Electronic Letters on Computer Vision and Image Analysis 4(1):32-48, 2004

An Interactive Algorithm for Image Smoothing and Segmentation

M. C. de Andrade*

**Centro de Desenvolvimento da Tecnologia Nuclear - CDTN, P.O. BOX 941, Belo Horizonte, MG, Brazil*

Received 2 June 2004; accepted 22 July 2004

Abstract

This work introduces an interactive algorithm for image smoothing and segmentation. A non-linear partial differential equation is employed to smooth the image while preserving contours. The segmentation is a region-growing and merging process initiated around image minima (seeds), which are automatically detected, labeled and eventually merged. The user places one *marker* per region of interest. Accurate and fast segmentation results can be achieved for gray and color images using this simple method.

Key Words: Image Segmentation, Partial Differential Equations, Watershed, Image Denoising, Seeded Region-Growing, Deformable Models.

1 Introduction

Image denoising and segmentation play an important role in image analysis and computer vision. Image denoising reduces the noise introduced by the image acquisition process, while image segmentation recovers the regions associated to the objects they represent in a given image. Image segmentation typically relies on semantically poor information, directly obtained from the image around a spatially restrained neighborhood and, for this reason, is broadly classified as a *low-level* treatment [6].

Image segmentation often requires pre- and post-processing steps, where user judgment is fundamental and feeds information of highly semantic content back into the process. Pre-processing is an essential step, in which specialized filters smooth the image, simplifying it for the subsequent segmentation step. Interactive segmentation allows the user to intervene directly in the segmentation process thus contributing to its success. Additionally, post-processing may be required to complete the task, if the segmentation itself fails to produce the desired results.

Image segmentation is an application-oriented problem. There is no general-purpose segmentation method. The choice of a particular technique depends on the nature of the image (non-homogeneous illumination, presence of noise or texture, ill-defined contours, occlusions), post-segmentation operations (shape recognition, interpretation, localization, measurements), primitives to be extracted (contours, straight segments, regions, shapes, textures) and on physical limitations (algorithmic complexity, real-time execution, available memory) [6]. Moreover, other important issues concerning fundamental aspects of image segmentation methods such as, initialization, convergence, ability to handle topological changes, stopping criteria and over-segmentation, must be taken into account. Therefore, the performance of a segmentation method can not be evaluated beforehand, its quality can only be evaluated by the results

Correspondence to: mca@cdtn.br

Recommended for acceptance by Enric Martí
ELCVIA ISSN: 1577-5097
Published by Computer Vision Center / Universitat Autonoma de Barcelona, Barcelona, Spain

obtained from the treatments using the extracted primitives. However, many of the difficulties found in image segmentation can be reduced by adequately smoothing the image during the pre-processing step.

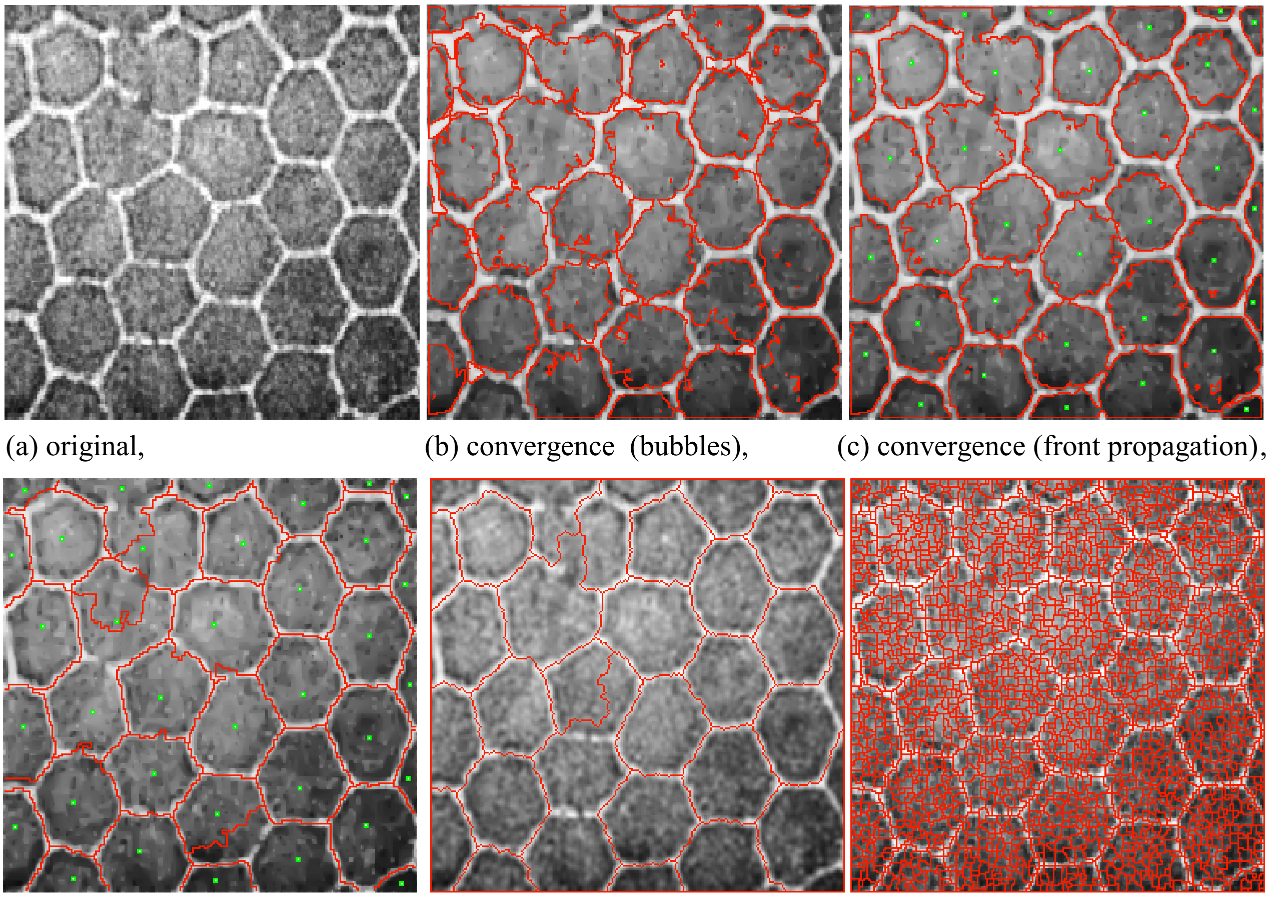
Segmentation by *deformable models - DM* describes contours, which evolve under a suitable energy functional. The pioneer work of Kass et. al. [12], the *snakes method* uses image forces and external constraints to guide the evolution of the *DMs* by minimizing the energy of spline curves and surfaces. Former versions of this method required the initialization to be done close to the boundaries of the objects, to guarantee proper convergence and to avoid being trapped by local minima. The *gradient vector flow* [28], an improved version of the *snakes method,* largely solved the poor convergence problem. The *balloon method* [7] adds an inflation force to the *snakes*, to move the initialized model into the neighborhood of the edges, avoiding local minima. However, the inflation force often pushes the contour over weak edges.

