8th Conference on Informatics and Information Technology with International Participation (CIIT 2011)

IMAGE SEGMENTATION USING BILATERAL FILTER

Ivana Lj. Shopovska
Faculty of Electrical Engineering
and InformationTechnologies
Skopje, Macedonia

Zoran A. Ivanovski
Faculty of Electrical Engineering
and Information Technologies
Skopje, Macedonia

ABSTRACT

This paper presents a simple segmentation method based on K-means clustering procedure and a preprocessing step. The first step applies a bilateral filter for smoothing the image in the regions of low color variations that would enhance segmentation. After smoothing, the image is segmented by K-means clustering. We compare the bilateral filter to other image smoothing filters and show its advantages in its use in image segmentation.

I. INTRODUCTION

Image segmentation is an important and challenging task in image analysis and in many computer vision, image interpretation, scene understanding and pattern recognition systems, with applications in scientific and industrial fields such as medicine, remote sensing, microscopy, content based image and video retrieval, document analysis, industrial automation and quality control. Semantic segmentation can be said to simulate the cognitive task performed by the human visual system to decide what people see, and relies on a priori assumptions. One main reason for this is that human vision despite low level, is based also on high level prior knowledge about the semantic meaning of the objects that compose the image. The performance of color segmentation may significantly affect the quality of an image understanding system. The most common features used in image segmentation include texture, shape, grey level intensity, and color. The choice of the right data space is a common problem in the segmentation and classification methods. In order to construct realistic classifiers, the features that are sufficiently representative of the physical process must be searched.

Image segmentation techniques can be divided into the following basic concepts: pixel oriented, contour oriented, region oriented, model oriented, color oriented and hybrid.

Edge preserving filtering is usually used in medical, satellite and aerial image processing, primarily as denoising methods [1, 2, 3].

As a segmentation method, k-means and mean-shift clustering are widely used. Mean shift segmentation is getting more popular, however, the basic mean-shift is more time consuming than the k-means algorithm.

The paper is organized as follows: in Section II we discuss about what a “good” segmentation is and what visual cues are relevant for segmentation. We also compare some simple preprocessing methods that would improve segmentation. In Section III we discuss about some edge-preserving smoothing methods and give our choice of an edge-preserving method. Section IV explains the bilateral filter as an edge-preserving filter used in our algorithm and discusses about its effect in smoothing and in segmentation. In Section V the segmentation process is explained. Section VI shows the results compared to some other segmentation algorithms.

II. IMAGE SEGMENTATION PREPROCESSING

Image segmentation is the process of grouping pixels of similar features in several groups (sets, segments). The features for comparison and the similarity metrics used vary depending of the application. Segmentation of natural images usually requires that the segments obtained correspond to the intuitive human perception of segments. However, achieving perceptually meaningful segmentation is a very challenging, maybe impossible task.

The assumption we made is that usually objects or parts of objects in the scene are made from one material and therefore they are homogeneous in color. Therefore, high color variations would mean boundaries between objects. Accordingly, different segments would mean different objects or parts from objects. We also assume that intuitively it is much safer to divide one object into several segments instead of joining multiple objects into one segment. It is also possible to use texture as characteristic assuming that one object is homogeneous in texture. However, distinguishing objects by color is more natural to human perception.

The main idea in this algorithm is to achieve relatively fast and perceptually meaningful color image segmentation. We want to join pixels of high color similarity into several segments, but still, correctly to preserve meaningful region boundaries. Our approach is based only on color similarity.

We will briefly analyze and compare some of the simplest methods of feature extraction for the purpose of segmentation.

The simplest approach is to take into consideration the color values of every pixel in the image without any preprocessing (Fig. 1. first row, left column). In the segmentation process, edges and details would be preserved, but there would exist segments consisted of pixels of similar color that are (spatially) scattered. Additionally, noisy pixels would erroneously affect segmentation. Of course, these are unsatisfactory results (Fig. 1. first row, right column).

