Image Segmentation by Tree Pruning

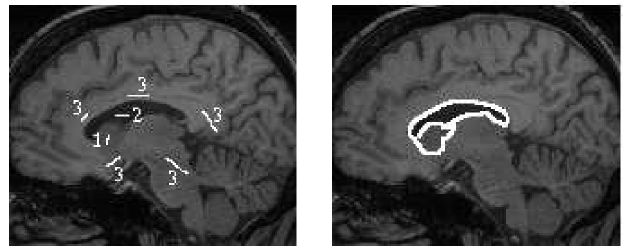
Alexandre X. Falcão, Felipe P.G. Bergo, Paulo V. Miranda
Institute of Computing – State University of Campinas (UNICAMP)
CEP 13084-851, Campinas, SP, Brazil
{afalcao,felipe.bergo,paulo.miranda}@ic.unicamp.br

Abstract

The Image Foresting Transform *(IFT) has been proposed for the design of image operators based on connectivity. The IFT reduces image processing problems into a minimum-cost path forest problem in a graph derived from the image. It has been successfully used for image filtering, segmentation, and analysis. In this work, we propose a novel image operator which solves segmentation by pruning trees of the forest. First, an IFT is applied to create an optimum-path forest whose roots are pixels selected inside a desired object. In this forest, the background consists of a few subtrees rooted at pixels on the object's boundary. These boundary pixels are identified and their subtrees are eliminated, such that the remaining forest defines the object. The tree pruning is an effective alternative to situations where image segmentation methods based on competing seeds fail. We present an interactive implementation of the tree-pruning technique, show several examples and discuss some experiments toward fully automatic segmentation.*

1. Introduction

Image segmentation methods based on competing seeds can be roughly described in three steps [1, 18, 14, 13, 15, 10]: (i) seed pixels are selected inside some objects, including background (Figure 1a), (ii) each seed defines an *influence zone* which consists of the pixels that are "more closely connected" to that seed than to any other, and (iii) each object is defined by the union of the influence zones of its internal seeds (Figure 1b). These methods present a *leaking* problem due to the absence of boundary information— a situation very common in practice. Leaking occurs when the influence zones of the internal seeds invade the influence zones of the external seeds, and vice-versa, as illustrated in Figure 2 for the watershed transform [1, 18]. Note that the identification of the leaking parts along the object boundary can be used to solve the segmentation problem.

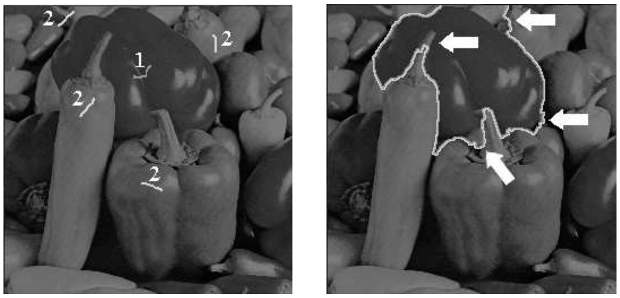


(a) (b)

Figure 1. Image segmentation by watershed transform. (a) An MR image of the brain with seed pixels selected inside the caudate nucleus (1), lateral ventricle (2), and background (3). (b) Object boundaries in the resulting segmentation.

In this work, we propose an alternative approach using the *Image Foresting Transform* (IFT)— a general tool for the design, implementation, and evaluation of image processing operators based on connectivity [3]. In the IFT, an image is interpreted as a graph whose nodes are image pixels and whose arcs are defined by an *adjacency relation* between pixels. For a given set of seed pixels inside a desired object, we define a suitable *path-cost function* and the IFT computes a minimum-cost path forest in the graph. The roots of the forest are drawn from the seed set, such that each tree consists of the pixels more closely connected to its root than to any other seed. The choice of the path-cost function intends to connect object and background by a few optimum paths which cross the object's boundary through its "weaker" parts (called *leaking pixels*). The topology of the forest is used to identify the leaking pixels and eliminate their subtrees, such that the remaining forest defines the object. The method can also be applied to multiple-object segmentation with seeds selected inside each object, excluding the background.

