Image Segmentation by Water-Inflow

Stephan Fischer
Praktische Informatik IV, University of Mannheim
D-68131 Mannheim, Germany
fisch@pi4.informatik.uni-mannheim.de
phone: +49-621-292-1407 fax: +49-621-292-5745

December 30, 1996

Abstract

This paper describes an image segmentation technique based on water-inflow. We show how images can be segmented by a mere distinction between water or ground using a filling process. We explain how the rising water level can be used to create object hierarchies. These hierarchies cannot be achieved by many other algorithms. We also compare the results of our algorithm with those of other segmentation techniques.

Keywords: Image processing, image retrieval, image segmentation, medicine imaging

1 Introduction

In recent years many researchers tried to solve the problem of an automatic image segmentation for many different applications like medicine imaging, military observation, optical character recognition or alert systems([15, 6, 7, 14, 11, 9, 2]). Most of these ideas are not suitable to segment an image satisfactorily. Many researchers even think that the problem is either unsolvable or that only a combination of many different approaches might yield the desired results([8, 1]). It is still not known very well how the human perception process works and how human experience helps in image understanding([17]). The most promising approach might be to use a segmentation process in combination with an expert system and motion information included in digital video. The basis for such a system is a still image segmentation which is as good as possible.

Among the great variety of known algorithms different directions of research have evolved the following segmentation algorithms:

- Edge-oriented approaches which try to segment images based on local color differences([3, 4, 10])

- Region-based approaches which try to find images by region growing([13, 16])
- Energy-based methods
- Neural net based approaches([8])

These strategies can be reduced to two main approaches: a local analysis approach defined by objects encircled with borders and a global analysis approach defined as prominent signal variations. These two approaches have to be combined to obtain a segmented image of good quality([8]).

The algorithm to be described in this paper integrates both approaches. The principal idea is to see a grey-level image as a kind of ground relief. The image is filled with water stepwise. In each step a new image consisting of water and ground is created and segmented. We will show that this procedure outperforms the algorithms we tested. Furthermore it allows us to gain depth information which is lost during the mapping of 3D-real-world images to 2D-digitized images.

The remainder of this article is structured as follows. Section 2 describes in depth the segmentation algorithm. Section 3 compares the water-inflow algorithm with others. Section 4 reports on experiments and shows segmentation results compared with other segmentation strategies. Section 5 concludes the paper.

2 The Segmentation Algorithm

The principal idea of the algorithm is to fill the image with water stepwise. The grey-values of the pixels determine the height of the image. The higher the water rises, the more pixels are under water. In each step we obtain water regions and ground regions. It is then easy to determine objects separated by water. In a second step the objects segmented in the different phases of the algorithm have to be associated. This allows the construction of a hierarchy tree representing the development of the objects.

2.1 Segmentation on a single water level

The preparation for this step is the conversion of the grey color image to a binary image. If the value of the pixel color is greater than the water level, the pixel value of the binary image is 1, otherwise 0.

After the conversion we apply a non-recursive region growing step. The recursive one is more elegant but most of today's computers are not able to work with stack heights of more than 2000. This value is reached in almost any image. The region growing algorithm we use is based on an 8-connectivity search. All pixels which are connected to the start pixel and which are not water become part of the object. The algorithm uses the stack operations PUSH, POP and EMPTY. The filling algorithm for a single object is called with a start pixel (x,y) and works as follows:

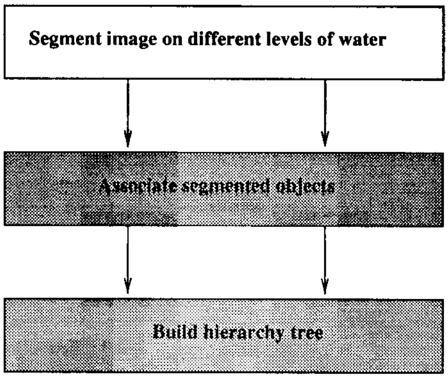

Figure 1: Steps of the segmentation algorithm

```
PUSH (x,y)
while not EMPTY()
    POP(x,y)
    mark (x,y) as object part and as visited
    find right border (x,y) of object
    if pixel at right upper side is unvisited and 1 and
      if pixel is inside image bounds
           PUSH(y-1,x+1)
    if pixel at right lower side is unvisited and 1 and
      if pixel is inside image bounds
           PUSH(y+1,x+1)

    while left border of object has not been reached
        x=x-1
        examine pixel above current pixel
        if it is unvisited and part of the object
            PUSH(y-1,x)
        examine pixel below current pixel
        if it is unvisited and part of the object
            PUSH(y+1,x)

    if pixel at left upper side is unvisited and 1 and
      if pixel is inside image bounds
            PUSH(y-1,x-1)
    if pixel at left lower side is unvisited and 1 and
      if pixel is inside image bounds
            PUSH(y+1,x-1)
```

The filling is repeated with an increased object number until all pixels have been marked as visited. To avoid redundancies the algorithm does not push a pixel above or below the current position if the one right to the pixel above or below has already been pushed. The result of the filling at a specific level is an image which contains objects separated by water.

