Experimental Combination of Intensity and Stereo Edges for Improved Snake Segmentation¶

D. Markovic and M. Gelautz

Institute for Software Technology and Interactive Systems, Vienna University of Technology Favoritenstrasse 9–11/188/2, A–1040 Vienna, Austria

e-mail: markovic@ims.tuwien.ac.at; gelautz@ims.tuwien.ac.at

Abstract—In this paper, we present an algorithm to combine edge information from stereo-derived disparity 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 disparity edges, which are usually not perfectly aligned due to errors in the stereo reconstruction. Our experiments show that the proposed edge-combination approach can significantly improve the segmentation results of an active contour algorithm.

DOI: 10.1134/S1054661807010154

1. INTRODUCTION

For many image processing and computer vision tasks, object segmentation is an important basis. One frequently 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 when 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 sources of information about the objects present in the scene is the depth information ([2, 4, 8]). In our experiment, we utilize stereo-derived disparity maps to improve the quality of the snake result. In most cases, however, the object contours cannot be perfectly recovered from the disparity map alone, due to matching errors along depth discontinuities. To overcome these problems, we design and implement an algorithm which we refer to as "edge combination" in the following. The concept is illustrated in Fig. 1, where the left image of the stereo pair is shown in Fig. 1a, the corresponding disparity map in Fig. 1b, and their edge representations are given in Figs. 1c and 1d, respectively. The final result, the edge-combination image, can be seen in Fig. 1e.

The idea of the edge-combination approach is to utilize the disparity edges in order to localize those edges of the original intensity/color image that are located along the object boundaries. In other words, if we detect edges in the disparity image (see Fig. 1d), we will notice that this edge image contains the information about the object boundaries needed for object extraction, but due to the imperfectness of the depth discontinuities, the detected edges are not at exactly the same position as in the intensity edge image. On the other hand, the edges in intensity edge image (see Fig. 1c) are almost perfectly located, but the image also contains edges from texture and other features which can pull the snake away from the object boundaries and will therefore not give satisfactory results. The new image that we create, the edge-combination image, contains object contour edges in exact shape and position, while superfluous or erroneous edges caused by texture or matching errors are greatly suppressed.

The results of our experiments demonstrate that the proposed edge-combination approach based on disparity maps and active contours can significantly improve the segmentation results, especially in textured regions, where snakes often fail to produce satisfactory results.

2. OVERVIEW

A summary of the processing steps involved can be seen in Fig. 2. A stereo image pair consisting of a left and right stereo image is processed by the module stereo matching, which delivers as output the stereo-derived disparity map in the geometry of one of the two input images. The algorithm comprises the following steps. Firstly, an edge detector is applied to both the left stereo image and the computed disparity map. In most

¶The text was submitted by the authors in English.

Received April 5, 2005

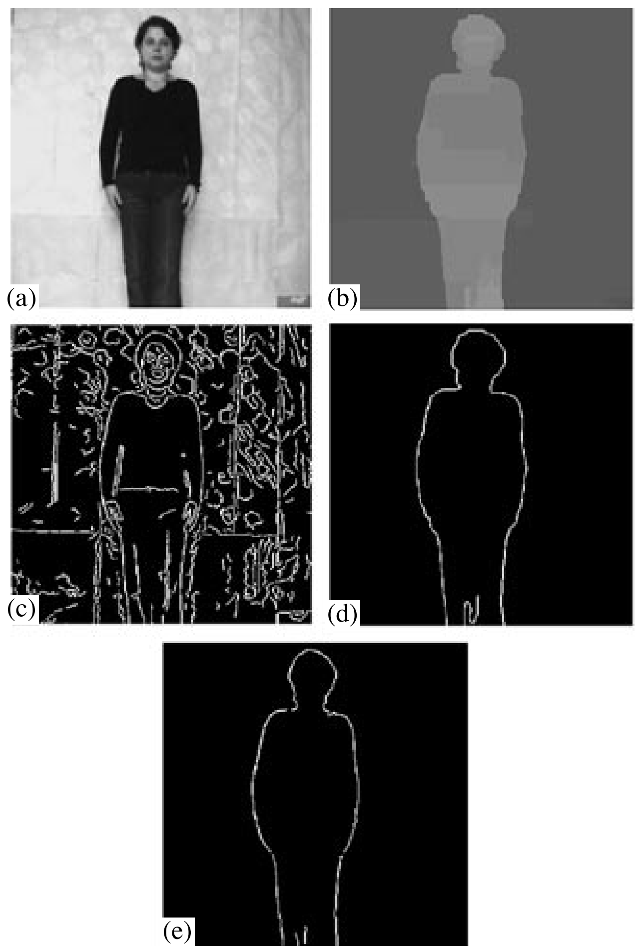

Fig. 1. (a) Left camera image, (b) disparity image, (c) original edge image, (d) disparity edge image, (e) edge-combination image.