The applications of the IFT include watershed transformations [11, 5, 12, 2], fuzzy-connected segmentation [2], multiscale skeletonization [4], optimal boundary tracking [3, 7, 6], shape analysis using contour saliences and multiscale fractal dimension [16], morphological reconstructions [5, 3], geodesic path computation [3] and distance transforms [4]. This is the first time that the topology of the optimum-path forest computed by the IFT is exploited for region-based image segmentation.



(a) (b)

Figure 2. Leaking in image segmentation using watershed transform: (a) An image of peppers with seed pixels selected inside (1) and outside (2) the one at the top. (b) Resulting segmentation, where the errors are indicated by arrows.

Section 2 presents the tree-pruning approach for image segmentation. In Section 3, we describe its interactive implementation. Experimental results involving interactive and automatic image segmentations are presented in Section 4. We state our conclusions and discuss future work in Section 5.

2. Tree-pruning segmentation

We start from a rationale similar to that of the watershed transform [1, 18, 11]. Consider the flooding process over the topographic surface of a gradient-like image (e.g. the magnitude of the Sobel's gradient [9]), with seed pixels selected only inside a desired object, and one source of water at the location of each seed. Instead of erecting barriers wherever two bodies of water coming from distinct sources meet, let the water leak to the background through the lower pixels on the object's boundary. In practice, boundaries do not usually have the same height and these leaking pixels are not so many. That is, the streams of water reaching the background usually pass through a few leaking pixels. The identification of the leaking pixels allows to erect a barrier at their location, separating the water inside and outside the object. This process can be easily implemented using the *Image Foresting Transform* (IFT) [3].

The IFT computes a minimum-cost path forest in a graph whose nodes are image pixels and whose arcs are defined by an *adjacency relation* between pixels. We are interested here in simple connectivity relations, such as 4- and 8-neighborhood. A path $\pi = \langle p_1, p_2, \ldots, p_n \rangle$ in the graph is a sequence of distinct and adjacent pixels (p_i, p_{i+1}), $i = 1, 2, \ldots, n-1$. The cost of a path is determined by an application-specific *path-cost function*, which usually depends on local image properties along the path, such as brightness, gradient, and pixel position. For example, f_{peak} is a suitable function to simulate the aforementioned flooding process.

$$f_{peak}(\pi) \quad = \quad \max_{i=1,2,\ldots,n} \{G(p_i)\}, \qquad (1)$$

where $G(p_i)$ is the height value of the pixel p_i in a gradient-like image. The IFT assigns one minimum-cost path from the seed set to each pixel, in such a way that the union of those paths is an oriented forest, spanning the whole image. That is, each root of the forest defines an influence zone consisting of the pixels that are more closely connected to it than to any other seed. Paths of the same minimum cost can be resolved by some *tie-breaking* policy. Two tie-breaking policies are discussed in [3]. We are interested in the *first-in-first-out* (FIFO) policy that assigns any ambiguous pixel to the tree rooted at the first most closely connected seed to reach it. There are three important attributes assigned to each pixel in the forest: its predecessor in the optimum path, the cost of that path, and the corresponding root. The predecessor map represents the optimum-path forest, where trees can be pruned by turning one or more of their nodes (except the root) into single-rooted trees. We are interested in pruning trees at the leaking pixels, such that the remaining forest provides the object of interest.

Figure 3a shows, as an example, the graph of a gradient-like image G with 4-connected relation and one seed pixel (bigger dot) inside an object represented by the central basins. The resulting optimum-path forest for f_{peak} is shown in Figure 3b together with the optimum cost of each path. By counting the number of pixels in the subtrees rooted at each pixel of the predecessor map (number of descendants), we can observe in general that (Figure 3c): (i) optimum paths that reach the background have pixels with a higher number of descendants, (ii) the number of descendants decreases considerably (from 84 to 31 and 51) when these optimum paths leave the object, and (iii) the leaking pixels (such as the one with 84 descendants) are the last pixels before this fall. The segmentation can be solved by pruning the subtrees of the leaking pixels (Figure 3d), and properties (i)-(iii) are important for automatic image segmentation (Section 4).