Modelling the contours in the level set framework [20, 21], easily solves the topological problem, i.e., merging of the non-significant regions (or curves enveloping them). The *active contours method* presented by Caselles et. al. [5] and the *front propagation method* introduced by Malladi et. al [13, 14], for example, greatly simplify the topological problem but do not address the initialization and convergence issues. Initialization is usually difficult and time-consuming requiring the manual introduction of polygons around the features of interest. Convergence is also difficult since some models are still evolving while others have finished the evolution or, worse, have leaked through weak boundaries. The geometrical version of the *active contours method* is stable and retrieves simultaneously several contours but do not retrieves angles [5]. The *bubbles method* [23] simplifies the initialization process by allowing, for instance, contours to be initialized at the image minima or at predefined grid cells having homogeneous statistical properties. However, *bubbles method* requires fine tuned parameters in order to achieve simultaneous convergence of bubbles. Moreover, it is slow as compared to *watershed-based* methods [25, 16].

Conventional *region-growing and merging* methods work well in noisy images but are sensitive to seed initialization and produce jagged boundaries. For example, the s*eeded-region-growing method - SRG* [1, 15], introduces a competition between regions by ordering all pixels according to some suitable criteria, a property inherited from the *non-hierarchical watershed method - NHW* [25, 26]. This global competition ensures that the growth of regions near weak or diffuse edges is delayed until other regions have the chance to reach these areas. However, SRG does not incorporate any geometric information and hence can leak through narrow gaps or weak edges. Another approach, the *region competition method – RC* [30] combines the geometrical features of the DM and the statistical nature of *SRG*. This method introduces a local competition that exchange pixels between regions, resulting in a decrease in energy, thus allowing recovery from errors. However, *RC* produces jagged boundaries and depends on seed initialization, which eventually might lead to leakage through diffuse boundaries, if the seeds are asymmetrically initialized [19].

The *non-hierarchical watershed method* as proposed by Vincent [25, 26] – *NHW*, treats the image as a 3D surface, starts the region growing from the surface minima, and expands the regions inside the respective *zone of influence* of each minimum. The region-growing process evolves all over the image, stopping where adjacent regions get into contact. At these points *barriers* are erected. This solution provides a powerful stopping criterion, difficult to achieve in the PDE-based level set framework. However, *NHW* may leads to a strong over-segmentation if proper image smoothing is not provided. There are solutions to the over-segmentation problem like CBMA [2] and Characteristics Extraction [24], however, they depend on interactively tuning parameters related to geometric features of the regions of interest. The *watershed method* as proposed by Meyer- *MW* [16] starts the region-growing process from markers. *MW* is optimal since each pixel and its immediate neighbourhood are visited only once. However, highly specialized filters are required to extract the markers. Finally, the *skeletally coupled deformable models method - SCDM* [19] combines features of curve evolution deformable models, such as *bubbles* and *region-competition* methods and introduces an inter-seed skeleton to mediate the segmentation. However, it requires an elaborated sub-pixel implementation [19, 22].

Not intended as a comparison but only as an illustration, Figure 1 shows some of the main issues of the above mentioned image segmentation methods. This microscopic image shows bovine endothelial corneal cells acquired through a CCD camera attached to a microscope. The original 256 gray-scale image is depicted in (a). The simultaneous convergence problem can be observed in (b) using the *bubbles method* with bubbles initialised at image minima and in (c) using the front-propagation method with 36 seeds initialised by hand. Notice that while some bubbles are still evolving, some have converged and others are being merged. Another problem, "leaking" can occur through weak or diffuse edges, as can be observed in (d) and (e), with *seeded region-growing method* and *CBMA*[2] respectively. Over-segmentation (f) results from the excessive number of local minima and occurs in *watershed method* if appropriate denoising is not provided.



(a) original, (b) convergence (bubbles), (c) convergence (front propagation),

(d) leaking (SRG), (e) leaking (CBMA), (f) over-segmentation (watershed).

Figure 1. (a) original image of bovine endothelial cells. (b) and (c) simultaneous convergence problem. (d) and (e) leaking through weak or diffuse edges. (f) over-segmentation occurs if appropriate denoising is not provided.

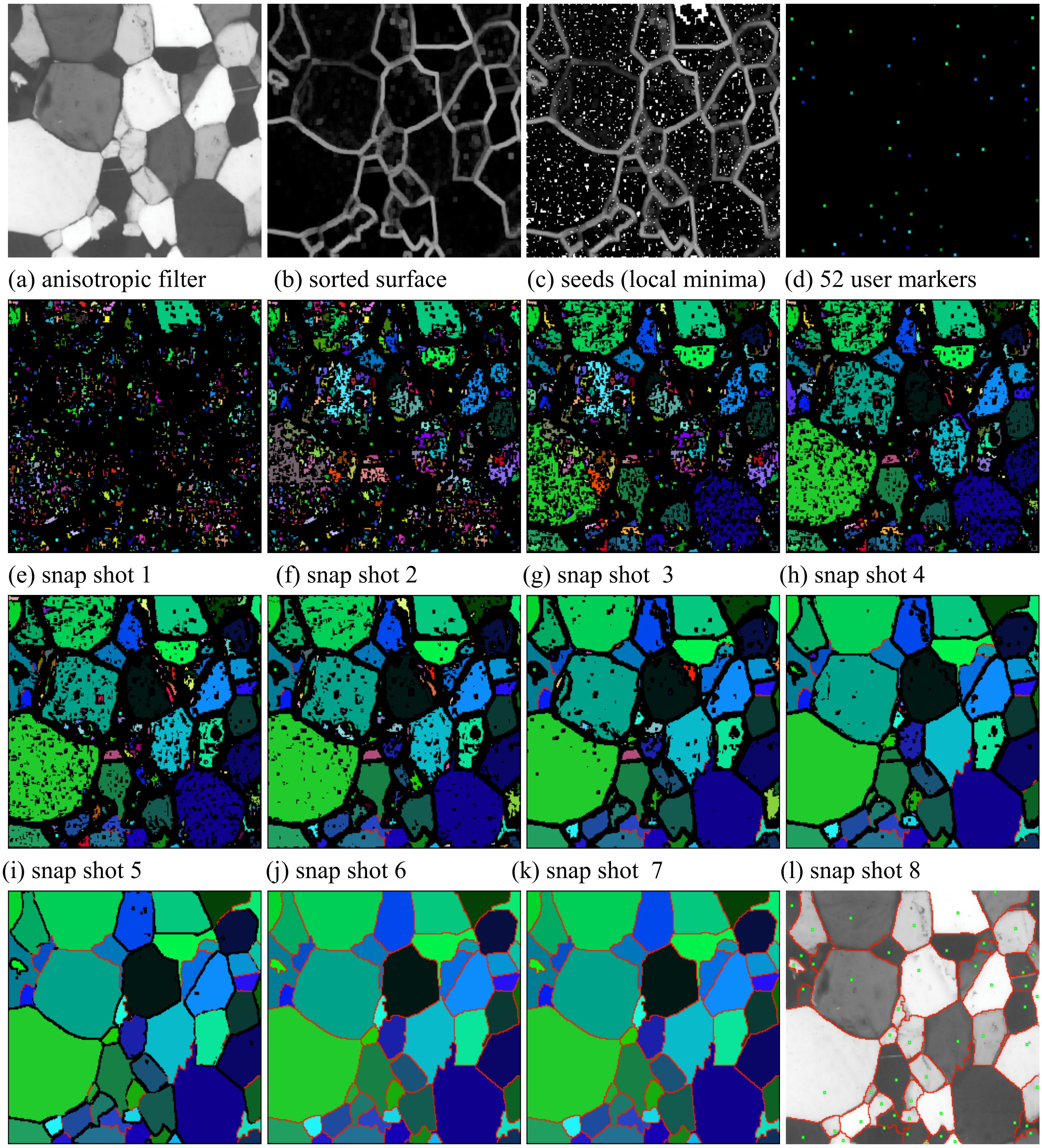
In this paper, an interactive algorithm for image smoothing and segmentation – ISS is introduced. This approach overcomes some of the limitations of previous methods, while retaining some of their most attractive features. ISS combines a noise removal step, which preserve the edges with an interactive image segmentation step, resulting in a robust and easy-to-use technique where higher level knowledge about the image can readily be incorporated in the segmentation process. ISS simplifies the problem of initialization, and provides an integrated solution to the problems of automatic stopping, simultaneous convergence and over-segmentation.