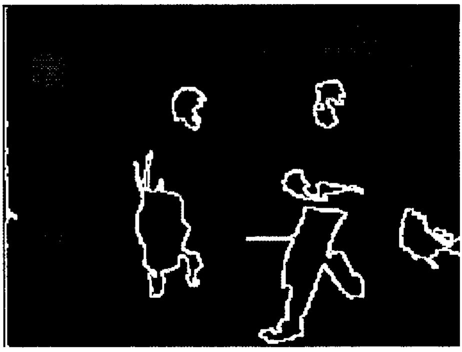

Figure 2: Result image for water level 30

Parts covered by water are not examined. This is a guarantee for the algorithm to become faster at higher water levels where greater parts of the image are already filled with water. Figure 2 shows the segmentation result for level 30. In contrast to the final segmented image(see chapter 3) only parts of the objects are visible.

2.2 Combination of the levels

After the segmentation on a specific level of water $\omega-1$ the level is increased and the procedure repeated. The result of the segmentation of level ω has to be combined with the result of the level before. An important condition therefore is that objects cannot increase in size. At a higher water-level more and more pixels disappear and therefore objects become smaller. In the association process of an object of level $\omega - 1$ with one of level ω two cases have to be considered:

- The object becomes smaller because object pixels are under water. The difference between the smaller object and the greater is less than a threshold λ. A variation of λ determines the granularity of the resulting segmented image.
- The object becomes smaller because it splits up into new objects. The difference between the smaller object and the greater is greater than λ.

If an object splitting has been found the values of level ω are copied to the final result image, else the final result image is not changed. Small objects of a size less than a threshold α are not copied. A regulation of α also determines the granularity of the result image.

The process of splitting an object into smaller objects raises another problem. It may happen that the new objects do not cover the old object completely. To avoid the occurrence of small rests of the old object the rests are associated to the new objects using a nearest neighbor search.

2.3 Construction of hierarchy graphs

The use of the algorithm yields a set of segmented images, one for each level of water. The construction of a hierarchy graph to be described now offers a great advantage: depth information which usually gets lost in the mapping of a 3D-real-world image to a 2D-image can be partially recovered. Imagine an arm in front of a human body. While most algorithms divide the body into three parts - one for the arm and two for the rest of the body, our algorithm offers the possibility to recover the depth information using the hierarchy graph. This graph stores the development of the objects during the splitting process. It is therefore possible to retrieve an object in another image using different granularities. The body, for example can be retrieved in its smallest parts or as a bigger conglomerate of small objects. This is stored in the hierarchy graph.

The hierarchy graph is a tree. The root is the whole image. Each time an object is divided into new objects, new nodes are inserted into the tree, one for each new object. Each of the nodes stores the level of water where it was created and a seed pixel. The development of the objects can then be reconstructed by finding a path from the root to the object node.

3 Comparison with other approaches

Comparing the water-inflow approach with edge-based techniques, the advantage of the new approach is that it is a global approach, not being restricted to edges. The edge-linking problem does not occur. On the other hand, local information as the steepness of the gradient and thus the strength of edges is not exploited.

Global approaches like the split-and-merge approach have the problem of finding a good similarity measure to form the regions. The use of a color criterion results in a loss of object parts containing ramps and vice versa. The water-inflow approach outperforms these techniques.

Neural network-based approaches are quite similar to the water-inflow approach in terms of calculating a set of segmented images which are combined by neural networks. A combination of the different water levels using neural networks is currently examined.

4 Experiments

We implemented a prototype on a DEC Alpha workstation to test the performance of the water-inflow algorithm.