Another approach is to smooth the image before segmentation, for example, to give each pixel average value of the pixels in its neighborhood. This would result in smooth regions of lower color variations than the original image (Fig. 1. second row, left column), which would enhance segmentation in terms of obtaining more contiguous segments. However, no details would be preserved, edges would be blurred, and the correct region boundaries would be unrecognizable. This method would often result in grouping only the blurred edge pixels into one segment image (Fig. 1.

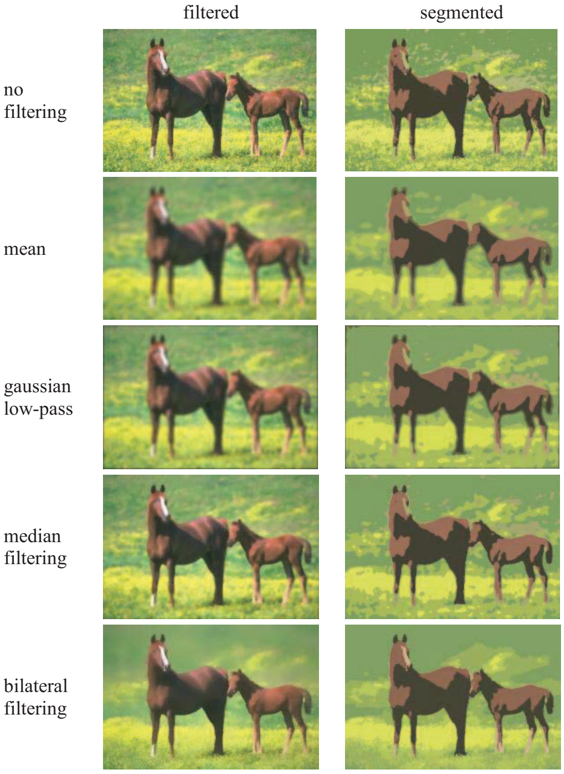

Figure 1. Comparison of different smoothing methods

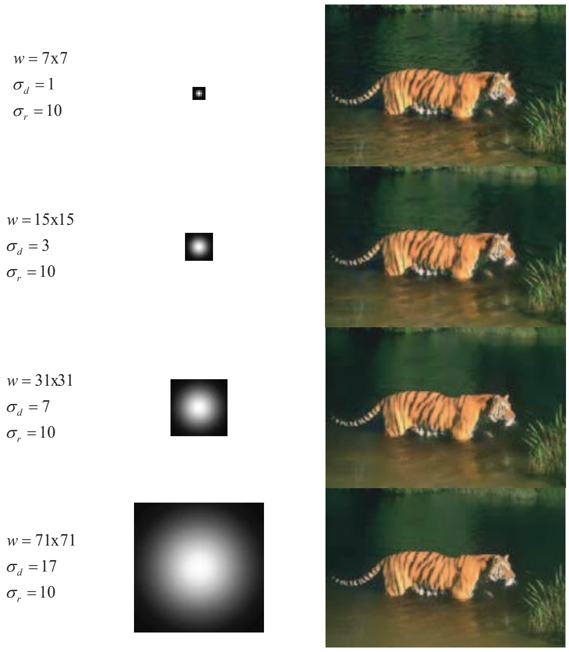

Figure 2. Bilateral filter applied with different spatial filter parameters w and σ_d

second row, right column). This type of segments is not semantically relevant.

Third possible preprocessing approach that would give better results than the both mentioned above is to assign each pixel a value that would be a weighted average of the pixels in its neighborhood. To better preserve pixel color, but still to obtain some color averaging, a Gaussian filter is one possible solution to the problem. (Fig. 1. third row, left column). Gaussian low-pass filtering computes a weighted average of pixel values in the neighborhood, in which, the weights decrease with distance from the neighborhood center. The assumption of slow spatial variations fails at edges, which are consequently blurred by low-pass filtering. This method improves the results, but is still sensitive to large color differences between pixels. This filter preserves details only for small values for the variance. As it can be seen on Fig. 1. (third row, right column) it improves segmentation compared to the previous two methods but still edge pixels are grouped into one segment.