2. The interactive image smoothing and segmentation algorithm - ISS

ISS treats the image as a 3D surface in evolution. This construction serves a dual purpose. At first, implemented in the PDE-based level set framework [20, 21], an edge preserving smoothing algorithm removes noise by constraining the surface to evolve according to its vertically projected mean curvature [29, 27]. Secondly, inspired in the watershed transformation [26] and implemented in the Mathematical Morphology framework [3, 4, 16, 17, 25, 26, 2, 8, 9], a fast and robust algorithm segments the image simulating an immersion on its surface. In this context, segmentation can be described as a *region growing and merging* process starting from surface local minima. To deal with the over-segmentation problem, ISS merges non-significant regions as the immersion simulation takes place. The immersion obeys an order of processing selected by the user according to a criterion based on the image characteristics. All image pixels are previously sorted according to the selected criterion. Sorting provides an order of processing and assures that pixels lying around the edges have their processing postponed. Previously sorting all image pixels in ascending order also provides a way to make detection and labelling of the surface *minima* fully automatic. A detailed explanation of the different sorting criteria can be found in Section 2.3. ISS segments the image into as many regions as the number of *markers* interactively placed by the user. This means that one and only one *marker* per region-of-interest is required. Simple rules guide the merging process: two adjacent regions, growing around local minima, are blindly merged if they do not have *markers*, or if only one of them has a *marker*. Hence, merging is only prevented when two adjacent regions already having *markers,* get into contact. At this point an edge has been found. These rules assure that the topological changes required to reduce the over-segmentation be easily handled through this merging mechanism.

Figure 2 illustrates the steps in the evolution of the ISS algorithm for a sample of rock. Figure 2a shows a 256 gray-scale microscopic image of a polished rock after applying the PDE based denoising filter for 10 iterations. This particular image presents sharp transitions between regions presenting homogeneous but different intensities. A convenient processing order can be established, in this case, by sorting pixels according to the difference between the maximum and minimum gray-levels (morphological gradient) inside the pixel neighbourhood *N(p).* Since this difference is higher around the edges, sorting all image pixels in ascending order according to this criterion will assure that pixels lying around the edges will be the last ones to be processed. Figure 2b shows the morphological gradient image. Figure 2c shows the minima of Figure 2b (white spots) superimposed on it. These minima constitute the set of *seeds,* which are automatically detected and labelled by the ISS algorithm as the evolution takes place. Figure 2d shows the 52 *markers* placed by the user (colored squares) and associated to each region-of-interest. By comparing Figures 2d and 2p it is clear that there is a one-to-one correspondence between each *marker* and each region extracted by the ISS algorithm. Figures 2d to 2o show snap-shots of the region-growing evolution. Finally, Figure 2p shows the ISS segmentation result superimposed on the original image, after all non-significant regions have been merged.

As an interactive segmentation algorithm, ISS requires manual inclusion and exclusion of *markers*. The user repeats the process until satisfactory results are achieved. Interactivity improves the segmentation results by allowing high-level information about the image to be fed back into the process.



(a) anisotropic filter (b) sorted surface (c) seeds (local minima) (d) 52 user markers

(e) snap shot 1 (f) snap shot 2 (g) snap shot 3 (h) snap shot 4

(i) snap shot 5 (j) snap shot 6 (k) snap shot 7 (l) snap shot 8

(m) snap shot 9 (n) snap shot 10 (o) snap shot 11 (p) ISS – 52 markers

Figure 2. ISS algorithm in action

2.1 Edge preserving smoothing under controlled curvature motion

Surface evolution under partial differential equations (PDEs) based level set framework has successfully been used to perform both image denoising and image segmentation. For the purpose of image denoising, PDEs can be utilized to modify the image topology and implement an edge preserving smoothing under controlled curvature motion [29].

By treating the image *I(x,y,z(t))* as a 3D time-dependent surface and selectively deforming this surface based on the vertical projection of its mean curvature, effectively removes most of the non-significant image extrema. For smoothing purposes, the surface height *z* at the point *p(x,y)* is initialized as the value of the local gray-level. The local surface deformation is computed from the local mean curvature κ expressed by the following relation between the second derivatives of *I*:

$$\kappa = \frac{I_{xx}(1+I_y^2) - 2I_x I_y I_{xy} + I_{yy}(1+I_x^2)}{2(1+I_x^2+I_y^2)^{3/2}} \tag{1}$$

To evolve the image *I* as a surface under this modified level set curvature motion is equivalent to repeatedly iterate the following edge-preserving anisotropic filter:

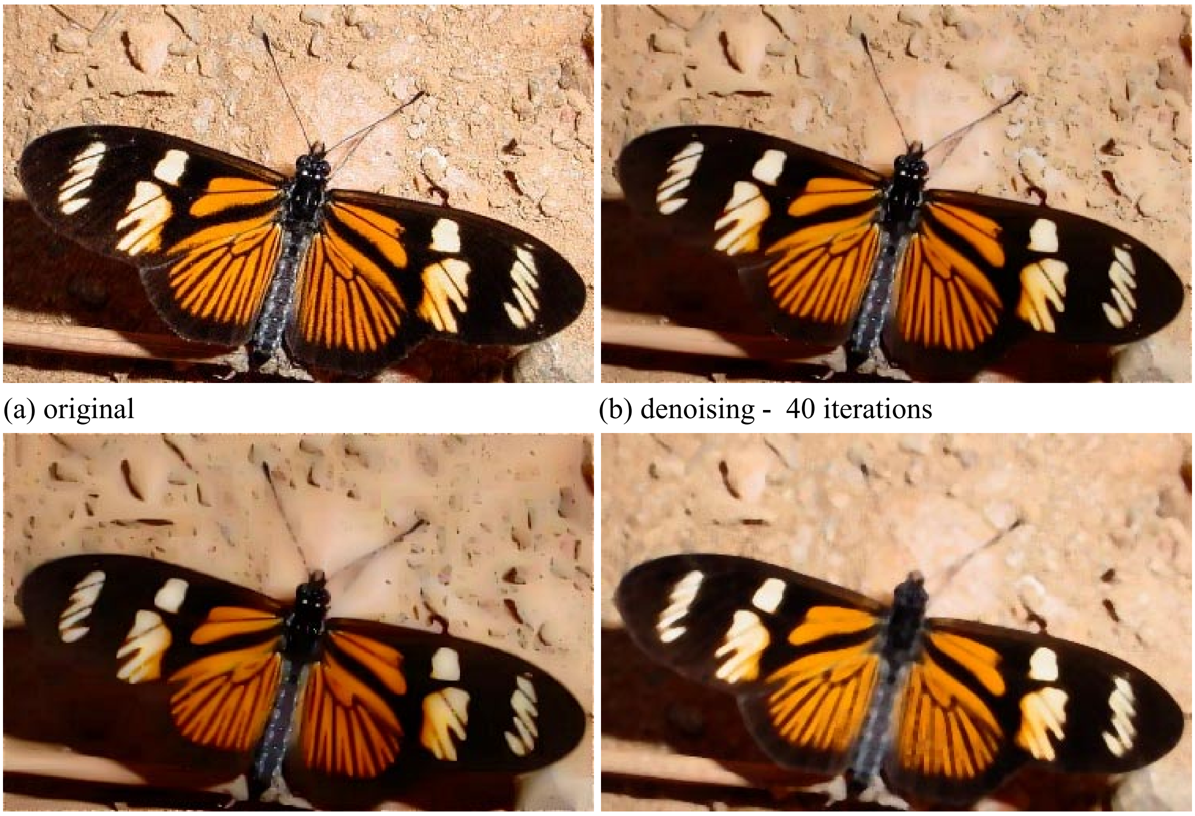
$$I_{t+1} = I_t + \kappa \tag{2}$$

Appendices I and II present ISS pseudo-code and ISS execution time for test-images, respectively.

2.2.1 Stopping criteria for curvature based denoising

The decision regarding when to stop the iterative process depends on the image characteristics and on the regions to be extracted. At each step, the image is slightly “flattened” according to its local curvature. It is important to notice that repeatedly applying this filter may “erase” the image, therefore user judgement is crucial in deciding when to stop. If features being extracted are relatively homogeneous a slight denoising may be sufficient to remove noise allowing good segmentation. Images presenting inhomogeneous regions may require more iterations, while some images may be segmented without smoothing at all.

Figure 3 illustrates an example of image denoising using Equation 2. The original RGB image of a butterfly is shown in Figure 3a. Figures 3b and 3c illustrate the results of applying the anisotropic filter on the original image during 40 and 80 iterations, respectively. It can be observed that as the number of iteration increases, regions become more homogeneous at the expenses of loosing some fine detail. For the purpose of comparison, Figure 3d shows the *median* filter applied on the original image.



(a) original

(b) denoising - 40 iterations

(c) denoising - 80 iterations

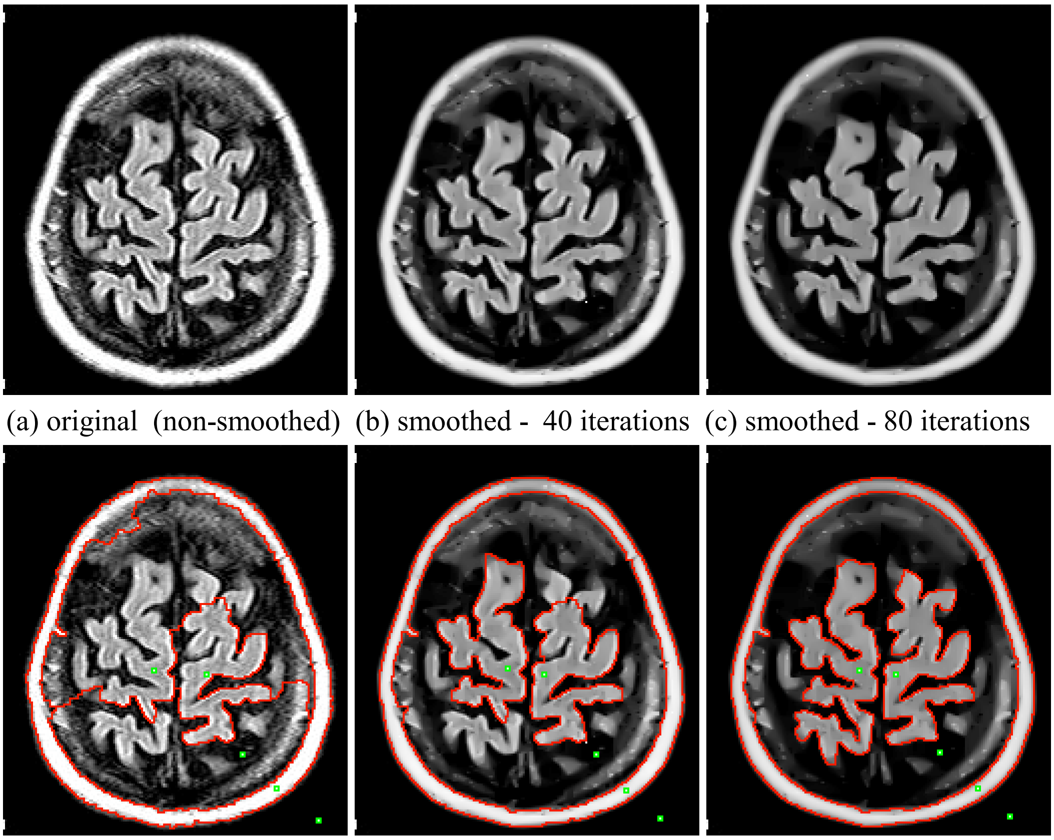
(d) median filter

Figure 3. (a), original RGB image of a butterfly. (b), denoising after 40 iterations. (c), denoising after 80 iterations. (d) median filter.

2.2.2. Effect of denoising on the ISS

Denoising increases region homogeneity by removing or reducing local extrema. This is translated into smoother and better-localised edges after segmentation. Usually, the effort spent on denoising varies depending on image characteristics. The effect of denoising on the ISS segmentation can be perceived on Figure 4. Figure 4a shows a 256 gray-scale MRI image of a brain slice. Figures 4b and 4c show the result of applying the anisotropic filter described by Equation 2, for 40 and 80 iterations, respectively. Figures 4d, 4e and 4f show the ISS segmentation result for the corresponding filtered and non-filtered images. Notice that 40 iterations were insufficient to extract the edges. However, after 80 iterations regions became sufficiently homogeneous. It can also be perceived that after denoising edges became less jagged and more precisely localised.

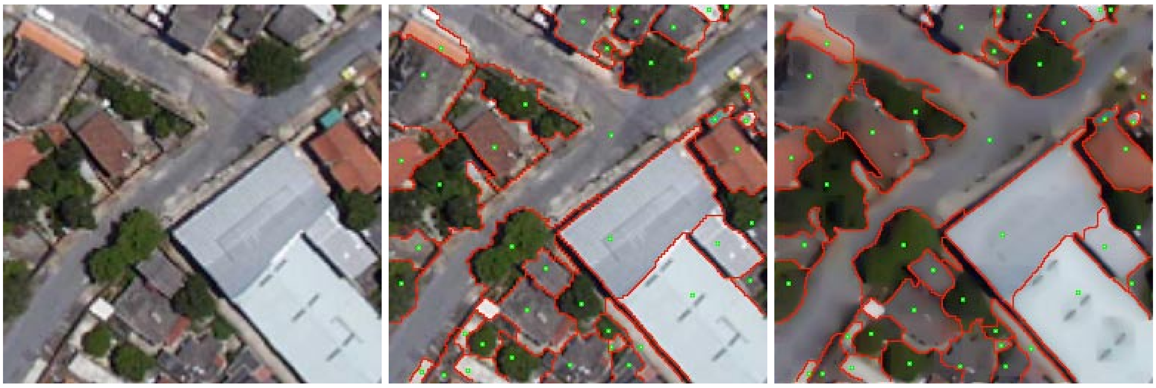
Another example of the effect of denoising on the ISS segmentation can be observed in the aerial image of Figure 5. In this image denoising had little effect on segmentation, since the original non-smoothed image already presented highly homogeneous regions and sharp transitions between them. Comparing segmentation results in Figure 5b (non-smoothed) and Figure 5c (smoothed during 40 iterations) shows that denoising slightly improved the edges.



