EXPERIMENTAL COMBINATION OF INTENSITY AND STEREO EDGES FOR IMPROVED SNAKE SEGMENTATION

D. Markovic[1], M. Gelautz[2]

Institute for Software Technology and Interactive Systems, Vienna University of Technology
Favoritenstrasse 9-11/188/2, A-1040 Vienna, Austria
[1] markovic@ims.tuwien.ac.at, [2] gelautz@ims.tuwien.ac.at

In this paper, we present an algorithm to combine edge information from stereo-derived depth maps with edges from the original intensity/color image to improve the contour detection in images of natural scenes. After computing the disparity map, we generate a so-called "Edge Combination image", which relies on those edges of the original image that are also present in the stereo map. We describe an algorithm to identify corresponding intensity and depth edges, which are usually not perfectly aligned due to errors in the stereo reconstruction. Our experiments show that proposed Edge Combination approach can improve significantly the segmentation results of an Active Contour algorithm.

Introduction

For many image processing and computer vision tasks, object segmentation is an important basis. An often used approach for image segmentation is based on Active Contour Models, also known as snakes. The idea of Active Contours was first introduced in [5]. During recent years, a variety of snake models have been proposed, among them the Gradient Vector Flow (GVF) technique [9], which we employ in our study.

However, despite years of research, how to reliably extract contour information is still an open problem in many image and video processing tasks. Contour based approaches often have difficulties dealing with natural scenes, mostly because of highly textured regions or low contrast. Natural images usually contain textures, noise, or other effects such as shadowing that can prevent the Active Contour from converging to object boundaries. To overcome these problems, one of the most valuable information about the objects present in the scene is depth information ([2], [4], [8]). In our experiment, we utilize stereo-derived depth maps to improve the quality of the snake result. In most cases, however, the object contours can not be perfectly recovered from the disparity map alone, due to matching errors along depth discontinuities. We design and implement an algorithm which we refer to as "Edge Combination" in the following. The idea of the Edge Combination approach is to utilize the disparity edges in order to extract those edges of the original intensity/color image that are located along object boundaries.

The results of our experiments demonstrate that the proposed Edge Combination approach based on disparity maps and Active Contours can improve significantly the segmentation results, especially in textured regions, where snakes often fail to produce satisfactory results.

Algorithm

A summary of the involved processing steps can be seen in Fig. 1. A stereo image pair consisting of the left and right stereo image is processed by the module stereo matching, which delivers as output the stereo-derived depth map in the geometry of one of the two input images. The core of the processing chain is the Edge

Combination algorithm which we developed to determine those edges in the intensity image that are also present in the depth map. The algorithm comprises through following steps:

- **Edge Detection**: The first step is to detect edges in both the original image and its corresponding disparity/depth map. We use an implementation of Canny's edge detector provided by Matlab (version 6.5.1), which we modified to extract edges from color images. Also, for every edge pixel we gather information about the orientation of the corresponding edge at this location.

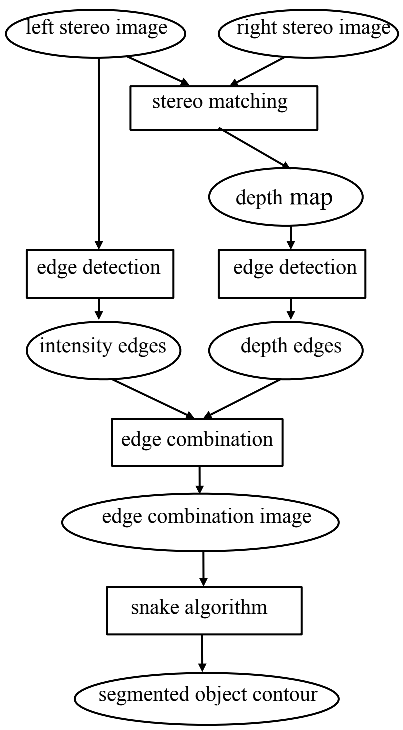


Fig. 1. Overview of processing chain.