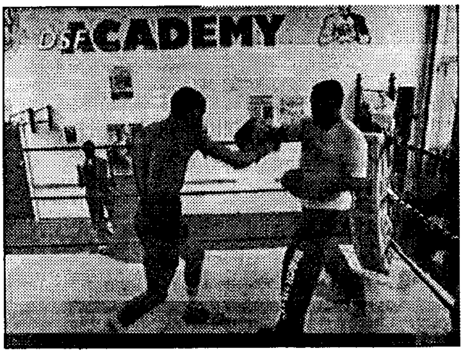

Figure 3: Original image 1

Figure 4: Canny image 1

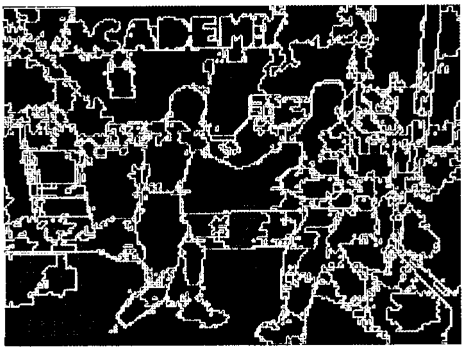

Figure 5: Image 1 segmented with neural networks

Figure 6: Image 1 segmented with region growing algorithm

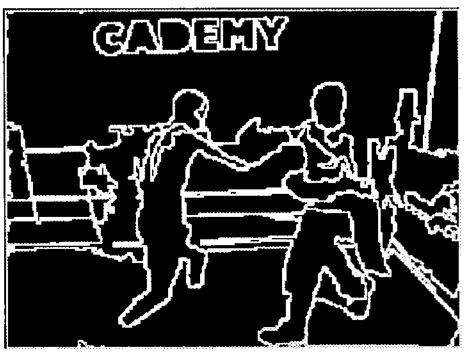

Figure 7: Image 1 segmented with water-inflow algorithm

Figure 8: Original image 2

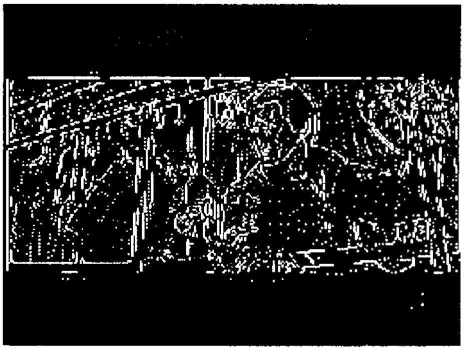

Figure 9: Canny image 2

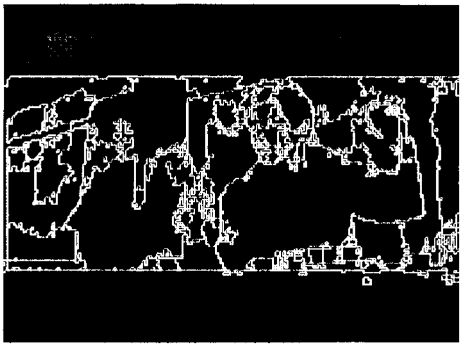

Figure 10: Image 2 segmented with neural networks

Figure 11: Image 2 segmented with region growing algorithm

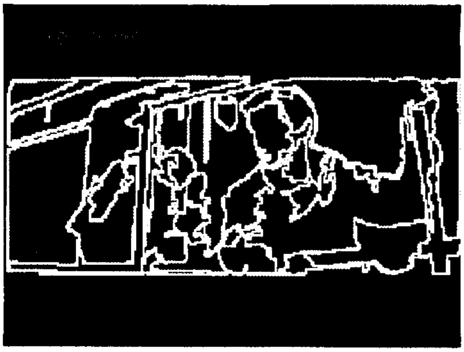

Figure 12: Image 2 segmented with water-inflow algorithm

To be able to compare the performance of the water-inflow algorithm with that of other algorithms we chose the following algorithms:

- An edge-based approach(Canny algorithm([3])
- A neural network based approach([8])
- A region oriented approach([12]).

We chose the default settings of the standard deviation σ and the fraction of pixels considered edge noise δ. These are 1.0 for σ and 0.3 for δ. Table 1

Algorithm	Time in percent
Water-inflow computation time	100.00
Canny computation time	79.23
Neural network computation time	178.3
Region oriented approach computation time	211.2

Table 1: Computation time for algorithms

shows the complexity of the algorithms in terms of computation time. The time needed by the water-inflow algorithm has been set to 100 percent. The application of the neural network algorithm was only possible using a WWW-page. As the given percentage includes the network transmission delay, the segmentation time should be less than 178.3 percent. The

Parameter	Value Range
λ	0.8 - 0.9
Minimal object size(α)	400 pixels
Maximal number of objects	60

Table 2: Parameter settings of the water-inflow algorithm

parameter settings for the water-inflow algorithm(see Section 2.2) can be found in Table 2.