(a) original (non-smoothed) (b) smoothed - 40 iterations (c) smoothed - 80 iterations

(d) non-smoothed (e) smoothed - 40 iterations (f) smoothed - 80 iterations

Figure 4. Effect of denoising on ISS segmentation.



(g) original (non-smoothed). (h) non-smoothed (i) smoothed – 40 iterations

Figure 5. Effect of denoising on ISS segmentation, another example.

2.2 The interactive region growing and merging step

In region-growing methods, the regions are expanded around *seeds* obeying a given *processing order*. Usually, the regions grow in successive layers until the growing process finally stops thus defining the location of the edges. From this perspective, the most important pixels are precisely those located in a narrow-band around the final location of the edges. Sorting all image pixels according to a convenient relation between each pixel *p(x,y)* and its neighborhood *N(p)* is, in most cases, sufficient to impose such *processing order*, deferring the processing of the pixels on the edges. Many useful relations can be used to sort the pixels.

This ordering can be established, for instance, by defining a *3D* surface whose height *z*, at each point *p(x,y)*, is given by this relation. Sorting the *z's* in ascending order allows the region-growing process to automatically start from the minima of the sorted surface. The following relations, for instance, were implemented in the ISS:

- In its simplest form, to z is assigned the value of the image gray levels themselves; or
- *z* could be computed as the difference between a pixel and mean value in *N(p)* as in the *SRG method*;
- *z* computed as the difference between the maximum and the minimum values in *N(p)*; It's equivalent to compute the morphological gradient;
- *z* as the mean curvature at *p(x,y)* as expressed by equation 2.

The first relation is useful when the image characteristics are such that the gray-levels already dictate a natural processing order. In the example shown in Figure 2a, the regions already have edges at *higher* elevations than their inner parts. The second relation is useful for images having homogeneous textures. The third relation is useful, for instance, in images having discrete transitions between the regions having homogeneous gray-levels, as shown in Figure 4a. In this case, taking the difference between the maximum and the minimum in *N(x)*, forces higher values at the edges and, also has the additional benefit of closing small gaps at the borders.

Finally, by adding a merging mechanism, controlled by user-placed seeds, the region-growing and merging process is complete. A correspondence table, as shown below, can be used to merge the regions. This table is initialized as a sequence of integers from 1 to *N*, where *N* is the number of minima present in the image. *N* is updated according to the temporal sequence of absorptions. If, for instance, the region having label = 1 absorbs the region having label = 3, the merging table is updated as shown below:

before	1	2	3	4	5	…	i	…	N
after	1	2	1	4	5	…	i	…	N

2.3 The ISS algorithm

- Apply to the image the edge preserving anisotropic filter, described by Equation 2. Repeatedly applying this filter can erase most of the significant information present in the image. Thus, the iterative process has to be stopped after a reasonable result is achieved. User judgement and the application requirements should be taken into account to decide when to stop. See Appendix I for a pseudo-code of this algorithm.

- By using a mouse, place one *marker* per region, labeling them from 1 to *N*. *N* is the total number of markers. A marker may be a single point or a set of points of arbitrary shape.

- Sort all image pixels in ascending order, by the address calculation technique presented by Issac et. al. [11], according to one of the criteria listed below:
  - gray level of the current pixel;
  - difference between the maximum and minimum values in *N(p)* of the current pixel;
  - difference between a pixel and the average of it's neighbors;
  - mean curvature at the current pixel;
  - any other criteria which can be used to defer the processing of the edges.
- For each pixel *p* extracted from the sorted list, find how many positive labelled pixels exist in its neighborhood *N(p)*. The three possible outcomes are:
  - There is no positive labelled pixel in *N(p)*. The current pixel receives a new label and starts a new region. New regions receive labels starting from *N+1*. Notice that labels from 1 to *N* are reserved for user placed *markers*. Labels starting from *N+1* are reserved to *seeds.*
  - There is only one labelled pixel in *N(p)*. The current pixel receives this label and is integrated into the corresponding neighbour region.
  - There are 2 or more positive labelled pixels in *N(p)*. If 2 or more neighbors have *markers* labels (label <= *N*), a border has been found, mark the current pixel as a "border", say a -1 label. Otherwise merge all neighbors into one region (the one having the smaller label; i.e., the first labeled in *N(p)*) and add the current pixel to it. If there are 2 labelled pixels in *N(p)* and one has *marker* label and the other a *seed* label, the one having a *marker* label absorbs the one having a *seed* label.
- By using a merging table, re-label all pixels to reflect the absorption they have undergone.
- Draw the segmented image according to the newly assigned labels.

Appendix I and II present ISS pseudo-code and ISS execution time for test-images, respectively.

3 Applications

This section illustrates some practical results obtained with the ISS algorithm for different classes of image and also the segmentation obtained with other methods. Figures 6, 7 and 8 present ISS segmentation for microscopic images of ceramic, geological and medical images. Figure 9 illustrates the performance of ISS and other segmentation methods on different kind of image. In the segmented images, user selected *markers* are shown as green dots and the extracted edges are shown as red lines. Figure 6a presents a micrograph of ceramic material containing grains (dark gray) separated by thin gaps (light gray). Observing that pixels on the edges are lighter than inside grains, they were sorted and processed according to the original intensity of the gray levels, i.e., from darker to lighter. Figure 6b shows the ISS segmentation result. Figure 7a shows a color micrograph of a geological sample containing several grains. As this image presents homogeneous regions and discrete transitions between them, pixels were sorted in ascending order and processed according to the intensities of the morphological gradient (difference between maximum and minimum gray in *N(p)*), thus delaying the processing of the pixels around the edges. Figure 7b shows the ISS segmentation result. Figure 8a shows the micrograph of a cross-section of a human renal glomerulus containing the Bowman's capsule, the vascular pole, and surrounding structures. Figure 8b shows the ISS segmentation result. Again, the morphological gradient was used to sort and process these pixels. Notice that even barely perceptible edges were precisely extracted in these images.

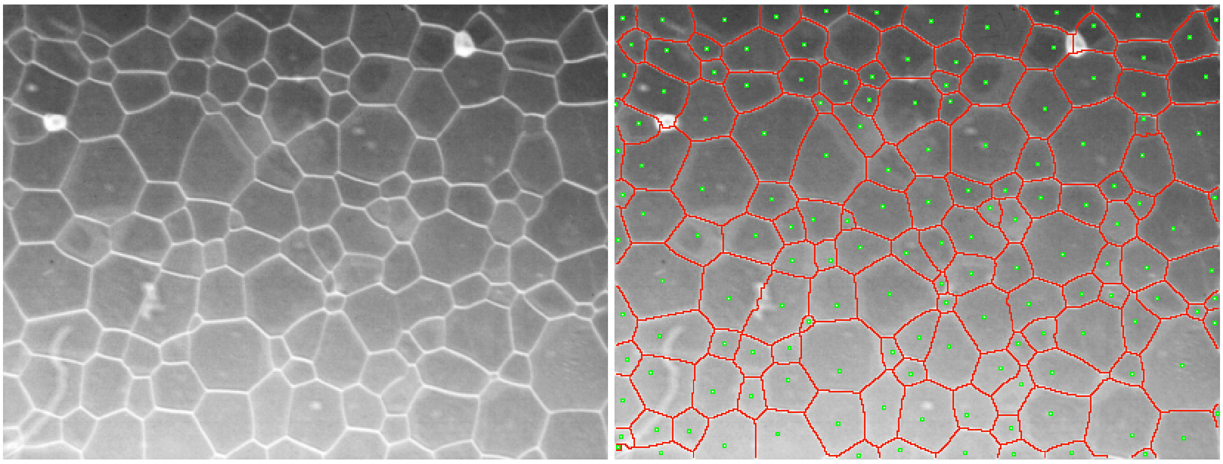


Figure 6 (a), micrograph of a ceramic sample. (b), ISS segmentation result (152 markers).