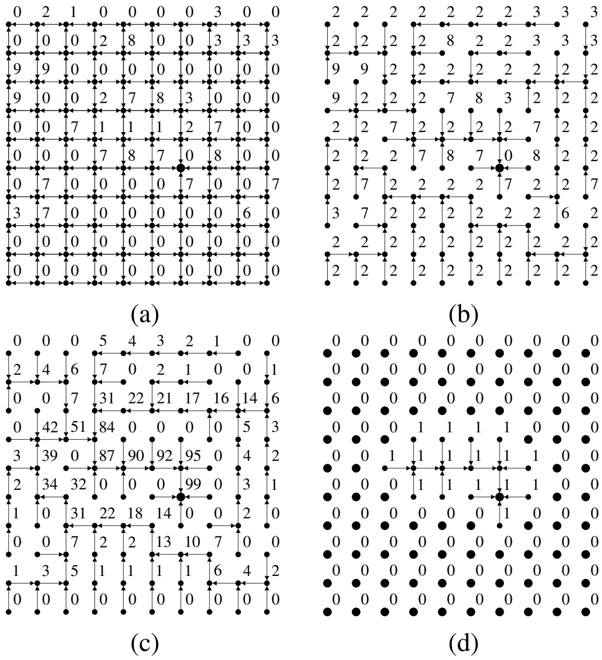


Figure 3. Segmentation by tree pruning. (a) A 4-connected graph representation of an image G with one seed pixel (bigger dot) inside the object. (b) An optimum-path forest for f_{peak}, with minimum path costs shown. (c) The number of descendants of each pixel in the forest. (d) Segmentation obtained by pruning the subtree of the pixel with 84 descendants. The numbers indicate object labels assigned to each pixel (0 for the background, 1 for the object).

More generally, we can use any path-cost function which takes into account the local dissimilarities between adjacent pixels, and perhaps, the dissimilarities between the object (as represented by the seed set) and each pixel. The whole idea is to measure a "strength of connectedness" between object and background, and disconnect them by breaking their weaker links which are expected to be across the boundary. A similar interpretation for object definition was proposed in [17]. Their method can be efficiently implemented by an IFT, where the strength of connectedness between a seed and a pixel is defined as the inverse of the cost of an optimum path from that seed to that pixel. The object is obtained by thresholding the cost map. A failure occurs when there are optimum paths reaching internal and external pixels with the same cost. The tree-pruning approach relies on another property rather than on the cost of the paths. It assumes that the optimum paths of interest, which reach internal pixels, do not pass through the background.

3. An interactive implementation

The tree-pruning approach essentially reduces the segmentation problem to the identification of seed and leaking pixels. The simplest alternative is to provide their identification in an interactive way. One can figure out user assistance in various different ways. We present an interactive implementation of the method which works as follows.

For a desired object with one or more internal seeds selected by the user, the program computes the IFT for the path-cost function f_{peak} (Equation 1), counts the number of descendants for each pixel, and displays the brightness of the original image with a colored overlay— where the color intensity of each pixel is proportional to the number of its descendants (Figure 4a). (Figure 4b shows the gradient-like image G used in Equation 1.) Since the color intensities of the optimum paths that reach the background are higher inside the object, the leaking pixels can be visually identified when these intensities are reduced across the object's boundary. The user can move the mouse over the image and the program simulates (with no noticeable delay) the pruning of each subsequent subtree, whose root is the pixel at the current position of the cursor (Figure 4c). Once a pruning point properly identifies a leaking pixel, the user can click on it to commit that pruning operation (Figure 4d). The user can make as many prunings as necessary to complete the segmentation process (Figures 4e-f).

4. Results and discussion

We observe that the background is usually more strongly connected to the object through a few leaking paths. The object can be completely disconnected from the background by breaking these paths at those leaking pixels. The method usually works even when the boundary contains long weakly defined segments (Figure 5).

Although the size of the trees is reduced with the increase of roots in the forest, the interactive approach still works in multiple object segmentation tasks (Figures 6 and 7). In this case, however, a same color is assigned to all seeds in a same object and the color of each object is propagated to the trees rooted at its seed pixels during the IFT. Note that, some objects (e.g. the ventricle[1] in Figure 6 and the squashes B and D in Figure 7) may be completely defined by simple seed selection and without any identification of leaking pixels. That is, the seed competition among distinct

