Easy-to-Use Object Selection by Color Space Projections and Watershed Segmentation

Per Holting and Carolina Wählby

Centre for Image Analysis, Uppsala University, Sweden
carolina@cb.uu.se
http://www.cb.uu.se

Abstract. Digital cameras are gaining in popularity, and not only experts in image analysis, but also the average users, show a growing interest in image processing. Many different kinds of software for image processing offer tools for object selection, or segmentation, but most of them require expertise knowledge, or leave too little freedom in expressing the desired segmentation. This paper presents an easy to use tool for object segmentation in color images. The amount of user interaction is minimized, and no tuning parameters are needed. The method is based on the watershed segmentation algorithm, combined with seeding information given by the user, and color space projections for optimized object edge detection. The presented method can successfully segment objects in most types of color images.

1 Introduction

Object segmentation embrace extracting an object from a non-trivial environment. To be of use for the non-expert, a segmentation tool should be easy to use with as little user interaction as possible. To delineate an object in an arbitrary color image is not trivial. Due to the infinite variation in possible image conditions, no assumptions can be made. The tool has to maximize the use of the information given by the user to try to understand what object is wanted. A Magic Wand is a typical selection tool implemented in many different image-processing softwares, e.g. [1]. Magic Wands are based on pixel similarity in color space, where the tolerated variance is given by the user as a tuning parameter. The tuning parameter is difficult to choose accurately, and repeated interaction is required for segmentation of multi-colored objects, and exact object edges can be difficult to find. The widely spread image processing software Photoshop [1] provides an Extract Tool which works quite well but requires that the user marks the approximate object edge. This may be very time consuming for irregularly shaped objects. Another interactive object selection tool is GrabCut [2]. This tool combines color and edge information, and provides a post-processing step where the edges are smoothed by matting.

1.1 The Presented Approach

The aim of the presented project was to achieve a high performance algorithm at the cost of only modest interaction by the user, yet maximizing the use of

F. Roli and S. Vitulano (Eds.): ICIAP 2005, LNCS 3617, pp. 269–276, 2005.

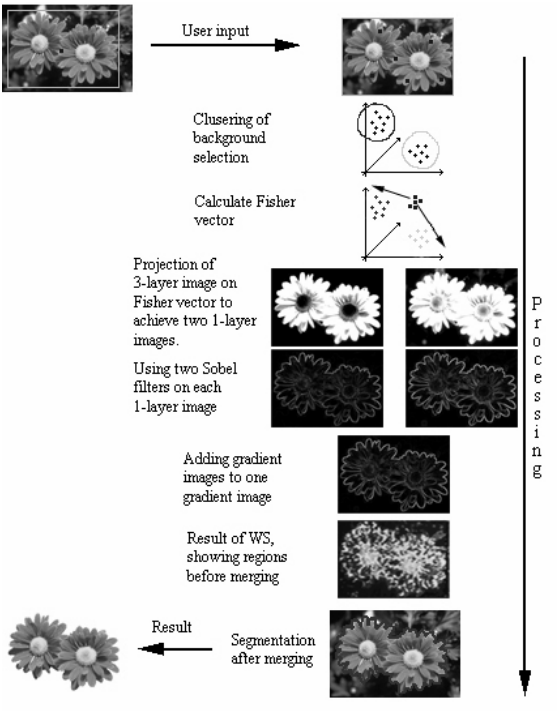

Fig. 1. An overview of the presented algorithm. The gray rectangle and the black dots represent the user input. Each of the steps is described in the method section.

this input. The algorithm is described graphically in Fig. 1. Starting with a color image in the RGB-space the user draws a rectangle around the object to define the background, and a point inside the object to define the object. The rectangle and the point are called the background seed and the object seed respectively. As the background may contain varying colors and textures, the pixels belonging to the background seed are clustered into two clusters. Two projection vectors are thereafter calculated to find the projection that best differentiates between the two background clusters and the object cluster. The gradient magnitudes of the two projected images are summed, and seeded watershed segmentation is applied to the gradient magnitude, using the user input as seeds. The algorithm will find the positions between the background seed and the object seed where the gradient magnitudes are the largest. If the initial result is not satisfying, the user can add more seed points, both background and object seeds, and run the algorithm a second time.

2 Methods

2.1 Image Gradient in Gray-Scale Images

Segmentation algorithms for monochrome images are generally based on one of two basic properties of image intensity values: discontinuity and similarity [3].