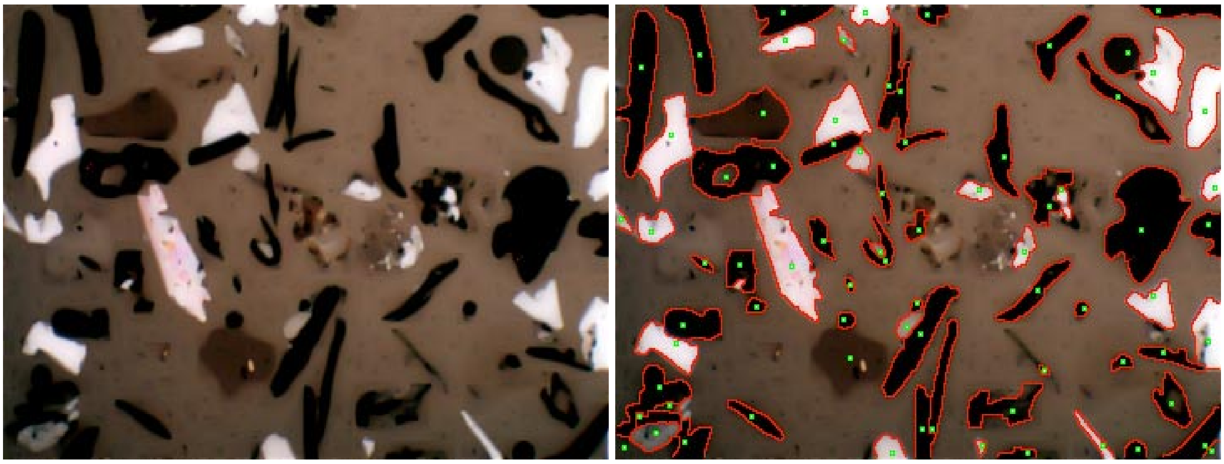


Figure 7 (a), micrograph of a geological sample. (b), ISS segmentation result (75 markers).

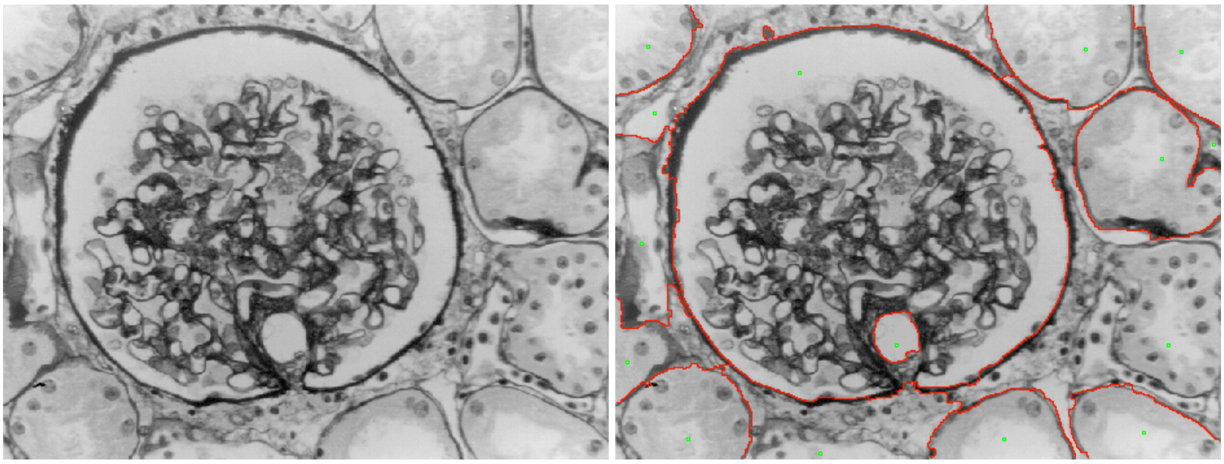


Figure 8 (a), micrograph of a glomerulus. (b), ISS segmentation result (15 markers).

Comparing the performance of image segmentation methods is not easy, since many variables are involved in the task and the methods often have different theoretical foundations. However, peculiarities of each method can be observed if they are applied to a set of images having characteristics such as irregular illumination, occlusions, reflexes, noisy or smoothed regions, sharp or diffuse edges and regions compound of more than one homogeneous regions. Figure 9 shows a set of images coming from specialized application fields such as medicine (finger x-ray and corneal endothelial cells), geology (microscopic hematite grains) as well as from ordinary scenes (peppers and flower) which present such peculiar characteristics. They have been chosen to briefly illustrate some of the problems above mentioned and how they can influence current image segmentation methods as those based on Deformable Models (Front Propagation - FP and Bubbles - BUB), Statistical Region Growing (Seeded Region Growing – SRG) and Immersion Simulation (ISS). Appearing in the first column of Figure 9 are the original non-filtered images. Second, third and fourth columns show segmentation produced by FP or BUB, SRG and ISS, respectively. Each image was segmented employing the same set of markers, with the exception of Figure 9j, which do not make use of markers. Markers appear as green squares and models - the set of points enveloping a region in evolution - as contours in red.

Homogeneous regions and sharp transitions between them often simplify the segmentation task. By comparing segmentation results in Figure 9 it becomes clear that simultaneous convergence of all models presents more difficulties to DM based methods because regions often do not present sufficient homogeneity and sharp transitions. The speed of a model depends on region homogeneity and its displacement is often delayed or even stopped by discontinuities. From the practical point of view this may result in models being pushed beyond some edges while others are still evolving, see images (b), (f) and (r). Due to stronger noise in image (f) model propagation is more difficult in than in image (n), for example. Homogeneity also plays an important role in statistical based methods like SRG, where the region growing process depends on the average intensity of each region. SRG may be trapped by the presence of more than one homogeneous sub-region inside a region-of-interest. SRG segmentation of petals image shown in image (o) illustrates this problem. Occlusion of two regions having similar intensities often lead to leaking. Leaking can be observed on the two peppers situated on the first plane in ISS segmentation image (s) and SRG segmentation (t) for peppers image and also in SRG segmented image (o). Compare segmentation results of SRG (o) to FP (n) and ISS (p). Initialisation also plays an important role in most image segmentation methods. Usually models are initialised by hand inside and/or outside the features of interest. In SRG seed size and position may change region initial average intensity thus interfering in the way model progress. Automatically initialising models at image minima or at preferential points as done by BUB simplifies the initialisation. However, simultaneous evolution of models inside and outside regions often results in double edge, see image (j). Initialisation in ISS is automatically done at image minima and because regions not having markers are blindly merged, ISS presents low sensitivity on seed size, position and noise. ISS fails if sorting do not effectively postpone the processing of pixels lying on the edges of the features of interest. Otherwise, ISS will produce segmentations of very good quality as can be observed in Figure 9.