It is also possible to use the median instead of the mean value. In median filtering, variance in neighboring values does not influence the result as much as the mean. This method preserves some details, but still blurs the edges. It is good at removing strong noise and is usually used as a noise reduction filter (Fig. 1. fourth row).

Best smoothing results would be achieved if we were able to locally analyze pixel neighborhood and only use those pixels that are close to and similar to the central pixel.

Closeness refers to vicinity in the spatial domain, similarity to vicinity in the range. Traditional filtering is domain filtering, and enforces closeness by weighing pixel values with coefficients that decrease with distance. Similarly, range filtering averages image values with weights that decrease with dissimilarity. Range filters are nonlinear because their weights depend on image intensity or color. Therefore, some edge-preserving smoothing filter would be much more suitable for feature extraction.

III. Edge Preserving Algorithms

Some examples of edge-preserving filters are [4]:

- Anisotropic diffusion
- Symmetrical Nearest Neighbor Filter (SNN)
- Maximum Homogeneity Neighbor Filter (MHN)
- Conditional Averaging Filter

These non linear algorithms are calculating the filtered gray value in dependence of the content of a defined neighborhood. From all of the neighborhood pixels, only those which have similar gray values compared to the pixel in consideration, are taken for the averaging. Each edge-preserving filter has its own specific algorithm, but they all have in common, that the effect of this smoothing strategy is to preserve edges. Unfortunately, these smoothing filters have the characteristic not to smooth satisfyingly, because small gray value fluctuations existing in the really homogeneous areas are emphasized and not reduced. In addition, the Symmetrical Nearest Neighbor Filter is unable to produce reliable results in case of small areas. They are usually used as noise reduction filters and are calculated on only one color channel.

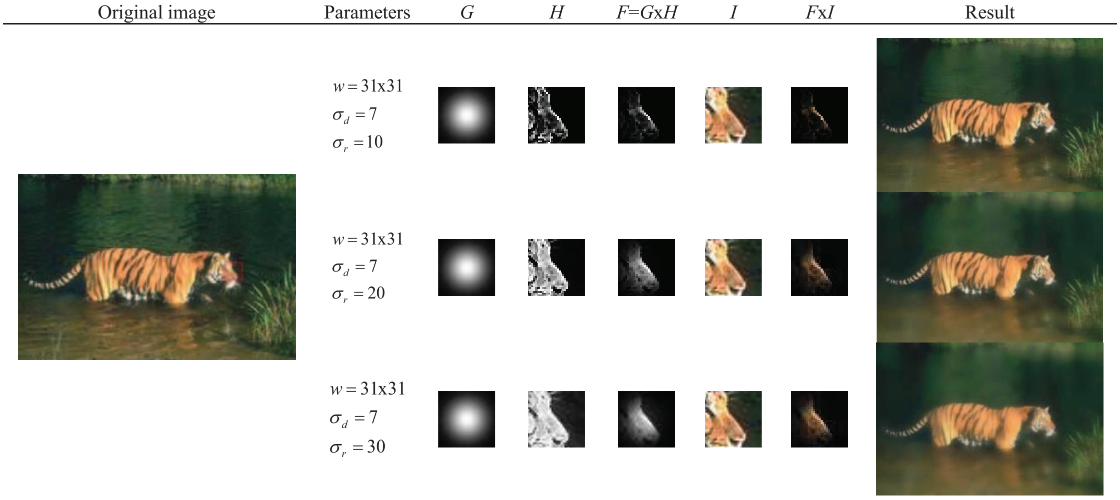

Figure 3. Bilateral filter applied with different range filter parameter σ_r

We present a method of image segmentation based on bilateral filtering. Bilateral filter is both domain and range filter. For the purpose of segmentation, the bilateral filter is of great use, since only perceptually similar colors are averaged together, and only perceptually visible edges are preserved. Bilateral filtering is of higher computational complexity than the previously mentioned approaches, however, since our algorithm is relatively simple, this additional complexity is acceptable, and the results achieved are highly improved (Fig. 1. fifth row).

