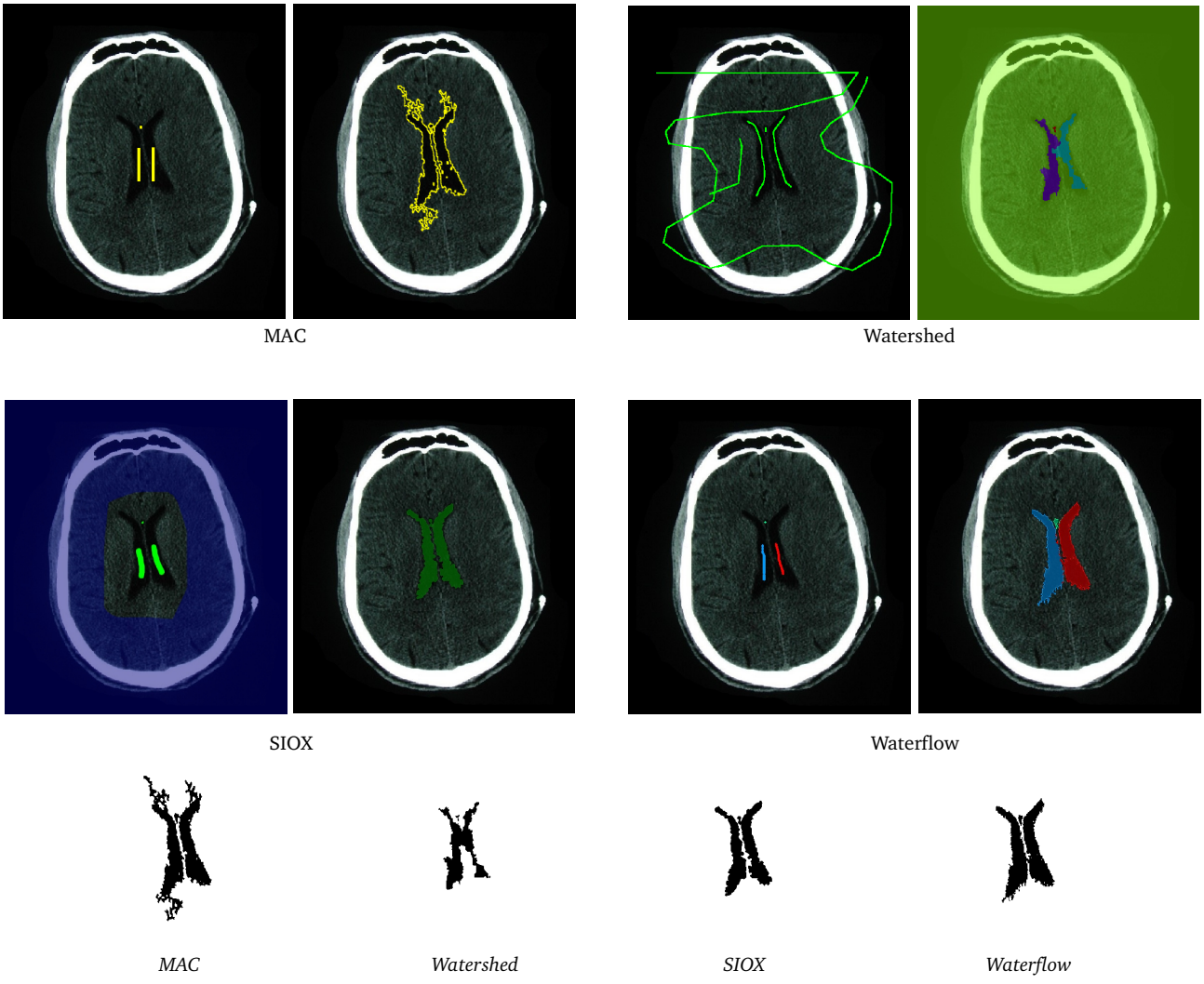


Figure 54: Segmentation of the 3rd and Lateral ventricels on a CT Head.

The waterflow model produced the most accurate segmentation, including the small third ventricle (which looks like a small cavity located above and between the two C-shaped lateral ventricles), colour-coded green in the result image. The segmentation strategy used involved increasing resistance away from a darker intensity, and more emphasis on the region-statistics image force (although still at a low actively processed area, at W=5%) compared to the edge force.

5.2.4.1 MRI Brain

Segmentation of the brain substance itself is best carried out on MRI images. Figure 55 shows a T1-weighted MRI Brain (T1 denoting the MRI settings favour anatomical differentiation rather than identifying pathology, due to areas rich in fluid appearing darker). The innermost white matter of the brain is easily differentiated visually on this modality from the outermost grey matter. Accurate segmentation of white matter can be useful, as white matter loss or imbalance can correlate well with certain pathological processes.