Discontinuities in an image are detected by using approximations of first and second derivatives. The first order derivative in a digital context is the gradient where the gradient amplitude is the local edge strength. The derivative of a digital function is often defined in terms of differences. There is a range of ways to define these differences but they all have to fulfill some conditions. The derivative has to be zero in flat, homogenous regions, non-zero at the onset of a grey-level step or ramp and non-zero along ramps. To achieve some smoothing effects and reduce the influence of noise it is appropriate to approximate the gradient operator using a mask consisting of a central pixel and a well defined neighborhood, where each position holds a weight. The Sobel edge detector [3] corresponds to a high pass filter, preserving rapid intensity changes and suppressing small variations and constant regions. A 3 by 3 pixel Sobel edge detector, one for vertical and one for horizontal edges, was used in the presented method.

The gradient calculation discussed above is, unfortunately, not defined for color images, as color images are vector quantities. One approach to this problem is to calculate the gradient for the RGB components separately and add the results [3]. A more sophisticated method is proposed by Di Zenzo [4]. This method calculates the maximum rate of change and lets it influence the final gradient result. We propose a method that optimizes gradient calculation to fit the segmentation seeds given by the user, by combining clustering and Fishers discriminant function.

2.2 Color Clustering

Before reducing the color information, the seeds representing the image background are clustered into two clusters. As the image background may vary on different sides of the same object, clustering before projection and gradient calculation improves the local edge information. More than two clusters would probably improve the result, but two clusters was chosen as a good tradeoff between performance and speed. The k-means algorithm [5] is an exclusive clustering algorithm meaning that data are grouped in exclusive clusters and a certain data point belonging to a defined cluster can only be included in this cluster and no other. The k-means clustering algorithm was used in the presented method, and it follows a simple and easy way to classify a given data set into a given number of clusters, k, fixed a priori. The main idea is to define k centroids, one for each cluster, and associate each data point with a centroid. In the presented approach, the background data is divided into two clusters, and the origin together with the data point at the greates Euclidean distance away from the origin are chosen as initial centroids. Thereafter, k new centroids are found and the data points, again, are associated with the nearest centroids. This loop makes the centroids change their location until no more data points are moved. Doing this, the algorithm aims to minimize an objective function, here the squared error function.

2.3 Fisher's Linear Discriminant

To be able to compute the image gradient with the Sobel filters the 3-layer (RGB-space) image has to be projected onto a 1-layer image. This can be done using

Fishers linear discriminant [6], which will find the projection that maximizes the difference between each of the two background clusters and the foreground cluster, keeping the internal cluster variance small. Data projected on an arbitrary line will often produce a confused mixture of samples from different clusters, but by moving the line around we can find an orientation for which the data is more or less well separated. This way, the dimension can be reduced while the class information is preserved. By calculating the Fisher vector for object seeds and background cluster 1 and for object seeds and background cluster 2, the 3-D problem is converted to a 1-D problem where the difference between the object seed and the two clusters from the background seed is maximized with respect to the projected means and standard deviations. A large difference between object pixel values and background pixel values is an advantage when trying to find edges between the object and the background. It is not trivial to combine the gradients in a way that gives a good segmentation result. The presented approach using Fishers linear discriminant results in a single image where the contrast between object and background is maximized. This means that the gradient separating object from background will also be maximized.

2.4 Watershed Segmentation

A general segmentation algorithm, known as watershed segmentation (WS), was originally presented by Beucher and Lantuèjoul [7]. This method can be applied to different kinds of image information, such as grey-level, distance or gradient information, to divide the image into regions. The method has been refined and used in many situations [8,9]. The WS algorithm can be explained by considering the image as a landscape, where high intensity values in the image correspond to mountains in the landscape, and low values correspond to valleys. Picture drilling a small hole at every local minima, and slowly submerging the landscape into water. The deepest valleys will start to fill with water, and as the water rises, water from different valleys will, in time, meet. At places where water from different valleys is about to meet, a watershed is built to avoid the water to merge. When the whole landscape has been submerged into water, all pixels in the image have been assigned to a region. The WS algorithm is commonly applied to the image gradient magnitude. WS can be implemented by using sorted pixel lists [10] so that essentially only one pass through the image is required. WS often results in many more (or fewer) regions than desired, i.e., over-segmentation or under-segmentation. This can be handled in many different ways, for example by seeding, marking the regions of interest. Seeds can be planted manually or automatically. In the presented method, the user puts seeds in the image background and in the desired object. In addition to the regions represented by the object and background seeds, the initial WS leads to one region per unseeded local minima. Reduction to only two regions is achieved by running a merging algorithm. The merging algorithm merges all non-seeded regions where the difference between the local minima and the corresponding gradient magnitude is the smallest, and the final segmentation will contain object and background regions only.