1 The dark object at the center of the image.

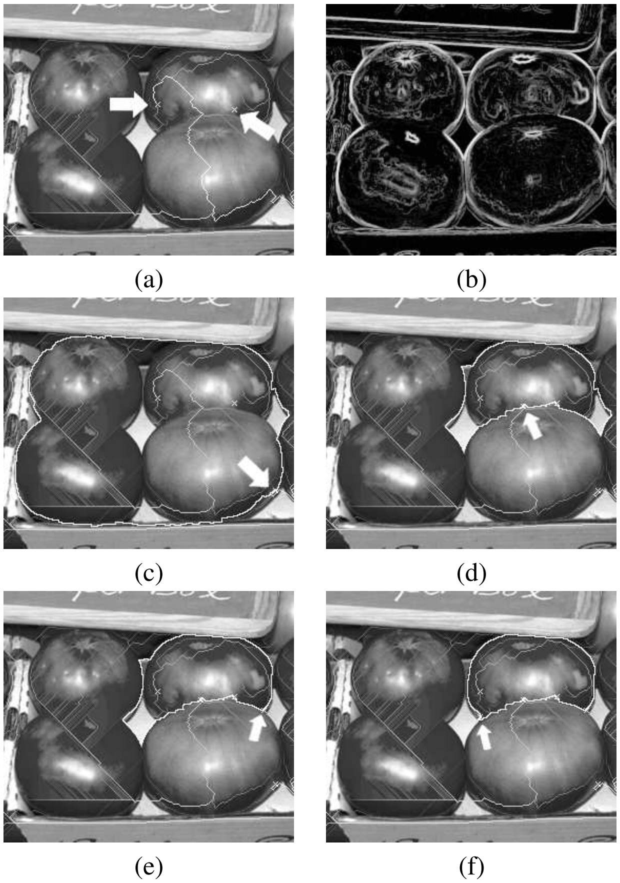


Figure 4. Interactive tree-pruning segmentation. (a) Image with the number of descendants of each pixel, after selection of two seeds indicated by arrows. (b) Gradient-like image G used in Equation 1. (c) Possible result of pruning for the current position of the cursor. (d-f) Results of committed prunings.

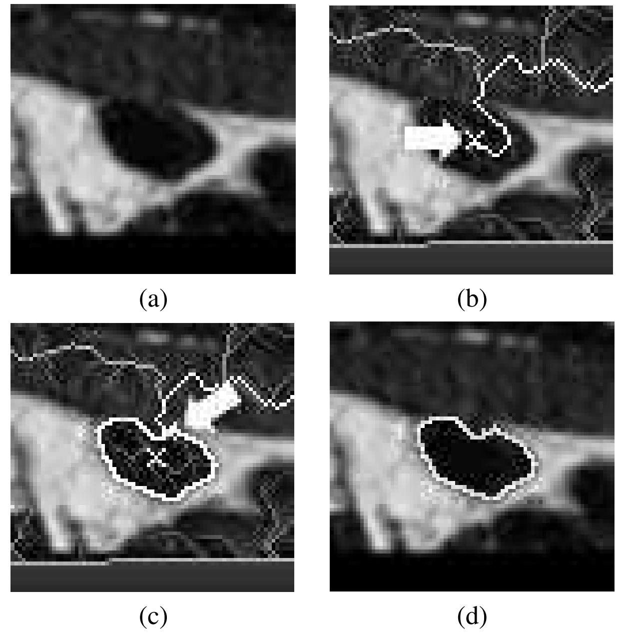


Figure 5. Tree-pruning segmentation of weakly defined boundaries. (a) Close-up of a vein in an MR image of the wrist. (b) Image with the number of descendants showing one seed (indicated by the arrow) selected inside the vein. (c) Pruning simulation for the point indicated by the arrow. (d) Result of segmentation.

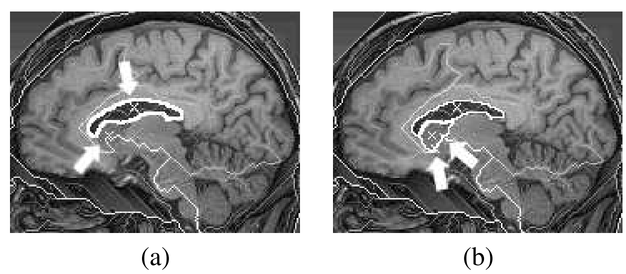


Figure 6. Multiple object segmentation by tree pruning. (a) An MR image of the brain with the number of descendants, after selection of one seed in the lateral ventricle and one seed in the caudate nucleus. (b) Result of two prunings indicated by arrows.

objects reduces the number of leaking pixels (i.e. user involvement). In theory, the leaking pixels should be completely eliminated with seeds in the background, in which case the method would be a watershed transform. However, this does not occur in several situations (e.g. Figure 2). In fact, it is not difficult to find examples such as that, where the tree-pruning approach is simpler and more effective than region growing methods based on competing seeds (Figure 8).