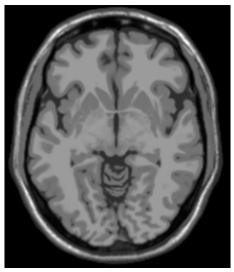


Figure 55: MRI Brain

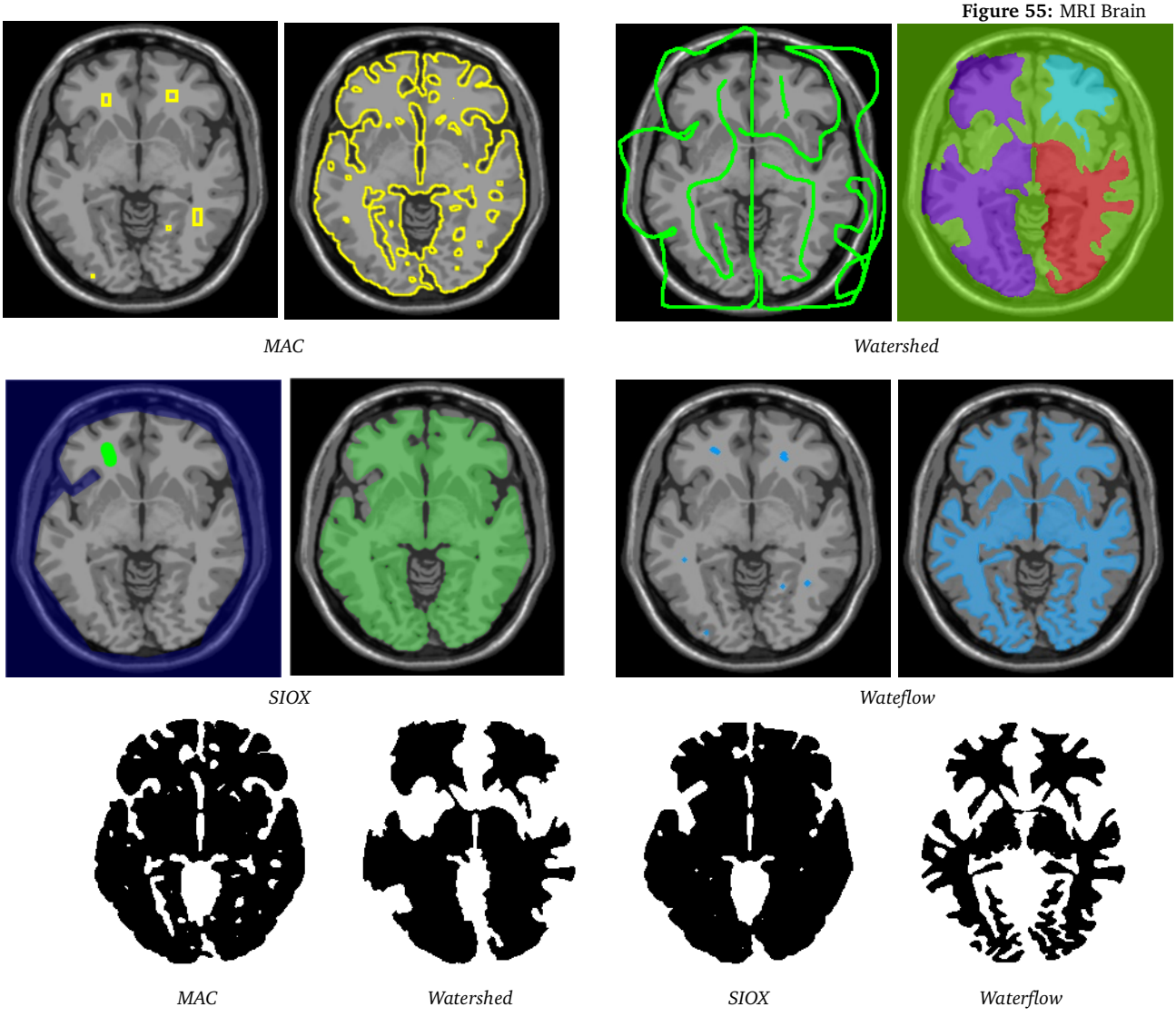


Figure 56: Segmentation of White Matter on MRI Brain

The waterflow model is the only model achieving correct segmentation of white matter only in this test. Since the intensity of white matter in this image is quite uniform throughout, the segmentation strategy used involved resistance increasing away from the intensity corresponding to that of a target white matter pixel, along with a rather high actively processed area at W=25%, for a better regional-statistics force effect.

5.2.4.2 MRI Knee-joint

While bone is generally better viewed with Xray-based modalities when looking for fractures, its relation to surrounding structures, such as muscles and ligaments, or signs of underlying infection are best determined by MRI. Figure 57 shows an MRI sagittal section at the level of the knee-joint. The spaces between the structures shown can be indicative of pathology, such as osteoarthritis (wearing of the articular cartilages between the bones), dislocations, ligamentous ruptures, effusions etc. Accurate segmentation could therefore help study these conditions more efficiently and effectively.

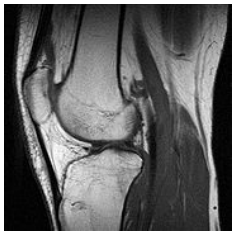


Figure 57: MRI of the knee-joint

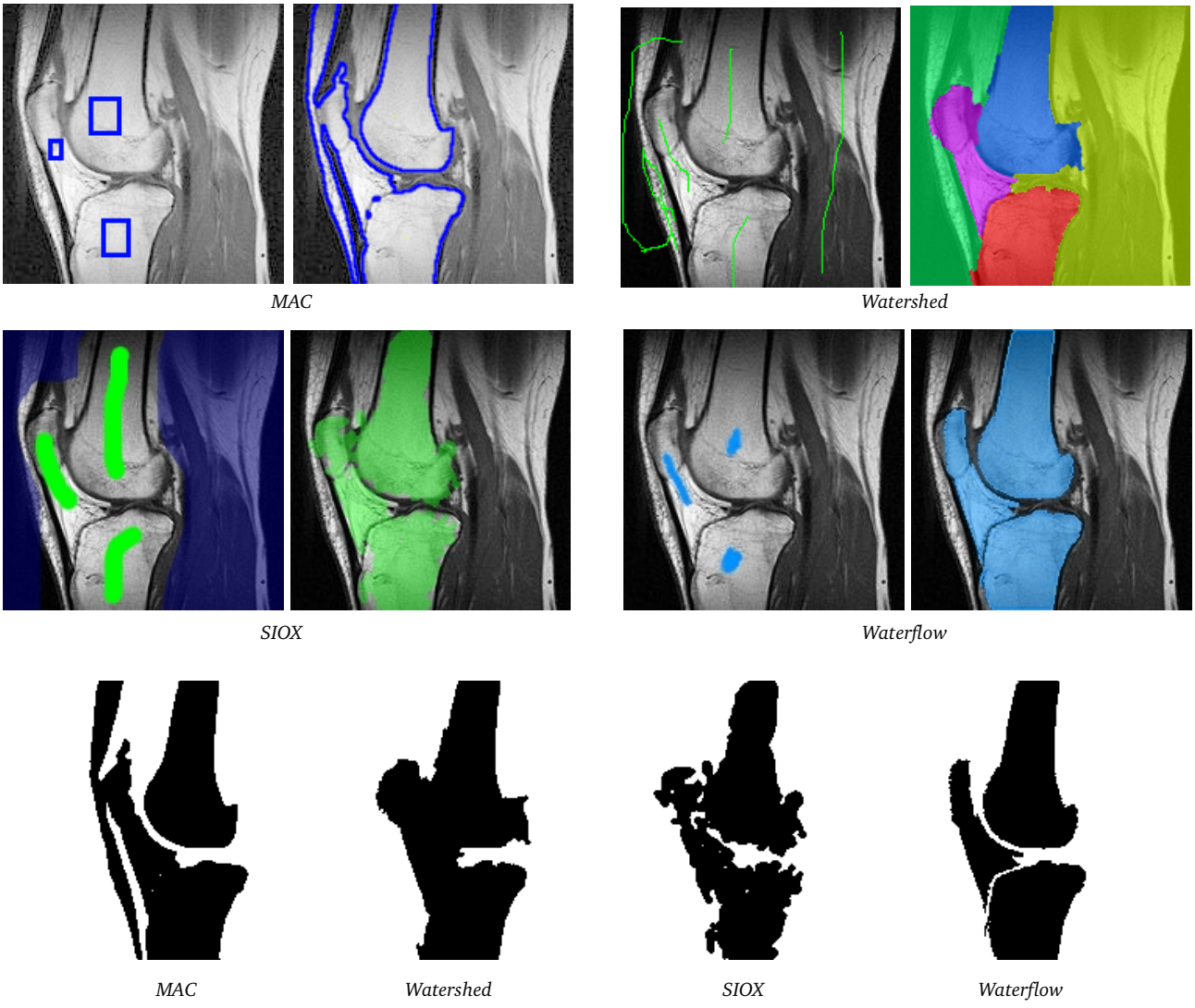


Figure 58: Segmentation of the femur, tibia, patella and patellar tendon at the knee joint.

The waterflow model achieves the clearest segmentation, allowing accurate measurements for bone diameter and articular width to be taken. Segmentation strategy was consistent for all three structures, as large velocity and resistance modifiers, with resistance increasing towards darker pixels. Note, however, that for images with multiple objects, different segmentation strategies can be used for each object separately, by freezing each consecutive segmentation. Given that the user controls the progression of the water layer, one could even freeze a water layer half-way through its evolution, to take advantage of positional features, and combine two half-segmentations arriving from areas with different features, and meeting halfway, thus producing a full segmentation.

5.2.5 Nuclear Imaging

Nuclear imaging commonly results in images that are of particularly low resolution and quality. It is often not easy to distinguish clear boundaries or focal areas of increased or decreased activity, unless the underlying pathological processes are reasonably established, and the difference is quite pronounced in comparison to a normal scan. As such, assessment of such modalities can rely on the 'feel' the clinician gets on interpreting it, rather than accurate measurements. Effective and consistent segmentation could therefore prove useful in providing clinicians with more concrete representations, as well as numerical data, which can then be used to evaluate trends and establish treatment goals.