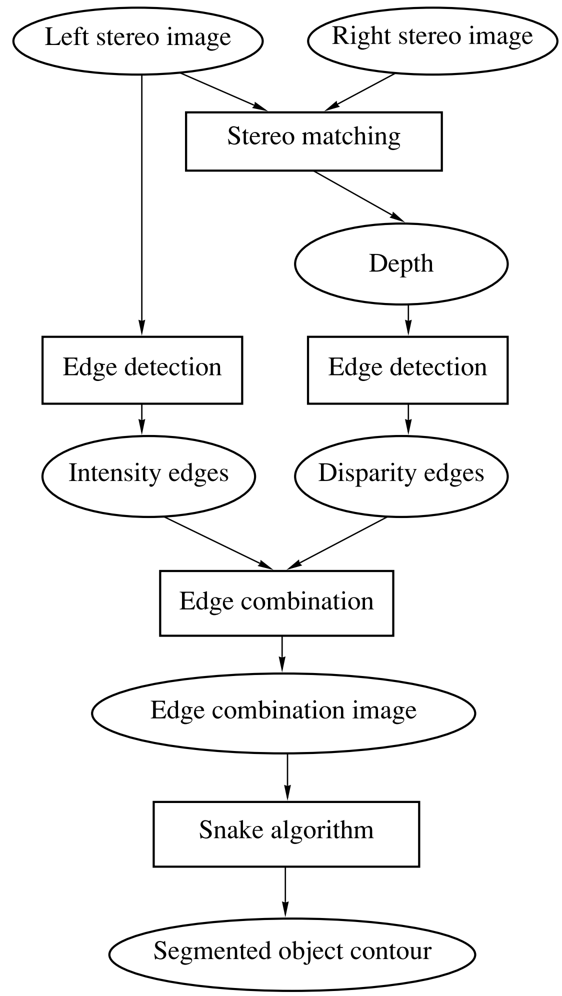

Fig. 2. Overview of processing chain.

cases, the object contours are not perfectly recovered in the disparity map, due to matching errors along depth discontinuities. In the next step, we determine those edges that are present in both the intensity edge image and the disparity edge image. The final result is an edge-combination image that contains the dominant contour edges of the objects in the scene. A more detailed explanation of all steps of the edge-combination procedure is given in the following sections.

As the final step of the processing chain from Fig. 2, we apply an active-contour (snake) algorithm to the edge-combination image in order to extract the object of interest from the background. For our experiments, we employed a snake implementation which follows the gradient vector flow (GVF) algorithm suggested by [9].

3. EDGE DETECTION

The first step is to detect edges in both the original image and its corresponding disparity 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. Moreover, for every edge pixel in both images, the intensity edge image and disparity edge image, we gather information about its orientation.

4. EDGE SEARCH

The intensity edge image (A) and the disparity edge image (B) are inputted to the edge-combination procedure, in which we determine the corresponding edge pixels based on edge orientation. For each edge pixel in image B, we search in image A for an edge pixel with similar orientation by using a search window whose size we increase in each step, until we find such a pixel or we reach a previously given maximum size of the window. The starting search position in image A is the initial position of the pixel in image B. To define the similarity in edge orientation, we employ a tolerance angle that is usually between 5° and 20°. If we find a corresponding pixel, we include it in the edge-combination image at the position of the pixel found in

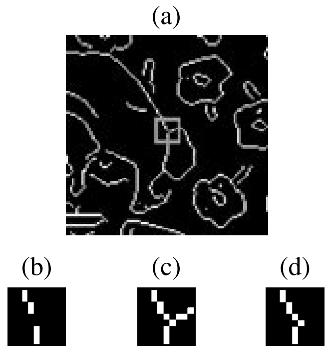

Fig. 3. (a) Part of the original edge image, (b) Subwindow before linking, (c) Subwindow after linking and before cleaning, (d) Subwindow after linking and cleaning.

image A. In this way, we build up a "basic" edge-combination image C.

5. EDGE LINKING

The previously created image C suffers from two distinct problems that arise due to imperfect disparity information:

—Incorrect pixel matches that are not part of the desired contour are considered noise pixels and should be removed. For this we determine for each labeled edge in A the number of corresponding pixels in C. If the obtained value is smaller than a previously specified threshold, we declare those pixels in C as noise and remove them.