As it is our goal to retrieve humans in digital images[5] we do not need a fine segmentation granularity. Therefore we chose the values 400 for the minimal object size and a value of 0.85 for λ. A greater value of *lamda* yields more objects since the object 1 is associated to the one of the level before(2) only if the percentage 1/2 is greater than λ.

Our experiments showed that it is not necessary to regard each level of water. In our tests it turned out that it is sufficient to increase the water level by a value of 10 to yield satisfactory results. Comparing the different segmented images it turns out that the water-inflow algorithm outperforms the examined algorithms. Considering the fact, that the Canny algorithm can only be used in combination with an edge linking algorithm the water-inflow algorithm is also faster than the other algorithms.

5 Conclusions and Outlook

In this paper we have described a new image segmentation technique, the image segmentation based on water-inflow. We showed how an image of grey colors is converted to a binary image for each water level. We explained the combination of the segmented images of the different levels yielding a single result image. Therefore we described the decision rules to split an object found in water level $(n-1)$ into more objects in water level n.

It turns out that the described segmentation technique outperforms the algorithms we examined. An advantage is, that the resulting image consists of objects. Therefore it does not have to be processed anymore to find objects by edge-linking like in the Canny case. As the examination of binary images is fast the algorithm is not too slow although it is not working in real-time.

Future expansion of our work will include examination of object hierarchies. Many other algorithms, for example, do not recognize the unity of a human body if an arm is in front. We currently examine the recognition of objects based on a comparison of their hierarchy graphs. Furthermore we will compare the water-inflow algorithm to other algorithms described in literature. We also have to examine more details of our algorithm. We currently do not know the increase rate of the water-level which still yields sufficient results. We also work on the deletion of line segments. These may connect different objects and lead to wrong results. Another important question is if the integration of water-flow processes yields better results.

6 Acknowledgments

I would like to thank Silvia Pfeiffer, Rainer Lienhart and Prof. Dr. Wolfgang Effelsberg for their help in developing ideas which yielded the described system.

References

[1] Sing-Tze Bow. *Pattern Recognition and Image Preprocessing.* Marcel Dekker, Inc., New York, NY, 1992.

[2] L. Gottesfeld Brown. A survey of image registration techniques. *ACM Computing Surveys*, 24(4), 1992.

[3] J. Canny. A computational approach to edge detection. In *IEEE Trans. on Pattern Anal. and Mach. Intell., Vol. PAMI-VIII, No. 6*, November 1986.

[4] J. S. Chen and G. Medioni. Detection, localization, and estimation of edges. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-11(2):191–198, February 1989.

[5] S. Fischer. Automatic violence detection in digital movies. In *Multimedia Storage and Archiving Systems, Proc. SPIE 2916-22*, 1996.

[6] Rafael C. Gonzales and Richard E. Woods. *Digital Image Processing.* Addison Wesley Publishing Company, Inc., 1993.

[7] John F. Haddon and James F. Boyce. Image segmentation by unifying region and boundary information. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(10):929–948, October 1990.

[8] Rolf D. Henkel. Segmentation in scale space. In *6th Int. Conf. on Computer Analysis of Images and Pattern(CAIP)*, 1995.

[9] B. Jähne. *Digital Image Processing.* Springer-Verlag, 1993.

[10] H. Kollnig and H.-H. Nagel. 3d pose estimation by fitting image gradients directly to polyhedral models. In *Proceedings of the Fifth International Conference on Computer Vision*, pages 569–574, 1995.

[11] R. Lienhart and F. Stuber. Automatic text recognition in digital videos. In *Image and Video Processing IV, Proc. SPIE 2666-20*, 1996.

[12] R. Mester. *Region oriented image segmentation based on stochastic image models(In German).* Fortschritt-Berichte VDI Reihe 10, Nr. 106, 1989.

[13] Theo Pavlidis and Yuh-Tay Liow. Integrating region growing and edge detection. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-12(3):225–233, March 1990.

[14] Todd R. Reed and J. M. Hans Du Buf. A review of recent texture segmentation and feature extraction techniques. *CVGIP: Image understanding*, 57:359–372, May 1993.

[15] Azriel Rosenfeld and Avinash C. Kak. *Digital Picture Processing.* ACADEMIC PRESS INC., London, 1982.

[16] Luc Vincent and Pierre Soille. Watersheds in digital spaces: An efficient algorithm based on immersion simulations. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, PAMI-13(6):583–598, June 1991.

[17] Harry Wechsler, editor. *Neural Networks for Perception : Computation, Learning, and Architecture.* Academic Press, 1991.