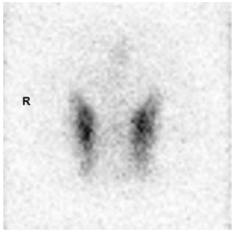


Figure 59: Thyroid scan showing increased uptake of radioactive Iodine in a patient with hyperthyroidism

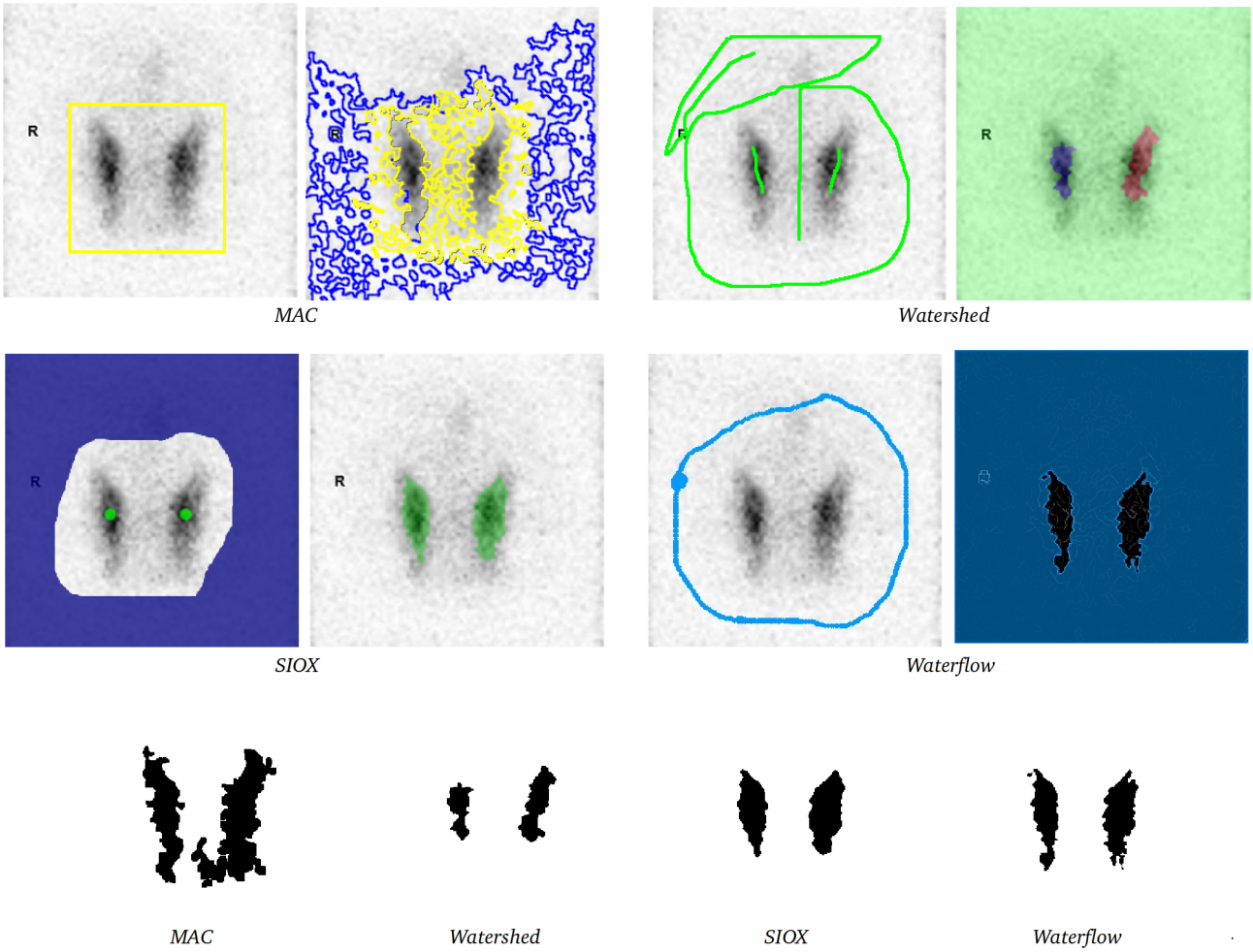


Figure 60: Segmentation on a thyroid scan

Segmentation was carried out by using a higher smoothing setting, and the absolute colour gradient edge map, producing less noise at the peripheries, and stronger, smoother edges towards the objects of interest. The default parameters for edge and region forces are then used as in the basic model, but with a lower actively processed area for improved speed. The initial segmentation resulted in many bubble artefacts at the peripheries, which required a couple more secondary initialisations to get rid of. Note that this could have been done as a simple post-processing step instead, by selecting only the areas of interest using a simple flood-fill selection; this was the case for the MAC model, as the full segmentation obtained was less useful due to the grainy nature of the image.

5.2.6 Ultrasound

Figure 61: Foetal ultrasound

Ultrasound is another modality that produces pictures which can be difficult to interpret, and the overall quality is greatly dependent on the skill and technique of the ultrasonographer carrying out the examination. However, ultrasound is often used to make accurate measurements, such as the diameter of abdominal organs like the gallbladder and kidneys. An ultrasonographer will attempt to focus on the organs of interest, which can make the surrounding structures become defocused and noisy, and create artefacts. More generally it is used to visualise particular features without exposing the patient to harmful radiation. Figure 61 shows an image taken from a foetal ultrasound. Segmentation could be used to track foetal position, make measurements (such as head diameter) or highlight examination features in the developing foetus.