- **Edge Search**: The original edge image (A) and the disparity edge image (B) are input to the Edge Combination procedure. For each edge pixel in A, we determine whether a corresponding edge pixel with a similar orientation can be found in B. We use a square search area with typical sizes between 4 and 12 pixels in one direction. To define similarity in edge orientation we usually employ a tolerance angle between 5° and 20°. We record every edge pixel in A that was found to have a corresponding edge pixel in B in order to include it in the edge combination image. In this way, we build up a "basic" edge combination image C.

- **Edge Linking**: Mostly because of imperfect disparity information, some pixels in the comparing process will not match, leaving a gap in the reconstructed contour line. In order to close minor gaps of this type, we implemented an edge linking procedure which repairs broken edges in C, if a continuous edge in A indicates that the edge segments should be connected.

First, we use a labeling algorithm to determine the connected edge components in A. For each end point of an edge in C, we search within a certain neighborhood - typically within a distance of 3 to 9 pixels - to find another end pixel in C. If both of them belong to the same edge in A, as determined by the previous labeling, we connect the two end points in C by inserting the corresponding edge segment from A. In practice, we copy an appropriate subwindow from A and insert it into C. Before insertion, we clean the subwindow by pruning superfluous parts of the copied edge pattern using the cleaning technique described in the following. The edge linking procedure terminates, if no more open end points that could be connected can be found in C.

- **Cleaning**: The effects of the edge linking and cleaning steps are illustrated in Fig. 2. The cleaning process, which relies on a maze solving strategy [6], is shown in more detail in Fig. 3. We use a subwindow of A that is bigger than the distance between the two end points that we want to connect. Fig. 3 (a) shows the subwindow from Fig. 2 (b) with the end points that should be connected marked red. Fig. 3 (e) gives the corresponding subwindow from A that we want to insert (compare Fig. 2 (c)). We remove the unneeded parts of the edge pattern in Fig. 3 (e) iteratively using the maze solving strategy. For every end pixel in (e), we check whether it coincides with one of the end pixels from (a) that we want to connect. If it does not coincide, we delete it. The end pixels of (e) are displayed in subfigure (b), and (f) gives the result after removing them from (e). Two more deletion steps are illustrated in (c), (d), (g), and (h). The procedure terminates if we only find end pixels that have the same position as those pixels in C that we want to connect. This condition is encountered in (h). Merging of (h) and (a) delivers the final result of the cleaning procedure presented in Fig. 2 (d).

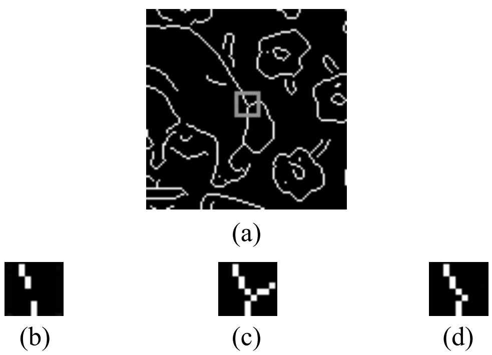


Fig. 2. (a) Part of the original edge image, (b) Subwindow before linking, (c) Subwindow after linking and before cleaning, (d) Subwindow after linking and cleaning.

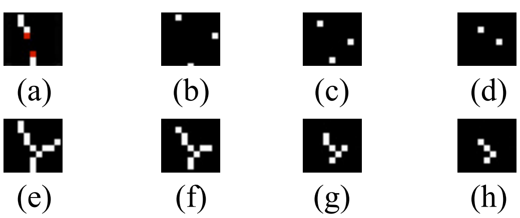


Fig. 3. Cleaning using the maze solving technique.

Test Data

In our experiments, we used a stereo configuration consisting of two Dragonfly IEEE-1394 color video cameras [7]. The camera set-up was calibrated using the calibration routines provided by Intel's Open Source Computer Vision (OpenCV) library [3]. For further processing, we transformed the stereo image pairs into epipolar geometry. An example of such a preprocessed stereo pair of video frames (size 400 x 400 pixels) is shown in Figs. 4 (a) and (b).