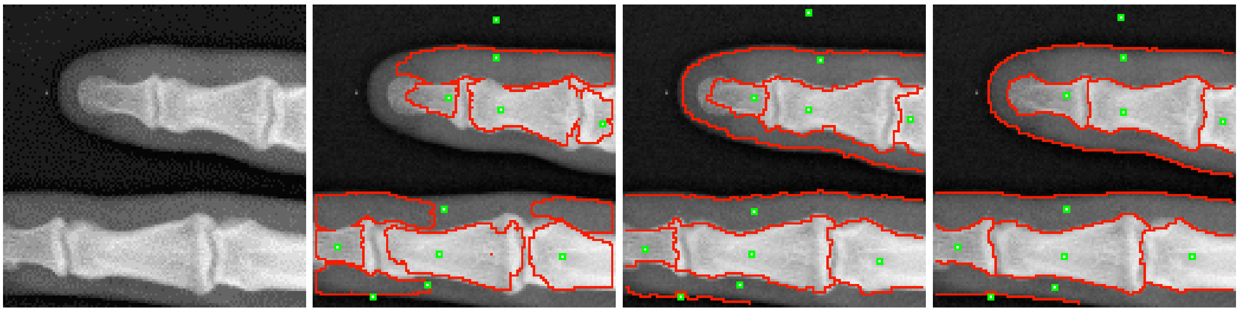
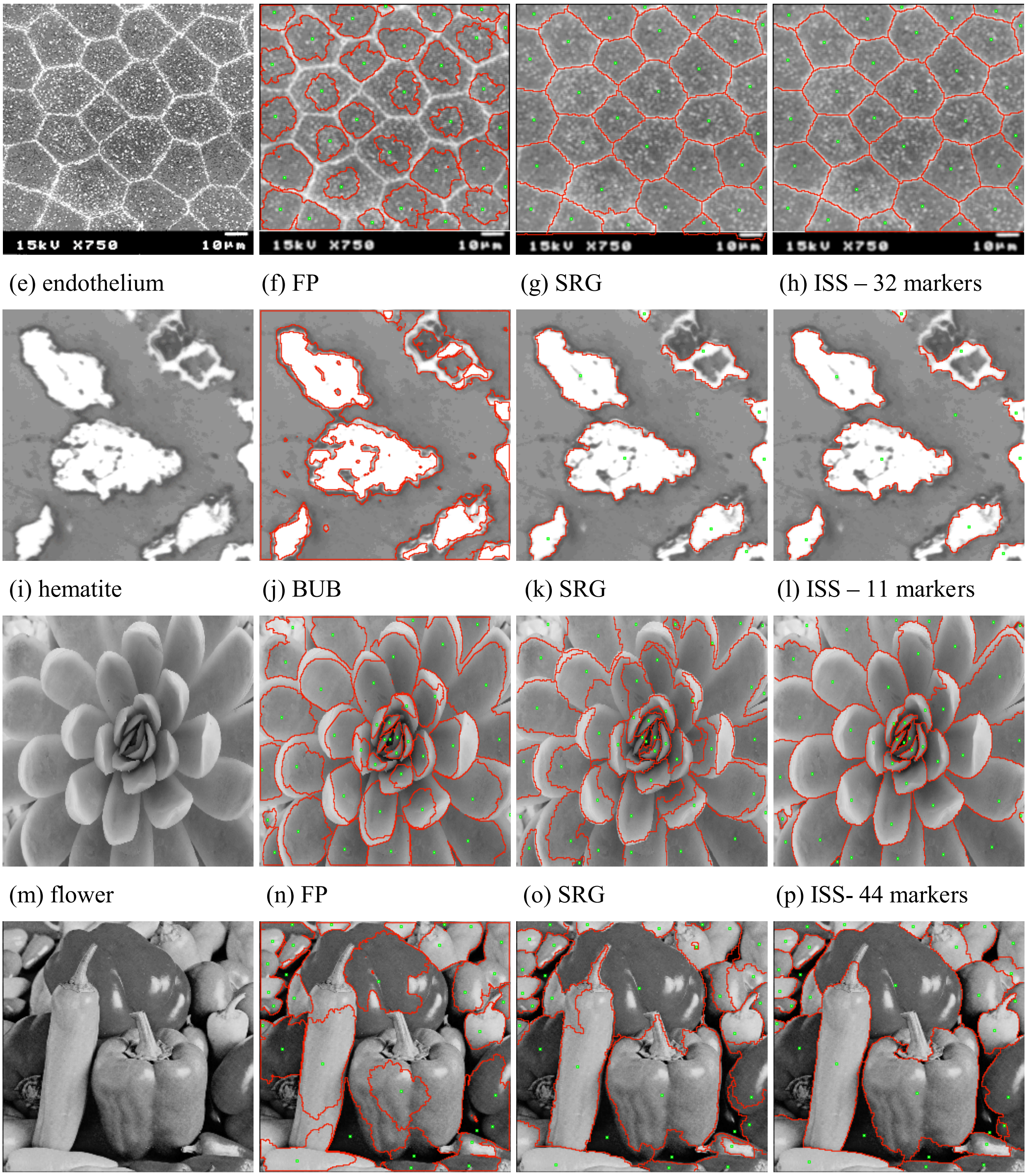

(a) finger x-ray (b) FP (c) SRG (d) ISS – 11 markers

Figure 9. Deformable Models (FP and BUB), SRG and ISS applied to different kind of image.



(e) endothelium (f) FP (g) SRG (h) ISS – 32 markers

(i) hematite (j) BUB (k) SRG (l) ISS – 11 markers

(m) flower (n) FP (o) SRG (p) ISS- 44 markers

(q) peppers (r) FP (s) SRG (t) ISS – 32 markers

Figure 9 cont. Deformable Models (FP and BUB), SRG and ISS applied to different kind of image.

4 Concluding remarks

The ISS combines some valuable features of known image smoothing and segmentation methods developed in the Mathematical Morphology and in the PDE-based level set frameworks, for instance:

- efficient edge preserving smoothing guided by PDEs, typical of surface evolution methods;
- ability to automatically detect all image minima and to make the regions grow inside the respective zones of influence, a property inherited from the *watershed transformation* (NHW);
- ability to automatically stop the growing process whenever two user labelled regions get into contact, a characteristic difficult to implement in the PDE based level set framework;
- global competition between all image pixels according to a pre-defined sorting criterion;
- ability to change the image topology by using a simple merging mechanism, thus dramatically reducing over-segmentation and the need of pre-processing;
- recovery from errors mediated by a user-guided segmentation;
- relatively low sensitivity to seed positioning;
- execution time directly proportional to image size;
- no need of tuning parameters;
- applicable to color or gray-scale in any number of dimensions.

However, ISS is not applicable where the situation requires automatic segmentation, like video segmentation. As other flooding simulation algorithms, ISS is sensitive to broken edges and may “leak” through *gaps* resulting in wrong segmentation results.

5 Acknowledgments

The author would like to acknowledge the CNPq - Conselho Nacional de Desenvolvimento Científico e Tecnológico of Brazil and FAPEMIG – Fundação de Amparo à Pesquisa de Minas Gerais, for the financial support and the CISE - Computer and Information Sciences and Engineering of the University of Florida, for the technical support

6 References

[1] R. Adams, et. al. Seeded region-growing. *IEEE Trans. Pattern Analysis and Machine Intelligence,* 16, 6, 641-647, 1994.