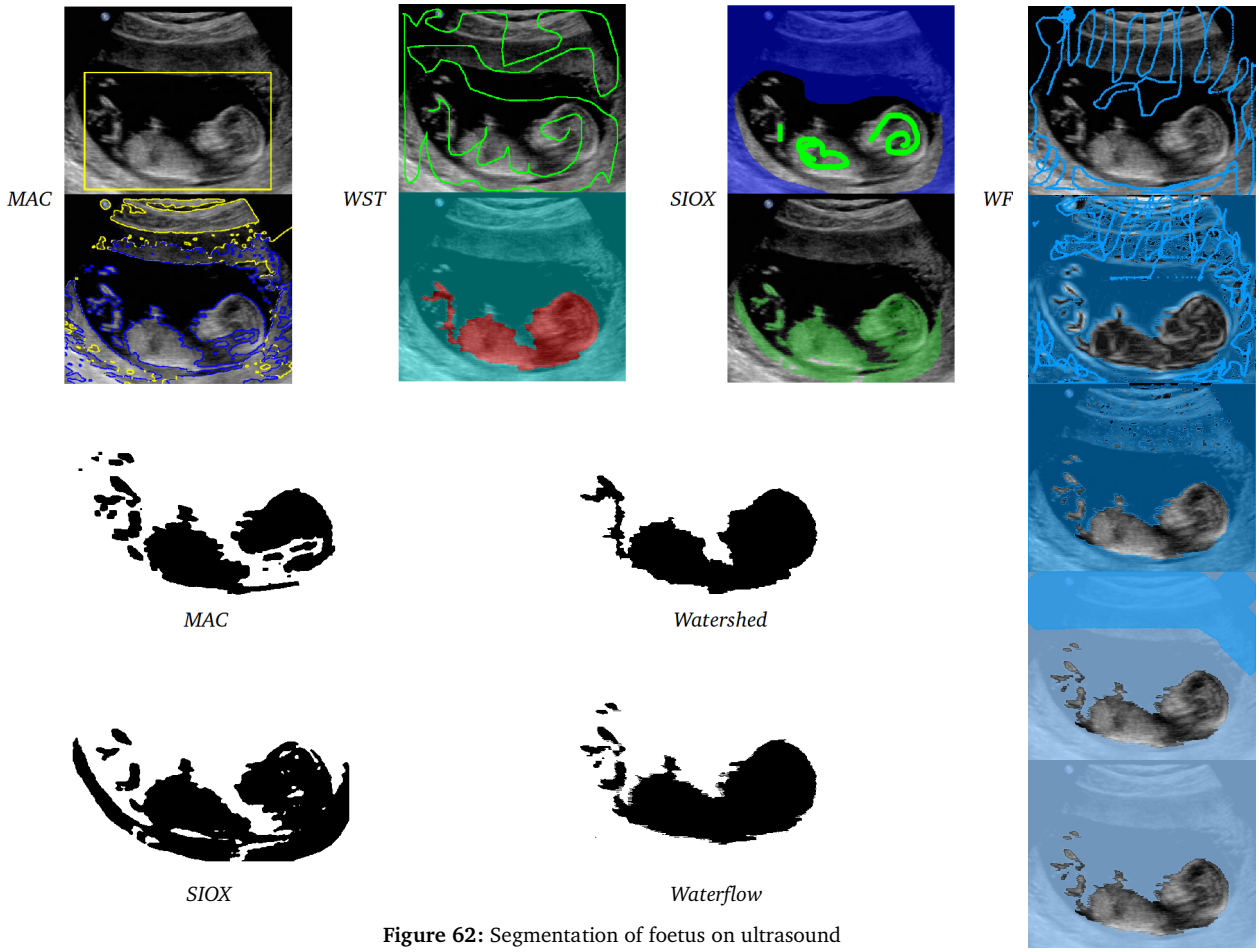


Figure 62: Segmentation of foetus on ultrasound

Segmentation proved difficult for this task, however the end result fares well compared to the other models. Large smoothing area over an intensity gradient map was used, as this produces relatively clean, thick edges, which a preserved adhesion force can use to prevent the water flooding the foetus. After initial segmentation, there were many bubble and hole artefacts (more on the top of the image, away from the object). Artefacts close to the object were eliminated using secondary initialisation with the same parameters. To eliminate the coarse artefacts on the top, the segmentation was frozen, and a new flow was started using a large velocity modifier, so as to expand freely until the user is satisfied the artefacts have been covered, but before the wave reaches the object. (Alternatively, the outlined object could have been extracted through post-processing.)

5.2.7 Retinal photographs

Retinal vessels can provide valuable information, such as vessel diameter, areas with tortuosities or narrowing, abnormal or excessive branching, and so forth, all of which can correlate to underlying pathological processes. Delineating these structures manually is both inefficient and ineffective. Segmentation could provide a way to analyse these features automatically, and would therefore prove invaluable in ophthalmological research and management.

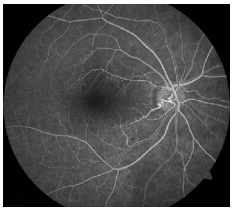


Figure 63: Fluorescein-dye retinal angiography

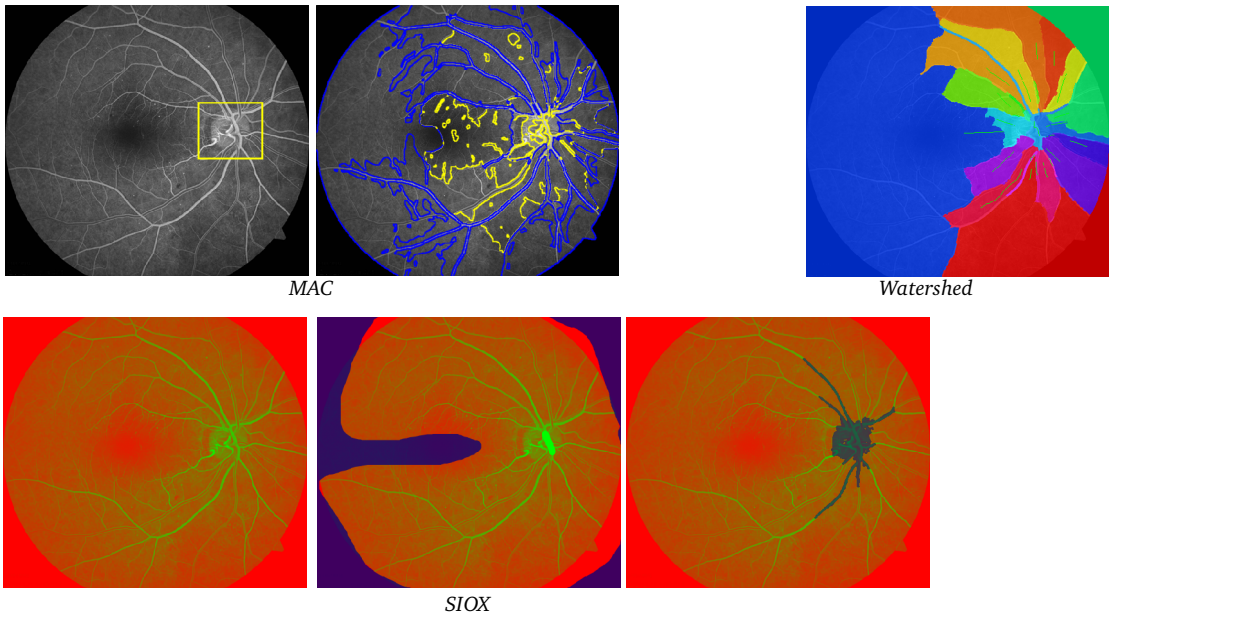


Figure 64: *Retinal vessel segmentation*

The watershed algorithm could not segment along the vessels, so was not used for segmentation. SIOX model did not segment the greyscale image, but false colourisation enabled a basic segmentation based on colour signatures. MAC achieved good initial segmentation but lacked termination control. The Waterflow model achieves the best results, but further propagation along the vessels would have been desirable in this problem, and proved difficult to get without leaking.

6. Discussion

6.1 The basic algorithm

We have confirmed the waterflow framework to be a potent and flexible segmentation framework, and the ideas that underlie its operation to be sound and reasonably straightforward to implement. The experimental results obtained demonstrate that even the basic form of the algorithm, as described originally by Liu and Nixon, achieves comparable segmentation results to other state of the art segmentation tools, from a variety of theoretical backgrounds, and in certain occasions outperforms them. Since segmentation is performed using a combination of both edge-related information and regional-statistics-related information, the waterflow framework shows great adaptability in dealing with objects that display one attribute disproportionately over the other, even when this balance varies within the image itself. The original authors' claims on its ability to segment multiple objects and deal with complex geometries and topologies, including objects with poorly defined edges, is supported by the results obtained using our implementation of the algorithm. Robustness to noise claims are also supported to an extent, particularly for low level noise, although in general we would argue that noise has a more pronounced effect on the basic waterflow model's segmentation outcome, compared to the other models we have examined.

As a substrate for a segmentation tool, the basic model has many merits:

- It is simple and intuitive to use. A user does not need to know the exact way in which the driving forces are implemented in order to achieve a good segmentation result. The paradigm of an expanding body of water, makes the contour evolution fairly predictable, enabling the user to select appropriate segmentation strategies (such as external vs internal initialisation) more intuitively.

- The parameters in the basic model are also intuitive, as they describe physical phenomena relating to the expanding contour.
  - Increasing the velocity modifier λ increases the chances of the water layer overcoming obstacles in general, while increasing the resistance modifier k decreases the chances that a water layer will penetrate (or exit, depending on the initialisation) objects with better defined boundaries.
  - The image force constant α is also reasonably intuitive to use, particularly if the user has access to the edge strength map, and can visually assess the quality of the edge information available, to decide whether to attribute more weight to edge-related forces, or to rely on region-statistics forces instead.

- The framework is quite flexible.
  - Beyond the flexibility granted by its parameters as discussed above, it allows the ability to track the evolving contour, initialise more flowpoints mid-process and selectively obliterate boundaries by initialising over them. This can be used to 'correct' any non-segmentation gaps as they occur (over-segmentation unfortunately is not as simple to remedy, and relies on parameter tweaking instead) or, to resume evolution after the algorithm has terminated, by altering the parameters to try something else from the existing contour, or add more flowpoints. Some snakes, including the MAC, do not provide such a feature (possibly as a further snake would not necessarily add much to the existing segmentation). In the waterflow model, this is a very useful feature, in pushing a 'near-perfect' segmentation, to a 'perfect' result, with only a few extra steps.
  - In our implementation we have included a 'freezing' function, i.e. the image is reset for a new segmentation to be carried out as normal, but the positions of the frozen pixels are

retained in memory, and included in the final segmentation result. This allows for even greater flexibility, as it allows the user to segment two separate objects consecutively, using a distinct set of optimal parameters for each, rather than find a single set to cover both objects, which, depending on the nature of the objects may not exist in the first place. As mentioned above, one could even segment portions of the same object using this strategy, where the nature of an object changes over the image, and have the two segmentation results 'meet half way'; such a strategy would obviously not be useful for automated tasks, but it would be well suited to users in an image manipulation environment. Note that 'freezing' is simply a convenient way of applying changes from within the segmentation tool, that would have otherwise required simple post-processing, such as combining or selectively excluding segmentations.