3 Results

The presented method was tested on hundreds of images, and some of the results are shown below. More results can be viewed at `http://www.cb.uu.se/~carolina/objectselection/`. The results depend on the user input, and thus, the user can influence the result of the segmentation. An unsatisfactory result can also be improved by additional input of seeds by the user. The described method of creating a gradient image by color clustering and projection outperformed other methods for color image gradients when searching for the best gradient image for the desired segmentation. When the object consists of several different areas of color or texture our algorithm needs more than the initial seeds. For relatively simple objects, segmentation can be performed with only one object seed, and no extra user interaction is needed. Fig. 2 shows an exam-

Fig. 2. The girl is the desired object in the image to the left. If a single object seed (grey star on the girl's dress, center image) is used together with a large outer seed, only the girl's dress and hair will be found. By adding a few extra seeds on the girl's legs, hands and chest, the full girl can be extracted (right).

Fig. 3. The edge of the fur of the birds is difficult to find, yet a fairly satisfactory result is achieved by just a single foreground and background seed

ple where a single object seed is not sufficient as the object consists of two very different colors (i.e., the black dress and the pale skin). A satisfactory result is achieved by adding a few more seed points.

Thin smoke, fur and hair in the boundary of objects is not trivial to segment, see Fig. 3. One way of improving the visual result at this type of transparent transitions between object and background is to use a matting technique [2],

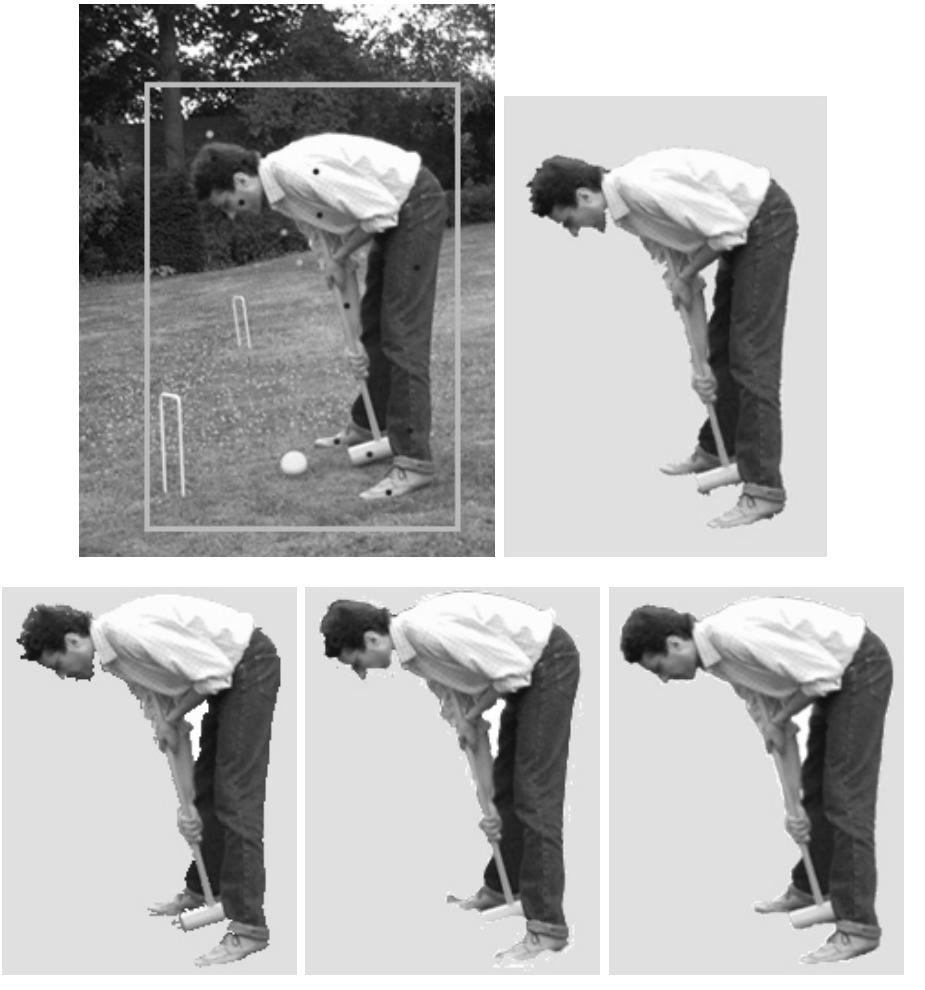

Fig. 4. The croquet player is the desired object in the image in the top left. The result of the presented method, together with input seeds in the top left, is shown to the top right. Below, from left to right, are the results of Magic Wand, Extract Tool, and Grab Cut. Both Magic Wand and Extract Tool need far more user interaction than the presented method. The amount of input needed for GrabCut is not clear, but most likely similar to that of the presented method.

where border-line pixels are given a transparency value based on their similarity to object or background.