Experimental Results

In our Edge Combination tests, we utilized an implementation of the Pixel-to-Pixel stereo matching algorithm described in [1] to extract the depth information. The algorithm matches scan lines of stereo pairs in epipolar geometry individualy using dynamic programming. The resulting disparity map is given in Fig. 4 (c). Figs. 4 (d) and (e) show the edges derived from the disparity map (c) and the original intensity image (a), respectively. Subfigure (f) contains the contour edges computed by the Edge Combination approach. One can recognize the smoother appearance of the combined edges in (f) when compared to the stereo edges in (d).

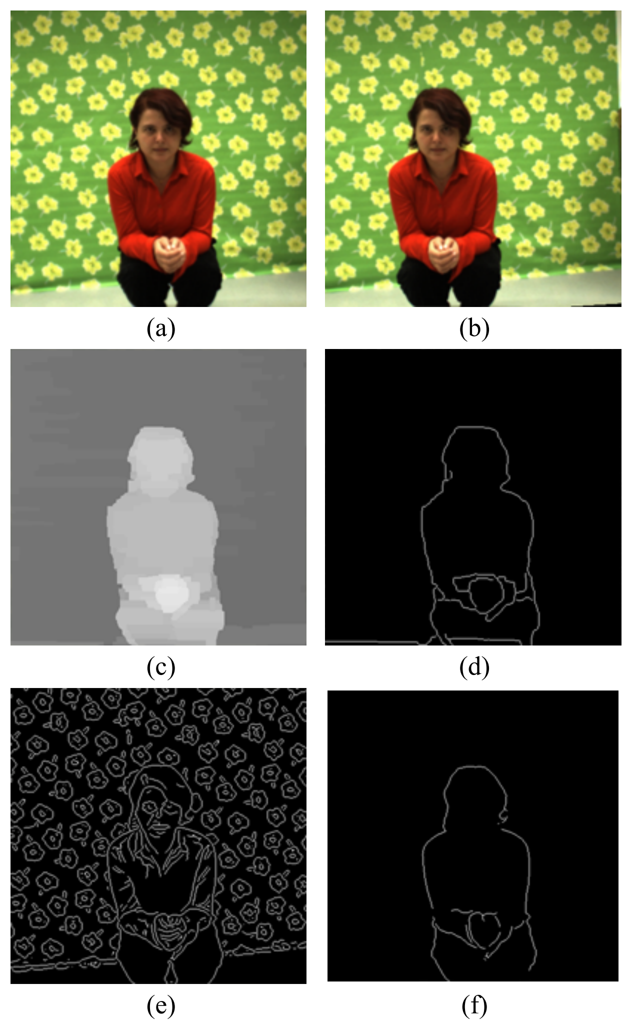


Fig. 4. (a) Left camera image, (b) right camera image, (c) depth image, (d) disparity edge image, (e) original edge image, (f) Edge Combination image.

Fig. 5 illustrates the results produced by applying the GVF snake to the original, disparity and Edge Combination image. The snake initialization, shown in Fig. 5 (a), and the parameters used for the snake computation were the same in all three cases. The results can be compared in Figs. 5 (b), (d), and (f). The Active Contour computed on the original image in (b) shows obvious errors caused by the background texture. Clearly, the background pattern pulls away the snake from the object of interest at several locations, which leads to poor segmentation results. More snake iterations resulted in even larger deviations between the computed and actual shape in (b). These errors are no longer present in the depth-derived snake result in (d). However, because of imperfectness of the stereo matching results, the final position of the GVF snake in (d) does not coincide exactly with the boundaries of the object. The errors in (b) and (d) are largely suppressed by the Edge Combination approach, as demonstrated by the almost perfect fit of the snake in Fig. 5 (f).

We carried out more experiments with other test data and obtained similar results. In all cases, the Edge Combination image produced a better snake result than the intensity image or depth map alone, which demonstrates the usefulness of the combined approach.

Summary and Outlook

In this paper we presented a method to combine intensity and stereo-derived edges for more reliable recognition of object contours. In experiments with stereo frames we demonstrated that the implemented Edge Combination algorithm can improve the performance of a GVF snake. In principle, the proposed approach can be applied as a postprocessing step to the output of any edge detection and stereo matching algorithm.