- It has a good initialisation invariance profile.
  - As long as the flowpoints are properly initialised 'inside' or 'outside' the objects of interest, then the exact position makes little difference to the end result. A slight exception to this is the case where an object has varying textures along its length, and a region-based strategy is used. Rich initialisation over the area of interest in this cases can benefit the end-result, as the statistics of the flooded region will be adjusted accordingly. Furthermore, while in many cases a single flowpoint is enough to segment an object, generous initialisation greatly increases the speed of the algorithm, as there is more water generated per iteration.
  - At the same time, the model is not 'too' initialisation invariant. The MAC model, for instance, boasts extreme initialisation invariance, by producing two snakes evolving in opposite directions, therefore producing the same segmentation whether initialised across a boundary, or away from it. However, this makes the choice of initialisation somewhat meaningless. The snakes produced do not necessarily correspond to an 'internal' and an 'external' snake, rather they cooperate to between them segment all possible areas available to them. It is then left up to the user to select which 'pockets' correspond to segmented regions or not, which is not always an easy task, particularly since occasionally the snakes result in 'open' shapes that cannot be filled. As discussed in the introductory section, initialisation largely accounts for the 'conceptual' components a computer algorithm cannot infer, and is the step which identifies the objects of interest. Therefore, *some* initialisation variance is desirable, to the extent that it intuitively corresponds to different objects. In the waterflow model, initialising across a boundary excludes that boundary from segmentation, which is both intuitive, and useful as a segmentation strategy.

- It achieves good position accuracy. We've demonstrated its ability to appropriately segment complex shapes, not only in terms of their shape, but also in terms of thickness. A disadvantage of edge maps is that, due to the quantized nature of pixels, an edge can appear on either side of the actual object boundary, or even on both sides. On images pre-processed with smoothing, the edge may be even thicker (or displaced in the case of the Canny algorithm). There is therefore the danger that the final contour rests on the wrong side of the actual object boundary, or even a few pixels away from it, contributing to an increased or decreased thickness accordingly (or a shift, if the edges aren't consistently inside or outside the object). The waterflow model is less susceptible to this problem, as it also makes use of pixel statistics, therefore contributing to contour termination before the actual boundary is crossed (compare again with the MAC model in figure 37).

In terms of its theoretical background and implementability, the model lends itself for easy modification and extension, as it relies on simple yet powerful principles, modeled from the physical world.

Perhaps a first criticism of the framework concerns its name. While admittedly, thinking of the expanding contour as a water layer potentially makes the process more intuitive for the user, in reality, the theoretical design of the model has less to do with a physical analogy of flowing water molecules, and more to do with an expanding contour, under the influence of internal and external forces, much like conventional snakes. These 'water forces' are implemented as a Gaussian force field, applied throughout the image, rather than molecule to molecule interactions, which would be closer to a physical analogy with water.

Where it differs to conventional snakes, is the fact that there is no 'whole-contour' energy functional to minimize. Instead, it relies on individual pixels reaching termination criteria, much like in the case of the geodesic snake. The geodesic snake expands or contracts at each contour pixel in a monotonic manner (i.e. the pixels belonging to the contour can only move either 'inwards' or 'outwards' for the entire duration of the segmentation process) at a velocity dependent on image properties – termination simply involves all contour pixels having reached a velocity of 0. The waterflow contour is similarly only capable of monotonic expansion (but not contraction), so perhaps a more fitting name would have been 'selective' or 'intelligent flooding', as the algorithm is essentially similar in concept to a flood-fill algorithm, but is more 'intelligent' in that it utilizes much more sophisticated criteria when trying to select which neighbouring pixels to flood and which not in each iteration.

The monotonic expansion and lack of a whole-contour energy functional to minimize also means that, if a pixel somehow manages to overcome an intended obstacle, there is little to stop it from expanding further (the internal forces were clearly not enough to stop it from overcoming the obstacle in the first place), even if this makes the segmentation less accurate on the whole. This happens since each contour pixel acts relatively independently of the rest of the contour, whereas a contour energy dependent snake would have prevented such destructive expansion, as it would (hopefully) be driving the energy functional to change in the wrong direction. Furthermore, if the contour is initialised in an otherwise unfavourable shape to begin with (such as, accidentally across a boundary, for instance), there is no ability to simultaneously expand and contract in the appropriate places to improve the segmentation, as would happen in a classical non-monotonic energy minimizing snake. At the same time though, in the particular case of initialising across a boundary, as pointed out earlier, this can be a positive thing for the algorithm, as it enables more intuitive and flexible initialisation.

The most significant weakness of the algorithm, however, is its speed. The need to iterate over the entire image per contour pixel, slows down the algorithm considerably. This is particularly noticeable in cases where the objects have a large, convoluted surface area, or where the route followed by the expanding water contour results in a non-smooth outer surface (such as when expanding through noise, for instance), at which point efficiency drops at an exponential rate. Liu and Nixon had concluded the algorithm to be accurate enough for use in a medical setting, but the reality in terms of any software implementation is that the low efficiency (which can lead to a single segmentation sometimes taking hours to complete), renders the model unusable in a typical clinical setting.

Another criticism is the manner in which the contour pixels' propagation and regional statistics act by only taking single neighbouring flood target pixels into account, as opposed to a wider neighbouring cluster. This leads to some unwanted side effects. In the case noisy images, for instance, when a noise pixel is detected which doesn't match the evolving water layer's statistics, it is left out. Similarly, edge-forces are not tested for neighbouring edges reinforcing the notion that an edge pixel is part of a boundary, rather than a single noisy edge. This creates 'bubble' artefacts in the evolving water layer, and grainy-looking boundaries in the segmented object. Perhaps if a small cluster of pixels was taken into account instead, in a similar way to the SIOX model, such bubble artefacts would be eliminated, as the single noisy pixel intensity wouldn't be the sole criterion for

flooding it or not, and also, the cluster could be classified as a cluster likely to contain a boundary or not, further helping the accuracy of the algorithm. Again, however, it's important to note that elimination of such bubble artefacts by choosing the largest segmented area, for instance, (which presumably corresponds to the object of interest in a single object segmentation) as a post-processing step, is a trivial step in terms of image processing (as it can be done with a simple flood-fill selection). In-algorithm hierarchical selection of the segmented contours based on size could also be applied as is the case in the SIOX model. Where bubble artefacts are limited, initialisation over them eliminates them from the segmentation. The grainy boundary appearance, however, is not as easily remediable an issue.

The basic model, as most snake algorithms, was not designed in such a way as to take advantage of colour in pixels. Both the edge information and the regional statistics step enact their effects based on greyscale intensity values, rather than colour information. Yet, use of colour information has the potential to greatly increase the differentiating potential of segmentation algorithms. From the results section, it may be obvious that the SIOX model, which draws its segmentation power from differentiating pixels based on their 'colour signatures', did not perform as well in greyscale images. Furthermore, introduction of false colour, even if technically not altering the overall pixel intensities (or worse, partly equalises the contrast), can still be useful in differentiating the pixels based on their colour statistics. An example of this was shown in figure 64, where the colourised version of the original greyscale image produced better results with SIOX than the original greyscale image (which completely failed to segment). The colourisation technique applied was simply inverting the intensity curve for the red channel, and eliminating the colour blue from the image, causing darker pixels to appear red, and whiter pixels to appear green. Note that the resulting picture has less information in terms of intensity (as the intensity contribution of blue has completely been removed), yet it still yields a better segmentation outcome.

6.2 The extended model

6.2.1 Efficiency

The extended model addresses a significant portion of the above criticisms. Speed is massively improved by the processing area parameter introduced, greatly increasing the usability value of the algorithm, at no real expense of segmentation accuracy.