Comparing the presented method with other methods shows that our result often is at least as good but with less user input, see Fig. 4. The most difficult areas in this image are at the boundary of the persons head and arm because of a soft transition between object and background, but by adding more seeds a satisfying result is achieved. The Magic Wand requires hundreds of mouse-clicks by the user, even when choosing a proper tolerance. There is also a risk of obtaining small holes within the object due to local color variations. The Extract Tool also manages to segment the object but it requires a careful and time consuming manual tracing of the approximate object boundaries. The presented method is outperformed by the GrabCut algorithm, but this visually more satisfying result is partly achieved by a matting step after the initial segmentation. The result of the GrabCut comes from Microsoft research GrabCut homepage [11], and it is unclear how much user input was required to achieve this result. All test images come from the Berkeley image database [12].

The runtime for the presented method depends much on the size of the processed object, determined by the background seed. For a selection of size 100*250 pixels the algorithm takes about one second and for a larger selection of 200*450 pixels it takes about 10 seconds. The processing time is also dependent on image texture, and can be reduced by smoothing the image before processing, at the price of a lower edge precision. The larger selection (200*450 pixels) with a smoothed image takes about 2 seconds to process. If the computation time is not crucial, the result may be improved by dividing the background seeds into more than two clusters, determined by a clustering algorithm that creates clusters with a specified maximum internal scatter.

4 Conclusions and Future Developments

This paper presents an interactive segmentation tool where foreground and background seeds applied by the user are used for optimized gradient detection and seeded watershed segmentation. The presented method can segment an object in most arbitrary images. In some situations only a rectangle defining background and a single object seed is needed to get a satisfying result. When the object consists of many different regions more user input is needed. For objects with smooth edges, the desired boundary can be found by placing background and object seeds on both sides if the boundary. Highly textured objects are often a problem and much user input is needed, and sometimes no satisfying result can be achieved. For objects with hair at the boundary, the result could probably be improved by post-processing using matting. Taken together, the method is sufficient for object segmentation and does not have any tuning parameters, making it easy to use and appropriate for a general user without expertise knowledge.

Acknowledgements. The authors would like to thank Révolte Development AB who initialized and supported this project.

References

1. Adobe Photoshop version 7.0
2. Rother, C., Kolmogorov, V.,and Blake, A.: GrabCut Interactive Foreground Extraction using Iterated Graph Cuts. ACM Transactions on Graphics **23** 309–314 (2004)
3. Gonzales, R. and Woods, R.: Digital Image Processing, Prentice Hall 2nd edition (2002)
4. Di Zenzo, S.: A Note on the Gradient of a Multi-Image. Computer Vision, Graphics and Image Processing **33**(1) (1986)
5. Tou, J. T. and Gonzalez, R.C.: Pattern Recognition Principles, chapter 3.3.5 Addison-Wesley Publishing Company (1974)
6. Duda, R.O. and Hart P.E.: Pattern Classification and Scene Analysis, chapter 4.10 Wiley-Interscience (1973)
7. Beucher, S. and Lantuèjoul, C.: Use of watershed on contour detection. International Workshop on Image Processing: Real-time and Motion Detection/Estimation, Rennes, France (1979)
8. Meyer, F. Beucher, S.: Morphological segmentation. Journal of Visual Communication and Image Representation, **1**(1) 21–46 (1990)
9. Vincent, L.: Morphological grayscale reconstruction in image analysis: Applications and efficient algorithms. IEEE Transactions on Image Processing, **2**,2 176–201 (1993)
10. Vincent, L. and Soille, P.: Watersheds in Digital Spaced: An Efficient Algorithm Based on Immersion Simulations. IEEE Transactions on Pattern Analysis and Machine Intelligence. **13**(6) 583–598 (1991)
11. Microsoft research GrabCut homepage, http://research.microsoft.com/vision/cambridge/segmentation/
12. Berkeley Image database, http://www.cs.berkeley.edu/projects/vision/grouping/segbench/BSDS300-images.tgz