Note that an entire segmentation process consists of two tasks [8]: *recognition* and *delineation*. Recognition consists of roughly determining "where" the object is and distinguishing it from other object-like entities. Delineation consists of precisely defining the spatial extent of the object region in the image. Human operator (application experts) usually outperform computer algorithms in most recogni-

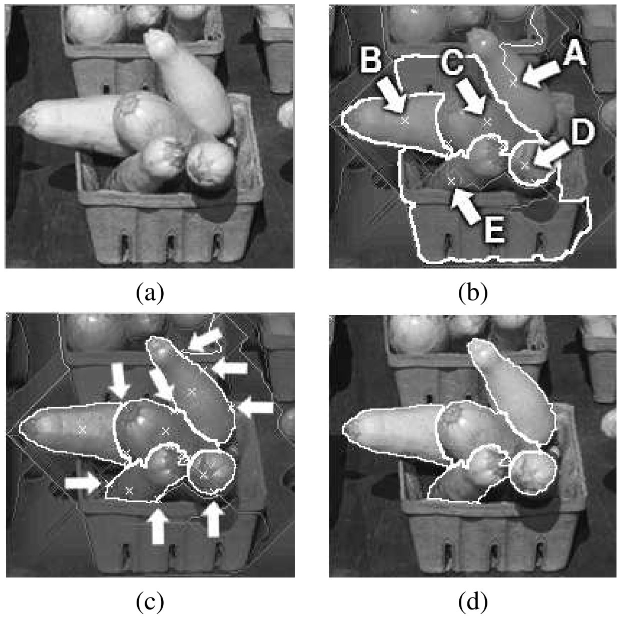


Figure 7. Multiple object segmentation by tree pruning. (a) An image of a basket with squashes. (b) Image with the number of descendants, after seed selection for each squash. (c) Result of a few prunings indicated by arrows. (d) Resulting segmentation.

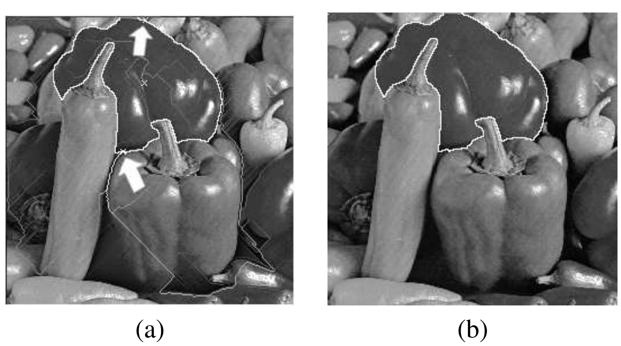


Figure 8. Tree pruning outperforming the watershed transform for the example shown in Figure 2. (a) An image of peppers with the number of descendants, after seed selection for the one at the top and two prunings (indicated by arrows). (b) Resulting segmentation.

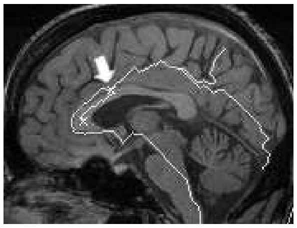
tion tasks, and the other way around can be verified in delineation tasks. In the interactive tree-pruning approach, recognition is performed by the user, when the user selects seed and leaking pixels and verifies the pruning results, while delineation is automatically performed by the IFT algorithm. This strategy has been shown to be very effective in exploiting the synergy between human operator and computer algorithms during interactive segmentation [2, 7, 6, 8].