Another way speed could have been improved, would have been to exclude from the acceleration phase any calculations for contour pixels that have been already immobilised by a boundary, thus limiting the number of pixels needing iteration over the whole image. This was, unfortunately, not implemented as it was a late concept. The basic idea, however, is that a status flag could be attributed to all contour pixels, that monitors whether a contour pixel has moved for the last few iterations. If not, it would be assumed that the pixel has become a 'stationary' contour, and would be flagged as such, therefore excluding it from the acceleration phase calculations. This would be particularly useful in segmentations involving propagation along a narrow 'tube', for instance, as only the contour pixels from the leading front would have to be included in the acceleration phase, whereas the large number of stationary pixels already lining the tube would be excluded.

Naturally, in the normal algorithm, an evolving contour could potentially evolve in such a way as to create new forces acting on a previously stationary pixel, rendering it mobile again. Therefore, it would be necessary for stationary pixels to be re-included every few cycles, to confirm their stationary status. The number of such cycles, as well as the number of iterations immobile pixels need before being labeled as 'stationary', could be controlled by parameters; a simple parameterisation study showing the effects on efficiency and accuracy could suggest optimal values.

Once the algorithm terminates, all stationary pixels would be 'reactivated' and tested again in another iteration over the image, to confirm none has been rendered mobile under the new forces. If no pixel has moved in this last iteration, only then the algorithm has truly terminated.

6.2.2 Leakage

As the ability to prevent leaking of pixels through weak or inappropriate edges is paramount due to the expansile nature of the algorithm, measures have been taken to further limit this behaviour, namely preserved adhesion forces of flooded pixels, and the edge connection algorithm, each dealing with different circumstances. Their use may rely on a 'try-it-and-see' approach, as their effect on the entire image can be unpredictable at times, depending on the image type, but in general they have proven useful when considering a segmentation strategy.

The edge connection algorithm is a very basic algorithm, without any advanced features of trajectory estimation, such as ones used in the canny algorithm. Given how useful a concept it has proven to be in segmentation, it would be useful to expand on this feature further. One idea would be to guide the edge connection process, by drawing thick strokes over the object boundaries. Then only suitable loose edge pixels found within that thick stroke would be connected, ensuring that incongruent connections are minimized. Obviously, a less elegant yet effective solution would be the ability to directly draw on the edge map, therefore ensuring any leakage sites are 'plugged' effectively. This would be useful for plugging minor artefactual holes visible in the edge map, but would require great accuracy from the part of the user, and would not be very useful for objects with broken or illusory contours, such as the ones we've encountered previously, which would in effect require nothing short of manual delineation over the edge map (hence making the segmentation part a rather pointless exercise.)

6.2.3 Colour

The improved segmentation ability of the extended waterflow algorithm where colour information is available, through the ability to use resistance to colour or hue, and an edge-map derived from absolute colour gradients as opposed to just intensity gradients, is clearly evident in the results section. It would have been interesting to complete the pack, by devising a regional statistics functional based on colour information, rather than intensity alone. While most (but not all) medical imaging modalities produce greyscale images, pre-processing with a range of false colourisations, might not only assist with single segmentation attempts, as shown by the SIOX example mentioned above, but could also potentially yield a range of segmentations, each corresponding to a different false colourisation scheme, that can be combined, or 'averaged' in some way, to obtain a more accurate result.

6.2.4 Parameter choice

One of the potential criticisms on the extended model is the increased number of parameters. On one hand, this offers increased segmentation powers and great flexibility in selecting a suitable segmentation strategy, and, for the most part, the extra parameters remain intuitive. On the other hand, however, a typical user should not be expected to 'fiddle with numbers' in order to get a reasonable outcome. In practise, in most cases this wouldn't be necessary, as the default values (which correspond to the basic model), achieve a good outcome in a large percentage of typical images (i.e. a reasonably distinct object on a reasonably differentiable background) – the 'percentage of actively processed area' parameter W would be the only one worthy of defaulting out of the basic model, to speed up the process; empirically, a value of 5% has been found to produce

good results, with good efficiency on images around 300×300 resolution. Less values can be used in higher resolutions, as the absolute area covered is still large enough for reasonably accurate pixel statistics.

In terms of medical images specifically, an observation from the results section is that resistance to brightness or colour tended to be a better segmentation strategy for many image types, rather than the default resistance to edge strength. Also, different images seemed to benefit from similar approaches within the same modality, but required very different segmentation strategies to images from other modalities. In the clinical setting, the number of completely different image types and modalities probably justifies the increased number of parameters, as great flexibility is needed to be able to segment images across so many different categories. However, given that similar images seem to require similar parameters and segmentation strategies, recommended parameter limits specifically suited to each modality could be used as default values, according to the type of image segmented.

An automation algorithm was conceived, to help provide such default parameters specific to a given modality, by optimising performance against a representative 'training' image. This, unfortunately was not included in the final implementation, as it was also a late inspiration. The basic concept is described here in steps:

1) Initialise a set of parameters
    - A random number is selected from a lognormal probability distribution with mean 1 and initial range [0.001,1000] (i.e. obtain random values in the range [0.001,1000], where values close to 0.001 and 1000 occur rarely, and values close to 1 occur most frequently). Such a distribution can be obtained by first using a Box-Muller transform[39] on 2 uniformly distributed random numbers in the range [0,1], returning a normally distributed random number in the range [-1,+1], with mean μ=0. This number is then multiplied by 1000, and then exponentiated using a base 10 (i.e. $\varphi = 10^{1000n}$, where n is the normally distributed random number, and φ the lognormally distributed random number in the range specified above).
    - Multiply each numerical parameter with its own φ value. For true/false parameters, a uniform distribution is used instead, with initial decision threshold at 0.5 (i.e. values below 0.5 give a value of false, otherwise true).
2) Once a complete set of initial randomized parameters is obtained, the waterflow algorithm is run, using those parameters. The accuracy of the segmentation is then recorded against the ground truth obtained from the representative test image.
3) Repeat steps 1 and 2 for *i* iterations (say, 10 for example). This should give 10 different segmentations, from 10 different randomized parameter sets, each with its own accuracy value.
4) Select the parameter set corresponding to the most accurate segmentation
5) Repeat steps 1 to 4 for *j* iterations (say 10 again, giving a total number of 100 experiments conducted), substituting the initial default parameter values with the values obtained at step 4 each time, thereby using progressively more optimised values as the centre of the lognormal distribution each time, which could theoretically converge to the true optimal values in a unimodal distribution, and assuming relative parameter independence. For true/false parameters, the threshold is moved towards the true/false ratio corresponding to the 5 most accurate of the 10 iterations in step 3.

Alternatively, the range of the lognormal distribution could also be changed based on the parameter ranges obtained in the experiments. An identical initialisation can be used each time, if one is thought to be representative enough. If one would want to test the parameters are consistent with initialisation invariance as well, a flowpoint could be placed within the object area randomly everytime, and the whole algorithm iterated again for 10 random initialisations (making the total number of experiments conducted 1000), and so forth. Limits could be imposed (particularly on W)

if particular behaviour is desired (W could be in the range 0-10, for instance, to ensure efficiency, otherwise 100 experiments would take a very long time to complete).

The resulting optimal parameters could then be tested against 'test' images of the same body area and modality, to confirm whether indeed optimality has been achieved compared to other default and manual initialisations.

6.3 The implementation

The algorithm was implemented on C++ using the OpenCV framework. One criticism is that on slower machines, the transition between evolving contours is not displayed on screen, and is buffered instead, resulting in only the final steps being shown. (For instance, if the algorithm is instructed to perform 100 iterations, only the 100^{th} frame will be shown, or the termination outcome, whichever comes first.)

The ability to be able to switch between the original image and the edge map has proven very useful, as edge information is very relevant to determining the segmentation strategy to be used. The SegmentIt watershed tool also provides this feature, while the MAC object saves an edge map imagefile in one of the program's native directories that the user can search for.