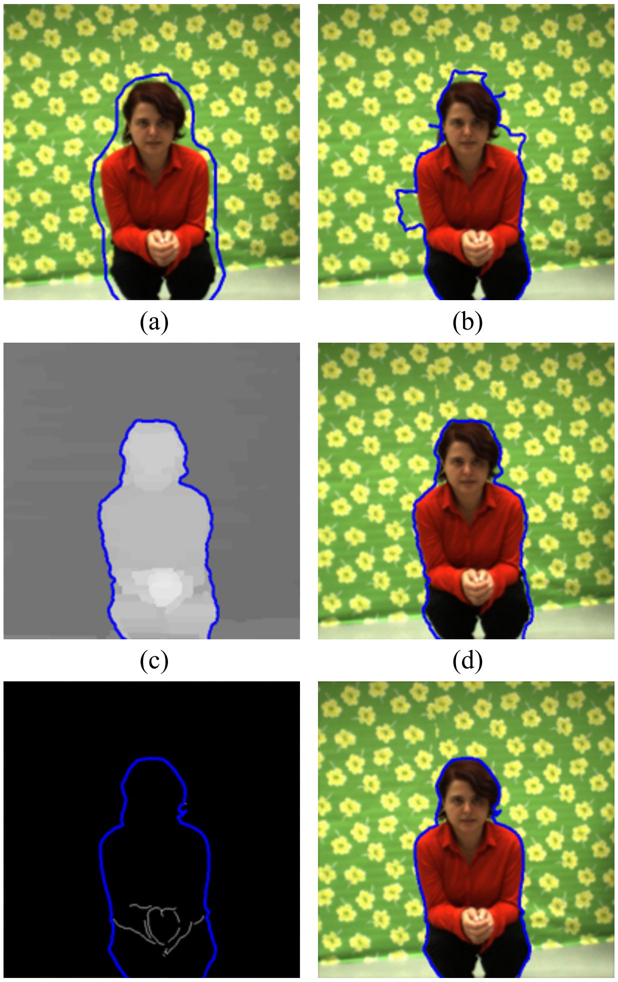
As part of an ongoing project, we are currently exploring possibilities to encode the extracted contour edges using efficient spline representation for subsequent image-based computer graphics rendering applications.

Acknowledgements

This work was supported by the Austrian Science Fund (FWF) under project P15663.

References

1. S. Birchfield, and C. Tomasi, Depth Discontinuities by Pixel-to-Pixel Stereo // International Journal of Computer Vision. - 1999. - Vol. 35, No. 3. - P. 269-293.
2. T. Darrell, G. Gordon, M. Harville, and J. Woodfill, Integrated Person Tracking Using Stereo, Color, and Pattern Detection // International Journal of Computer Vision. - 2000. - Vol. 37, No. 2. - P. 175-185.
3. Intel Open Source Computer Vision Library, version 3.1, 2003. http://www.intel.com/research/mrl/research/opencv/
4. E. Izquierdo, Disparity/Segmentation Analysis: Matching with an Adaptive Window and Depth-Driven Segmentation // IEEE Transactions on Circuits and Systems for Video Technology. - 1999. - Vol. 9, No. 4. - P. 589-607.
5. M. Kass, A. Witkin, and D. Terzopoulos, Snakes: Active Contour Models // International Journal of Computer Vision. - 1987. - Vol. 1, No. 4. - P. 321-331.
6. B. Nayfeh, Cellular Automata for Solving Mazes // Doctor Dobb's Journal. - 1993. - Vol. 18, No. 2. - P. 32-38.
7. Point Grey Research Inc., http://www.ptgrey.com, 2003.
8. W. Woo, N. Kim, and Y. Iwadate, Object Segmentation for Z-keying Using Stereo Techniques // Proceedings of ICSP. - 2000. - P. 1249-1254.
9. C. Xu and J.L. Prince, Gradient Vector Flow: A New External Force for Snakes // Proc. IEEE Conf. on Comp. Vis. Patt. Recog. (CVPR), Los Alamitos: Comp. Soc. Press. - 1997. - P. 66-71.



(a) (b)

(c) (d)

(e) (f)

Fig. 5. Experimental results with GVF snake: (a) original image with snake initialization, (b) final snake on original image, (c) final snake on depth image, (d) original image with snake from (c) overlaid, (e) final snake on edge combination image, (f) original image with snake from (e) overlaid.