[2] M. C. Andrade, et. al. Segmentation of microscopic images by flooding simulation: a catchment basins merging algorithm. *Proceedings of SPIE Nonlinear Image Processing VIII,* San Jose, USA, 3026, 164—175, 1997.

[3] S. Beucher. *Segmentation d'image et morphologie mathematique.* École Nationale Supérieure de Mines de Paris, PhD thesis, 1990.

[4] S. Beucher. Watershed, hierarchical segmentation and waterfall algorithm. *Mathematical Morphology and its Applications to Image Processing,* Kluwer Academic Publishers, 69—76, 1994.

[5] F. Caselles, et al. Image selective smoothing and edge detection by nonlinear diffusion. *SIAM Journal on Numerical Analysis,* 29, 1, 183—193, 1992.

[6] J. P. Cocquerez, S. Philipp. *Analyse d' images: filtrage et segmentation.* Masson, Paris, 1995.

[7] L. D. Cohen, at. al. Finite element methods for active contours models and balloons for 2D and 3D images. *IEEE Trans. Pattern Analysis and Machine Intelligence,* 15, 1131—1147, 1993.

[8] M. Grimaud. *La geodesie numerique en morphologie mathematique. Application a la detection automatique de microcalcifications en mammographie numerique,* École Nationale Supérieure de Mines de Paris, PhD thesis, 1991

[9] M. Grimaud. A new measure of contrast: the dynamics. *Proceedings of SPIE. Image Algebra and Morphological Image Processing,* 1769, 292-305, 1992.

[10] G. Guo et. al.. Bayesian learning, global competition and unsupervised image segmentation. *Pattern Recognition Letters,* 21, 107-416, 2000.

[11] J. Isaac, et al. Sorting by Address Calculation. *Journal of the ACM*, 169—174, 1954.

[12] M. Kass, et. al. Snakes active contour models. *International Journal of Computer Vision, 1,* 321-331, 1988.

[13] R. Malladi, et al. Shape modeling with front propagating: a level set approach. *IEEE Trans. Pattern Analysis and Machine Intelligence,* 17, 2, 158—175, 1995.

[14] R. Malladi, et al. A fast level set based algorithm for topology-independent shape modeling. *Journal of Mathematical Vision,* 6, 269-289, 1996.

[15] A. Mehnert, et. al. An improved seeded region-growing algorithm. *Pattern Recognition Letters 18,* 106-1071, 1997.

[16] F. Meyer. Un algorithme optimal de ligne de partage des eaux. VIII Congrès de *Reconaissance de Forme et d'Intelligence Artificielle.* Lyon, France, 847-857, 1991.

[17] F. Meyer, S. Beucher.. Morphological segmentation. *Journal of Visual Communication and Image Representation, 1, 1, 21-46,* 1990.

[18] P. Perona, et. al. Scale-space and edge detection using anisotropic diffusion. *IEEE Trans. Pattern Analysis and Machine Intelligence. 12,* 7, 629—639, 1990.

[19] T. Sebastian, et. al. Segmentation of carpal bones from a sequence of 2D CT images using skeletally coupled deformable models. www.lem.s.brown.edu , 2000.

[20] J. A. Sethian. Tracking Interfaces with Level Sets. *American Scientist*. May-jun, 1997.

[21] J. A. Sethian Level Set Methods and Fast Marching Methods. Cambridge Press, 2nd ed, 1999.

[22] K. Siddiqi, et. al. Geometric shock-capturing ENO schemes for sub-pixel interpolation, computation and curve evolution. *Graphical Models and Image Processing*, 59, 5, 278—301, 1997.

[23] H. Tek, et. al. Volumetric segmentation of medical images by three-dimensional bubbles. *Computer Vision and Machine Understanding,* 65, 2, 246—258, 1997.

[24] C. Vachier. Extraction de Caracteristiques, Segmentation et Morphology Mathematique. École Nationale Supérieure des Mines de Paris. PhD. Thesis, 1995.

[25] L. Vincent. *Algorithmes morphologiques a base de files d'attente et de lacets. Extension aux graphes..* École Nationale Supérieure de Mines de Paris, PhD thesis, 1990.

[26] L. Vincent, P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 13, 6, 583—598, 1991.

[27] J. Weickert. *Anisotropic diffusion in image processing. B.G.* Teubner Sttutgart, Deutschland, 1998.

[28] C. Xu, J. L. Prince. Snakes, Shapes , and Gradient Vector Flow. IEEE Trans. on Image Processing. 7, 3, 359—369, 1988.

[29] A. Yezzi. Modified curvature motion for image smoothing and enhancement. *IEEE Trans. on Image Processing,* 7, 3, 345-352, 1998.

[30] S. C. Zhu, et al. Region competition: unifying snakes, region-growing, and bayes/MDL for multiband image segmentation, *IEEE Trans. Pattern Analysis and Machine Intelligence,* 18, 9, 880—900, 1996.

7 Appendix I: ISS Pseudo-code

Pseudo code for denoising gray-scale images. For RGB images apply the code below to each channel.

```
step = n;  // number of iterations.
For each step do {
 For each row do {
  For each column do {
     Compute central difference differentials dx, dy, dxx, dyy, dxy, $dx^2$ and $dy^2$
          in the neighbourhood $N(p)$ of the central pixel $p$ using floating point arithmetic;
     $p$ = (int) ($p$ + (dxx*$dy^2$ + dyy*$dx^2$ – 2*dx*dy*dxy) / (1+$dx^2$+$dy^2$)  )// slightly modifies $p$ at each step.
  }
 }
}
```

Pseudo-code for ISS segmentation.

```
MaxNumOfLabels = MNL;  // Maximum number of labels
Obs: Labels 1 to N are reserved for markers; labels from N+1 to MNL are reserved for seeds.
Initialise a merging table vector with labels 1 to MNL;
Place one marker per region-of-interest labelling them from 1 to N;
Sort all pixel in ascending order by the address calculation technique [11], according to a chosen criterion, which postpone the processing of pixels lying around the edges.
For each pixel extracted from the sorted list do {
   Find how many different positive label exist in N(p);
   If (there is no positive labelled pixel in N(p) )
      Current pixel receives a new label starting a new temporary region;
   Else if (there is only one positive labelled pixel in N(p) )
      Current pixel receives this label;
   Else if (there is 2 or more positive labelled pixels in N(p) )
      If (2 or more positive labels <= N)
          Current pixel receives a “EDGE” label;
      Else {
          Merge all neighbours into one region; the one having the smallest positive label in N(p);
          Current pixel receives this label;
      }
}
By using the merging table relable all pixel to reflect the absorption they have undergone.
```

8 Appendix II - ISS Execution time for known test-images

IMAGE	SIZE	TIME* (ms)	TIME* (ms/iteration)	TIME* (ms)
		DENOISING		SEGMENTATION
LENA	127x127	440	15	22
	256x256	1540	51	86
	512x512	5270	176	286
	1024x1024	17850	595	1098
PEPPERS	127x127	440	15	22
	256x256	1650	55	88
	512x512	5770	192	330
	1024x1024	19770	659	1154
BOAT	127x127	390	13	22
	256x256	1540	51	76
	512x512	5820	194	308
	1024x1024	20050	668	1154
* for 30 iterations, Pentium IV class machine 1.7GHz, 768MB RAM, Windows XP				

Table I. ISS execution time for standard test-images (LENA, PEPPERS and BOAT).