It would have been useful to provide the ability to undo a number of steps, as if one wants to experiment with parameter choice, the image needs to be reset each time.

The tool is implemented using a very basic interface provided by OpenCV, displaying an image window, and a console driven menu. This works reasonably well and seems quite robust, but a GUI would have been more user friendly. Parameter selection using sliders could be useful, but the algorithm isn't really fast enough to be able to see the effects of parameter change on segmentation in real time.

Ideally, the algorithm could have been designed to be integrated as a supported plugin to the GIMP suite, enabling it to make use of its powerful features and native graphical interface, and enabling the open source community to experiment and possibly extend this project further.

6.4 Further directions

The framework is by its nature very amenable to modification and improvement. Below we suggest some more ideas possibly worth investigating, further to the suggestions offered above.

6.4.1. Regional Parameters

It would be useful to provide the ability to apply different sets of parameters within particular regions of effect, and the ability to smoothly transition towards the parameter values of new regions as the contour progresses towards them. This means that a segmentation strategy could be employed that takes into account the changing nature of an object along the image. For instance, an object which has clearly defined boundaries on one side, and weak boundaries but a well differentiated texture to its background on the opposite side, could be segmented by a contour which modifies its Image forces balance as it progresses through the object, to match the best image force for the task.

6.4.2 'Anisotropic Filtering' for force calculations.

Rather than selecting a value of W to 5%, ignoring the remaining 95% of the image, an effect similar to 'anisotropic filtering' can be employed, where an average force and average statistics are calculated for larger clusters of pixels, and this average value is used instead, for pixels that are more distant from the 5% range. This could provide slightly increased accuracy, by taking into account the distant forces to an extent, rather than completely obliterating them, but still retaining the relative speed of a curtailed processing range. Having said that, as we've seen previously, there is no significant accuracy penalty with the W parameter, and occasionally the segmentation benefits from it in unexpected ways, so this is not a crucial optimization by any means.

6.4.3 A more physiological water model

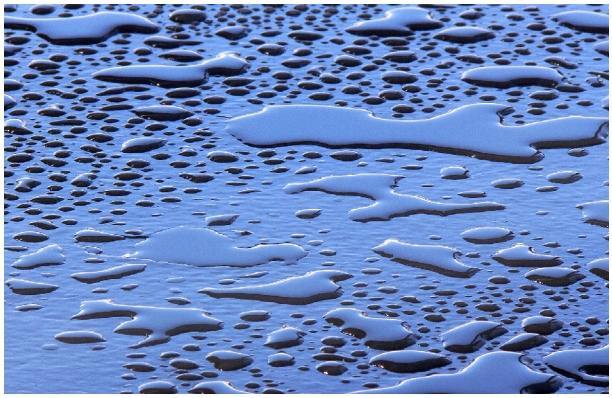


Figure 65: Water coalescing into puddles over a surface

It would be interesting to modify the model to mimic water in a completely physiological way, as 3 dimensional droplets over a 2 dimensional image. This, however, would include simulation of individual molecules, which would probably increase the computational cost significantly. The forces acting on water molecules are predominantly attractive, rather than the repulsive forces described in the waterflow model, and the only 'repulsive' force is essentially a resistance to compression. The attractive forces can give rise to physiological effects such as cohesion (the attraction of water molecules to each other, causing them to coalesce into larger droplets if in close proximity), adhesion, (the attraction of water molecules to surfaces, such as the image surface in this model, causing them to stick to it), and surface energy (causing the droplet to assume a round shape). Different 'liquid types' can be used, in terms of this behaviour (for instance mercury displays large cohesion and coalesces easily into extremely round droplets, but very little adhesion, causing it not to stick on surfaces).

A 'flowpoint' could be a point source from within the image, expelling water particles in random directions on its surface, which would then seek to coalesce and reduce their surface energy as they accumulate in numbers. As more water molecules are introduced into the image, the objective would be to get them to coalesce in order to minimize their surface energy, but also according to the adhesion provided by the image, which can vary in amount according to image properties, in a similar way that different surfaces provide different amounts of adhesion to different liquids. This behaviour solves the problem of monotonicity, as the body of water would now have an energy functional to minimize, namely its surface energy, as in nature, and could actively change shape fluidly in order to accommodate the new surface energy. The algorithm can terminate when the body of water has reached a maximum of compression or droplet height, and the segmentation is the area of the droplet in contact with the image surface.

Such a model would be intuitive to visualise, as an image is truly a 2D representation, rather than a topographic map, but the obstacles remain, represented by areas which essentially behave as hydrophobic or hydrophilic regions. In fact, such areas could also be designated manually, to control the aggregation and flow of water molecules as necessary, encourage entry into narrow passages, or prevent leakage. Furthermore, this model would be subject to all the physical laws that affect these forces, in the form of advanced controllable parameters, such as temperature, solute concentration, surrounding air pressure etc.

7. Conclusions

The waterflow algorithm described by Liu and Nixon[26] was successfully replicated in this study, and confirmed to achieve good segmentation accuracy in a variety of images, comparing favourably with other state of the art models. Based on an analysis of its strengths and weaknesses, the basic model was extended with considerable success, both in terms of improving segmentation accuracy and efficiency, as well as in terms of providing further useful features to it, allowing it to segment different types of images which would prove very difficult in the basic version. Its performance was assessed on medical images, featuring favourable segmentation compared to other algorithms, and great versatility, making it a suitable tool for medical image segmentation.

Acknowledgements

I would like to thank:

- My Project Supervisor, Prof. Mirmehdi for introducing me to this fascinating area of Computer Vision Research, and for his guidance
- My Course Supervisor, Dr Tim Kovacs for his help and guidance throughout the Masters Course, and particularly for his help and advice on my dissertation.
- My parents for their support
- My cat, Waffles, for keeping me company throughout this project.

Bibliography

1. M.W. Brown (1996): "Neuronal responses and recognition memory". *Seminars in the Neurosciences*, v8, p23–32.
2. L.G. Shapiro and G.C. Stockman (2001): "Computer Vision". p279-325, New Jersey, Prentice-Hall.
3. R.J. Baddeley and B.W. Tatler (2006): "High frequency edges (but not contrast) predict where we fixate: A Bayesian system identification analysis". *Vision Research,* v46(18), p2824-2833.
4. N. Rubin, K. Nakayama and R. Shapley (2002): "The role of insight in perceptual learning: evidence from illusory contour perception". In: *Perceptual Learning*, by M. Fahle and T. Poggio (Eds.), MIT Press.
5. E.N. Mortensen (1999): "Applications of Computer Vision to Computer Graphics". *ACM SIGGRAPH* v33(4). Accessed online at http://www.siggraph.org/publications/newsletter/v33n4/contributions/mortensen.html
6. N.A.M. Schellart (2009): "Compendium of Medical Physics, Medical Technology and Biophysics for students, physicians and researchers". p319-320 Dept. of Medical Physics, Academic Medical Center, Amsterdam. Available freely online at: http://www.biomedicalphysics.org/Textbook Medical Physics.htm
7. N.A.M. Schellart (2009): "Compendium of Medical Physics, Medical Technology and Biophysics for students, physicians and researchers". p263-265 Dept. of Medical Physics, Academic Medical Center, Amsterdam. Available freely online at: http://www.biomedicalphysics.org/Textbook Medical Physics.htm
8. N.A.M. Schellart (2009): "Compendium of Medical Physics, Medical Technology and Biophysics for students, physicians and researchers". p284-297 Dept. of Medical Physics, Academic Medical Center, Amsterdam. Available freely online at: http://www.biomedicalphysics.org/Textbook Medical Physics.htm
9. N.A.M. Schellart (2009): "Compendium of Medical Physics, Medical Technology and Biophysics for students, physicians and researchers". p277-279 Dept. of Medical Physics, Academic Medical Center, Amsterdam. Available freely online at http://www.biomedicalphysics.org/Textbook Medical Physics.htm
10. N.A.M. Schellart (2009): "Compendium of Medical Physics, Medical Technology and Biophysics for students, physicians and researchers". p339-341 Dept. of Medical Physics, Academic Medical Center, Amsterdam. Available freely online at http://www.biomedicalphysics.org/Textbook Medical Physics.htm
11. D.L. Pham, C. Xu and J.L. Prince (2000): "Current Methods in Medical Image Segmentation". *Annual Review of Biomedical Engineering* v2, p315-337
12. I.T. Young, J.J. Gerbrands and L.J. Van Vliet (1998): "Fundamentals of Image Processing". Delft University of Technology, Netherlands. Availably freely online at http://www.ph.tn.tudelft.nl/Courses/FIP/noframes/fip.html
13. R.C.Gonzales and R.E. Woods (2002): "Digital Image Processing (2nd ed.)". Prentice Hall, New Jersey, p572-585.
14. M. Kass, A. Witkin and D. Terzopoulos (1988): "Snakes: active contour models". *International Journal of Computer Vision* v1(4), p321-331
15. V. Caselles, R. Kimmel and G. Sapiro (1997): "Geodesic Active Contours". *International Journal of Computer Vision* v22(1), p61-79
16. C. Xu and J.L. Prince (1998): "Snakes, Shapes and Gradient Vector Flow". *IEEE Transactions on Image Processing* v7(3) p359-369
17. C. Xu and J.L. Prince (1998): "Generalized gradient vector flow external forces for active contours". *Signal Processing* v71, p131-139