IV. The Bilateral Filter

Bilateral filter was proposed by Weule (1995), Smith and Brady (1997) and Tomasi and Manduchi [5] (1998) for the purpose of image smoothing. It has also been widely used for image denoising, relighting and texture manipulation, dynamic range compression, illumination correction and image enhancement. It has also been adapted to other domains such as mesh fairing, volumetric denoising, optical flow and motion estimation and video processing.

Bilateral filter smooths the image while preserving edges by means of nonlinear combination of nearby pixel values. It combines gray levels or colors based on both their geometric closeness and their color similarity. Every pixel in the image is given a value that is a weighted average of the pixels in its vicinity. Bilateral filter is the combination of two Gaussian filters: a spatial one, and a range one. The spatial Gaussian acts like the normal Gaussian filter in assigning larger weights to nearby pixels and smaller weights to distant pixels. The main idea is to give more weight to the pixels that are closer in the spatial, but also in the range domain, to the current central pixel. This allows preserving of edges and significant details, while smoothing areas of slow color variation. The method is noniterative, local and relatively simple, thereby achieving satisfying results with only a single pass. Nonetheless, solutions have been proposed to speed up the evaluation of the bilateral filter.

The CIE-Lab color space has the property of having perceptually meaningful color similarity when the similarity metric used is the Euclidean distance. In contrast with filters that operate on the three bands of a color image separately, a bilateral filter can enforce the perceptual metric of the CIE-Lab color space, and smooth colors and preserve edges in a way that is tuned to human perception. Also, in contrast with standard filtering, bilateral filtering produces no artifacts along edges in color images, and reduces artifacts where they appear in the original image.

The bilateral filter is defined by the following equations:

The spatial domain Gaussian distance weight function:

$$G(x,y) = e^{-\frac{x^2+y^2}{2\sigma_d^2}} \qquad (1)$$

Where x and y are the distances between the center and neighboring pixels in the two directions, σ_d^2 is the variance in the spatial domain and the mean μ is zero.

The range domain Gaussian distance weight function:

$$H(dL,da,db) = e^{-\frac{dL^2+da^2+db^2}{2\sigma_r^2}} \qquad (2)$$

Where dL, da and db are the three color channels differences between the center and neighboring pixels, σ_r^2 is the variance in the range domain and the mean μ is zero. These two Gaussians are combined to form the bilateral filter that it is then applied over every pixel in the image.

The effects of changing the parameters of both functions are shown on Fig. 2. and Fig. 3. applied on a 640x320 image. Fig. 2 shows the effect of changing the parameters of the spatial domain Gaussian function while keeping the range parameter constant. We can notice that the bigger the support of the Gaussian (w is the window size), the stronger the blurring of the regions of slow color variations, but the details in all four cases are well preserved because of the low range variance parameter σ_r in all examples. Fig. 3. shows the