The inability to translate relevant global object-related knowledge needed for recognition to computable local operations has been one of the major obstacles to automatic segmentation. We believe that it is possible to make seed selection, tree pruning and verification automatically, in some specific applications. Some knowledge about local image properties of the object can be used to estimate seed pixels, and automatic verification can be performed by matching between global properties of the object and each possible remaining forest, considering several pruning simulations. The properties of the forest described in Section 2 can be exploited to identify the pruning points. Figure 9a illustrates, as an example, an image after seed selection showing only the number of descendants greather than 10% of the total number of pixels in the image. Observe that this correctly isolates the leaking path and some of its branches outside the object. Due to the FIFO policy, the leaking path splits into several branches at a leaking pixel such that the number of its descendants is the sum of the number of pixels along these branches. Therefore, we can expect abrupt variations in the number of descendants along the leaking path. These variations can be observed for one of the branches in Figure 9c— the number of descendants along the path is plotted following the predecessor pixels from the end to the beginning (root pixel) of the path. In fact, they are better detected as zero-crossing transitions of the second derivative of the curve (Figure 9d), where the most significant one occurs at the leaking pixel. When we prune the subtree rooted at this leaking pixel, the remaining tree defines the object (Figure 9b). The same process must be applied to each leaking path in the case of multiple leaking pixels.

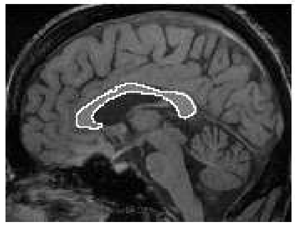
Figure 10 illustrates another example of automatic pruning for the detection of license plates, with the same strategy used in the example of Figure 9. The segmentation of the plates is a crucial step to the posterior analysis of its content. Note that the automatic tree pruning seems to be feasible for any internal seed location, and it occurs at the same points (Figure 11).

5. Conclusions and future work

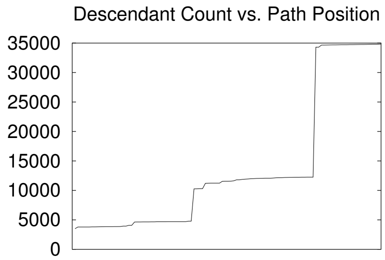
We introduced a new image operator based on the IFT and its application to image segmentation. We presented an interactive implementation of the method and several results using grayscale and colored input images. In both cases, the IFT of a gradient-like image is computed and the brightness of the original image is displayed with the number of



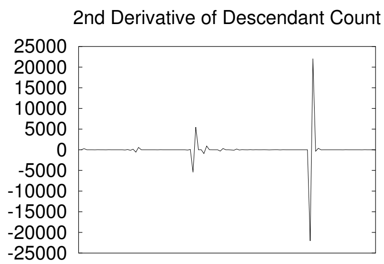
(a)



(b)



(c)



(d)

Figure 9. Example of segmentation with automatic pruning selection of the Callossum in an MR image of the brain. (a) The 10%-threshold path (gray lines) and the pruning point detected automatically (indicated by the arrow). (b) Resulting segmentation by tree pruning at the automatically-detected pruning point. (c) A plot of the number of descendants along one of the considered 10%-threshold paths. (d) The second derivative of (c), used to select the most likely points for pruning.

Figure 10. License plate segmentation by tree pruning. One seed is selected inside each plate, and the first pruning point detected automatically (indicated by the arrows) segments the plate.

descendants of each pixel in the forest. We discussed interactive and automatic approaches of pruning based on such information. The results indicate that tree pruning is a very promissing technique, being efficient and effective in many practical situations; That is, the number of seeds and pruning points are usually not prohibitive. This minimizes user interventions in interactive segmentation and favors automatic object definition in specific applications.

We are currently investigating procedures to identify leaking and seed pixels, extending the tree-pruning approach to multidimensional segmentation, and evaluating the method in some real applications.

Acknowledgements

The authors thank Prof. Roberto Lotufo (FEEC-UNICAMP) for the images of license plates. This work was supported by CNPq (Procs. 302966/02-1 and 472139/03-7), CAPES and FAPESP (Proc. 03/09793-1).