18. X. Xie and M. Mirmehdi (2008): “MAC: Magnetostatic Active Contour Model”. *IEEE Transactions on Pattern Analaysis and Machine Intelligence* v30(4), p1-15
19. S. Beucher and F. Meyer (1993): “The morphological approach to segmentation: the watershed transformation” In: *Mathematical Morphology in Image Processing* (Ed. E.R. Dougherty), p433-481 .
20. D.E. Ilea and P.F. Whelan (2006): “Color image segmentation using a spatial k-means clustering algorithm”. In: *IMVIP 2006 - 10th International Machine Vision and Image Processing Conference*, 30 Aug - 1 Sep 2006, Dublin, Ireland
21. G. Friedland, K. Jantz and R. Rojas (2005): “SIOX: Simple Interactive Object Extraction in Still Images”. *Proceedings of the IEEE International Symposium on Multimedia (ISM2005)* p253-259, Irvine (California), December, 2005
22. GIMP Team (2009). The GNU Image Manipulation Program. http://www.gimp.org
23. J.L. Bentley (1975): “Multidimensional binary search trees used for associative searching”. *Communications of the ACM.* v18, p509-517
24. X. Liu and M. Nixon (2006): “Water flow based complex feature extraction”. In: *Lecture notes in computer science 2006,* 4179, p833-845
25. X. Liu and M. Nixon (2007): “Image and Volume Segmentation by Water Flow”. In: *Lecture notes in computer science 2007*, 4842, p62-74
26. X. Liu and M. Nixon (2007): “Medical Image Segmentation by Water Flow”: In: *Medical Image Understanding and Analysis MIUA 2007,* July 2007, UK.
27. L. Kärkkäinen (2008): “Yet another Java vs C++ Shootout”. Available at http://zi.fi/shootout/. Accessed on Oct 1st 2009
28. D. Wolf (2001): “Essentials of Electromagnetics for Engineering”. *Cambridge University Press*.
29. S. Beucher (1999): “Image Segmentation and Mathematical Morphology” Accessed online at http://cmm.ensmp.fr/~beucher/wtshed.html on Oct 2009
30. Y. Rubner, C. Tomasi and L.J. Guibas (2000): “The Earth Mover's Distance as a Metric for Image Retrieval”. *International Journal of Computer vision* v40(2), p99-121
31. The Open Source Computer Vision (OpenCV) library. http://opencv.willowgarage.com/
32. The NetPBM suite. http://netpbm.sourceforge.net/
33. J. Canny (1986): “A computational approach to edge detection”. *IEEE Transactions on Pattern Analysis and Machine Intelligence.* V8, p679-714
34. R.C.Gonzales and R.E. Woods (2002): “Digital Image Processing (2nd ed.)”. Prentice Hall, New Jersey, p119-123
35. R. von der Heydt, E. Peterhans and G. Baumgartner (1984): "Illusory contours and cortical neuron responses.". *Science* v244(4654), p1260–1262
36. Y.M. Jung and J. Shen (2008): “First-order modeling and stability analysis of illusory contours”. *Journal of Visual Communication and Image Representation* v19(1), p42-55
37. P. Kumar, J.C. Leonidas, M. Ashtari, B. Napolitano and A.M. Steele (2005): “Comparison of lung area by chest radiograph, with estimation of lung volume by helium dilution during prone and supine positioning in mechanically ventilated preterm infants: A pilot study”. *Pediatric Pulmonology*. V40(3), p219-222.
38. D.P. Harrington (1983): “Digital subtraction angiography: A special issue” *Cardiovascular and Interventional Radiology* v6(4-6)
39. G.E.P. Box and M.E. Muller (1958): “A note on the generation of random normal deviates” *Annals of Mathematical Statistics* v29(2), p610-611

Image sources

All figures created manually by the author, using the relevant segmentation tools, and the GNU Image Manipulation Program (GIMP), except for the following figures:

Figure 1: Edited version of http://www.hsmo.org/m_adopt/images/ClaudineandParkeratTowerGrovePark_500.jpg
Figures 2 & 3: Adapted from [4]
Figure 4: Adapted from Z.S. Matar, M.S. Al Awami, A.A. Mohamed and M. Abukhater (2008):"Neglected Retrosternal Goitre" The Internet Journal of Surgery. V16(1)
Figures 5 & 6: Available at http://www.ct-scan-info.com/head-ct.html
Figure 7: Available at http://www.nkyxray.com/images/medical/hyperthyroidism.jpg
Figure 8: Available at http://www.radiologybeyondatextbook.com/radiology/tutorials/images/ultrasound.jpg
Figure 9: Adapted from T. McInerney and D. Terzopoulos (1996): "Deformable Models in Medical Image analysis: A Survey". *Medical Image Analysis* v1(2), p91-108
Figure 10: Adapted from http://cmm.ensmp.fr/~beucher/wtshed.html
Figure 11: Adapted from http://www.siox.org/gimp233.html
Figure 12: Original ying-yang image from http://z.about.com/d/taoism/1/0/0/-/-/-/yinYang.gif
Figure 14: Adapted from http://www.cs.bris.ac.uk/images/
Figure 16: Table adapted from http://en.wikipedia.org/wiki/Accuracy_and_precision
Figures 17, 18 & 22: Image concept reproduced from [24]
Figure 34: Adapted from [18]
Figure 42: Adapted from www.cs.bris.ac.uk
Figure 45: Adapted from http://en.wikipedia.org/wiki/Ishihara_color_test
Figure 47: Available from http://simplyfitnessgear.com/blog/breathing-exercises-for-your-lungs
Figure 49: Adapted from https://www.webpax.com/4673-1001-1438-2449/full_screen/series_402.php
Figure 51: Available at http://www.ct-scan-info.com/ctkidney.html
Figure 53: Available at http://www.pharmacology2000.com/822_1/Normal_brainCT.jpg
Figure 55: Available at http://lcib.rutgers.edu/~jon/COFEMI/T1brain.png
Figure 57: Avalable at http://mphotos.live.com/6542a267876ae2fe/d.aspx?rid=6542A267876AE2FE!555
Figure 59: Available at http://www.nkyxray.com/diag_RadioactiveIodineTherapy.html
Figure 61: Available at http://saysomethingfunny.files.wordpress.com/2009/02/babyultrasound.jpg
Figure 63: Available at http://ocularimaging.biz/images/web7.gif
Figure 65: Available at http://upload.wikimedia.org/wikipedia/commons/8/8e/Exploring_new_continents_1200728.JPG