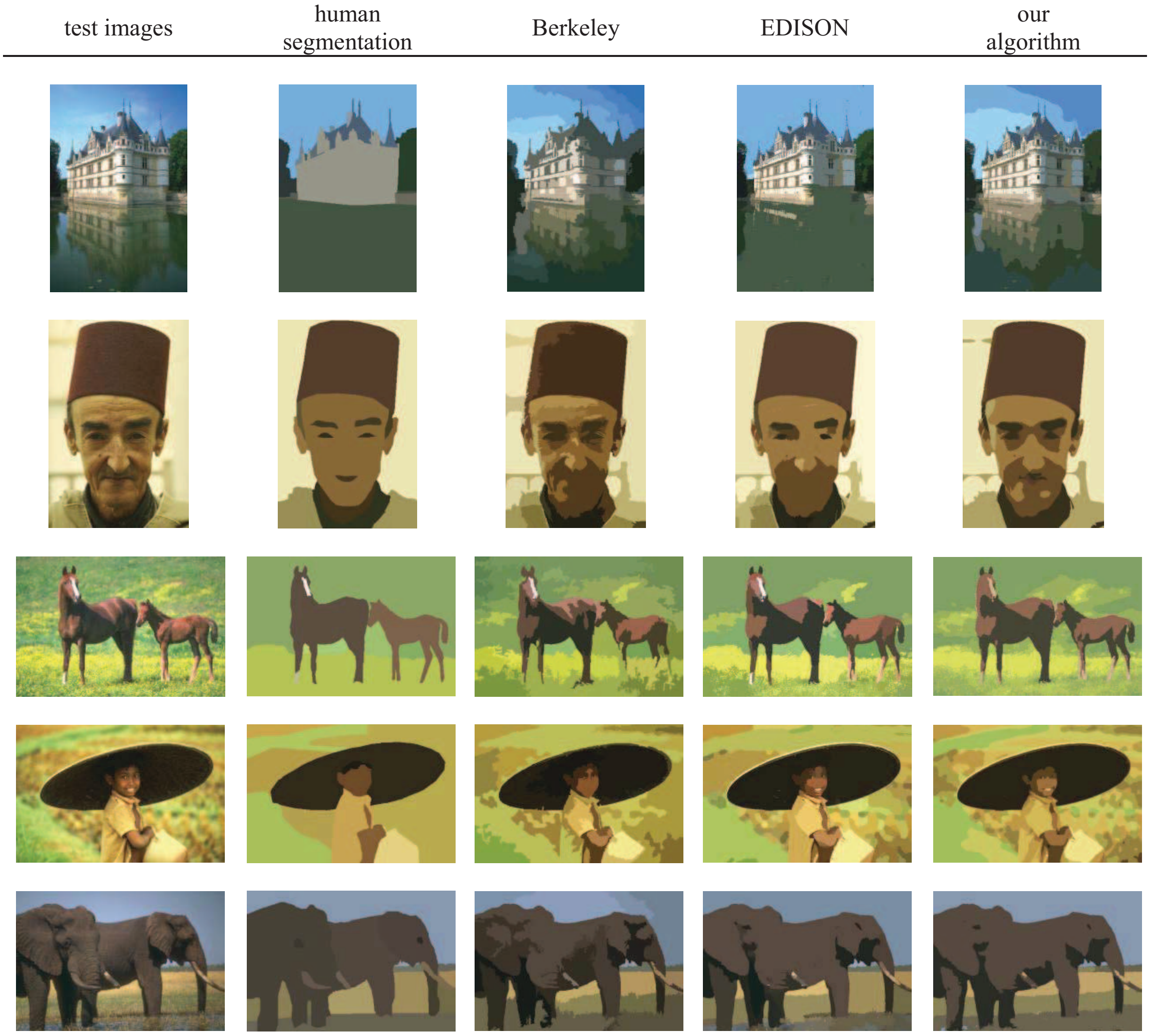

Figure 4. Comparison of our algorithm with other segmentation algorithms

effect of changing the range parameter σ_r with constant spatial range Gaussian. We can notice that increasing σ_r allows higher color vairance of pixels to be averaged. As can be seen in the example, increasing this parameter results in more blurred image, especially noticeable in the details. When the color variance σ_r is high enough, the bilateral filter acts like a Gaussian low-pass filter.

Tuning these parameters is easy and it is easy to achieve satisfying results, although the choice of the parameters is best to be adjusted to every image separately and according to the needs.

V. Image Segmentation Using the Bilateral Filter

After filtering with the bilateral filter, a feature descriptor is extracted for the image. In our algorithm, as mentioned before, the feature vector extracted for every pixel consists only from the three color components of the pixel, in the Lab color space. Using more complex descriptor, every feature added would increase the processing time and make segments more correct but less contiguous and less similar to human perception.