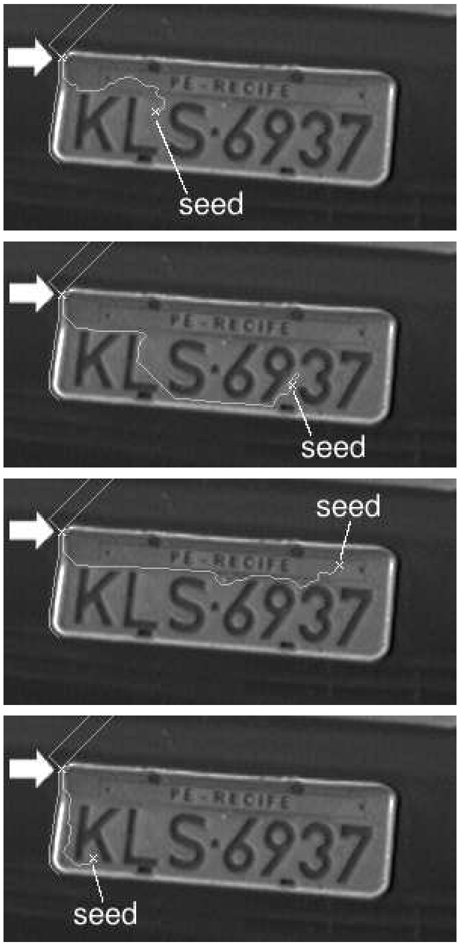


Figure 11. Segmentation of the same license plate with different seed locations: the automatically detected leaking point (indicated by the arrows) is always the same, since it is the stronger connection between the object and the background.

References

[1] S. Beucher and F. Meyer. The morphological approach to segmentation: The watershed transformation. In *Mathematical Morphology in Image Processing*, chapter 12, pages 433–481. Marcel Dekker, 1993.

[2] A. Falcão and F. Bergo. Interactive volume segmentation with differential image foresting transforms. *IEEE Trans. on Medical Imaging*, 2004. to appear.

[3] A. Falcão, J. Stolfi, and R. Lotufo. The image foresting transform: Theory, algorithms, and applications. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 26(1):19–29, 2004.

[4] A. X. Falcão, L. F. Costa, and B. S. da Cunha. Multiscale skeletons by image foresting transform and its applications to neuromorphometry. *Pattern Recognition*, 35(7):1571–1582, Apr 2002.

[5] A. X. Falcão, B. S. da Cunha, and R. A. Lotufo. Design of connected operators using the image foresting transform. In *Proc. of SPIE on Medical Imaging*, volume 4322, pages 468–479, Feb 2001.

[6] A. X. Falcão and J. K. Udupa. A 3D generalization of user-steered live wire segmentation. *Medical Imaging Analysis*, 4(4):389–402, Dec 2000.

[7] A. X. Falcão, J. K. Udupa, and F. K. Miyazawa. An ultra-fast user-steered image segmentation paradigm: Live-wire-on-the-fly. *IEEE Trans. on Medical Imaging*, 19(1):55–62, Jan 2000.

[8] A. X. Falcão, J. K. Udupa, S. Samarasekera, S. Sharma, B. E. Hirsch, and R. A. Lotufo. User-steered image segmentation paradigms: Live-wire and live-lane. *Graphical Models and Image Processing*, 60(4):233–260, Jul 1998.

[9] R. Gonzales and R. Woods. *Digital Image Processing*. Addison-Wesley, 1992.

[10] G. Herman and B. Carvalho. Multiseeded segmentation using fuzzy connectedness. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 23(5):460–474, 2001.

[11] R. A. Lotufo and A. X. Falcão. The ordered queue and the optimality of the watershed approaches. In *Mathematical Morphology and its Applications to Image and Signal Processing*, volume 18, pages 341–350. Kluwer, Jun 2000.

[12] R. A. Lotufo, A. X. Falcão, and F. Zampirolli. IFT-Watershed from gray-scale marker. In *Proc. of XV Brazilian Symp. on Computer Graphics and Image Processing*, pages 146–152. IEEE, Oct 2002.

[13] L. B. R. Adams. Seeded region growing. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 16(6):641–647, 1994.

[14] J. B. T. M. Roerdink and A. Meijster. The watershed transform: Definitions, algorithms and parallelization strategies. *Fundamenta Informaticae*, 41:187–228, 2000.

[15] P. K. Saha and J. K. Udupa. Relative fuzzy connectedness among multiple objects: theory, algorithms, and applications in image segmentation. *Computer Vision and Image Understanding*, 82:42–56, 2001.

[16] R. Torres, A. Falcão, and L. Costa. A graph-based approach for multiscale shape analysis. *Pattern Recognition*, 37(6):1163–1174, 2004. to appear.

[17] J. K. Udupa and S. Samarasekera. Fuzzy connectedness and object definition: theory, algorithms, and applications in image segmentation. *Graphical Models and Image Processing*, 58:246–261, 1996.

[18] L. Vincent and P. Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 13(6), Jun 1991.