Image segmentation is carried by k-means clustering of the descriptor. This method iteratively partitions data in a predefined number of clusters, K, and the algorithm stops when clusters converge or after a predefined number of iterations.

Because K-means is a local searching procedure, the final cluster centers are dependent on the choice of initial centers. Having well chosen initial centers is an important part of the clustering. To find good initial centers we used the technique described in [6] that partitions the data set along the data axis with the highest variance. The main idea is to find a partitioning point on the data axis with the highest variance that would minimize the total clustering error.

As a measure of similarity, the Euclidean distance was used.

VI. Results and discussion

Some of the results from our algorithm are shown on Fig. 4. In the first column the test images are given, on which we applied segmentation. The images were taken from the Berkeley hand-labeled segmentations set, which is a subset from the Corel dataset [7]. The segmentations are averaged annotations from 30 human subjects. We compare our results to these hand-labeled segmentations, the online Berkeley Image Segmentation (whose main goal are satellite and aerial images) [8], and the Edge Detection and Image Segmentation (EDISON) System [9]. By examining the hand segmented images we can see that the segmentation does not correspond to grouping only by low-level similarity. People also use high-level knowledge in the segmentation process, which is still unachievable even with the most sophisticated methods. Of course, with a simple low-level segmentation algorithm we cannot expect to achieve similar results.

The EDISON system incorporates Confidence Based Edge Detector, Mean Shift Based Image Segmenter and optionally Synergistic Image Segmenter. The results of this system are relatively natural and satisfactory. Our results are comparable to the results from EDISON System, and, in some cases, they are even more similar to human perception, although the proposed algorithm has low computational complexity.

VII. Conclusion

The paper presents simple, yet effective image segmentation algorithm. The algorithm is composed of preprocessing phase and segmentation phase. In the preprocessing phase, the image is filtered using bilateral filter in order to lower the influence of noise and texture details. In the segmentation phase L2 norm-based k-means algorithm is used due to its simplicity, fast execution time and good clustering properties. The experimental results show that its performance is close to the performance of much more complex algorithms.

There are few improvements that could be made in the algorithm. In order to make it faster, a less complex type of bilateral filter could be used, for example separable bilateral filter. For better segmentation it is possible to incorporate some spatial information about the pixels or add some spatial constraints, e.g. constrain the minimum or maximum segment size.

The proposed algorithm could be used as part of complex image processing techniques, such as content analysis and classification.

References

[1] Balafar, M. A., Ramli, A. R. and Mashohor, S., "Edge-preserving Clustering Algorithms and Their Application for MRI Image Segmentation." Hong Kong : s.n., 2010. Proceedings of the International MultiConference of Engineers and Computer Scientists.

[2] Mignotte, Max., "A de-texturing and spatially constrained K-means approach for image segmentation." Pattern Recognition Letters, 2010.

[3] Ilea, Dana Elena and Whelan, Paul F., "Adaptive Pre-Filtering Techniques for Colour Image Analysis." 2007. Machine Vision and Image Processing Conference.

[4] Garnica, Carsten., "A New Approach to Edge-Preserving Smoothing for Edge-Extraction and Image Segmentation." Amsterdam : International Archives of Photogrammetry and Remote Sensing, 2000.

[5] Tomasi, C. and Mancuchi, R., "Bilateral Filtering for Gray and Color Images." Bombay : s.n., 1998. IEEE International Conference on Computer Vision.

[6] Deelers, S and Auwatanamongkol, S., "Enhancing K-Means Algorithm with Initial Cluster Centers Derived from Data Partitioning along the Data Axis with the Highest Variance." 2007. Proceedings of world academy of science, engineering and technology volume 26.

[7] The Berkeley Segmentation Dataset and Benchmark. [Online] http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/.

[8] Berkeley Image Seg Online. [Online] http://internal.berkenviro.com/bis/.

[9] Georgescu, Bogdan and Christoudias, Chris., RIUL. [Online] Robust Image Understanding Laboratory. http://coewww.rutgers.edu/riul/research/code/EDISON/index.html.