—The second problem occurs when some pixels in the comparing process do not match, leaving a gap in the reconstructed contour line. In order to close these minor gaps, 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.

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 can be connected are found in C.

6. CLEANING

The effects of the edge linking and cleaning steps are illustrated in Fig. 3. The cleaning process, which relies on a maze-solving strategy [6], is shown in more detail in Fig. 4. We use a subwindow of A that is bigger than the distance between the two end points that we want to connect. Figure 4a shows the subwindow from Fig. 3b with the end points that should be connected marked red. Figure 4e gives the corresponding subwindow from A that we want to insert (compare Fig. 3c). We remove the unneeded parts of the edge pattern in Fig. 4e iteratively using the maze-solving strategy. For every end pixel in Fig. 4e, we check whether it coincides with one of the end pixels from Fig. 4a that we want to connect. If it does not coincide, we delete it. The end pixels of Fig. 4e are displayed in Fig. 4b, and Fig. 4f gives the result after removing them from Fig. 4e. Two more deletion steps are illustrated in Figs. 4c, 4d, 4g, and 4h. The procedure terminates if we find only end pixels that have the same position as those pixels in C that we want to connect. This condition is encountered in Fig. 4h. Merging of Figs. 4h and 4a delivers the final result of the cleaning procedure presented in Fig. 3d.

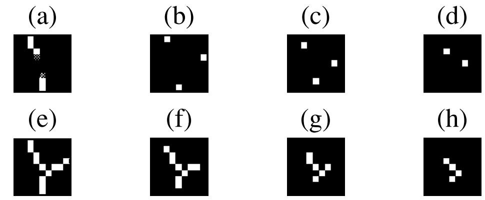

Fig. 4. Cleaning using the maze-solving technique.

7. 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 a thus-preprocessed stereo pair of video frames (size 400×400 pixels) is shown in Figs. 5a and 5b.

8. 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 individually using dynamic programming. The resulting disparity map is given in Fig. 5c. Figures 5d and 5e show the edges derived from the disparity map in Fig. 5c and the original intensity image in Fig. 5a, respectively. Figure 5f contains the contour edges computed by the edge-combination approach. One can recognize the smoother appearance of the combined edges in Fig. 5f when compared to the stereo edges in Fig. 5d.

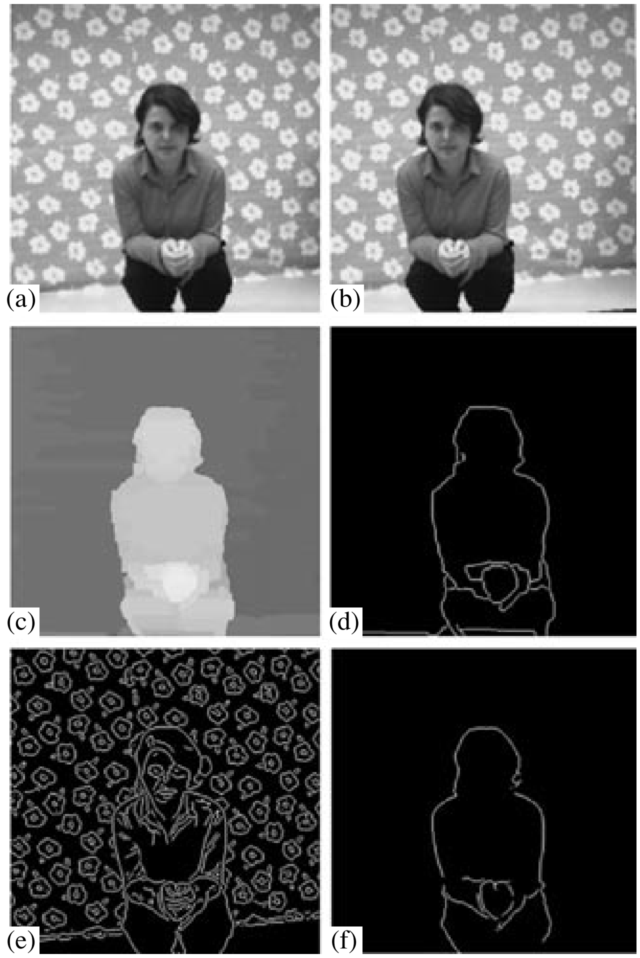

Fig. 5. (a) Left camera image, (b) right camera image, (c) disparity image, (d) disparity edge image, (e) original edge image, (f) Edge-combination image.

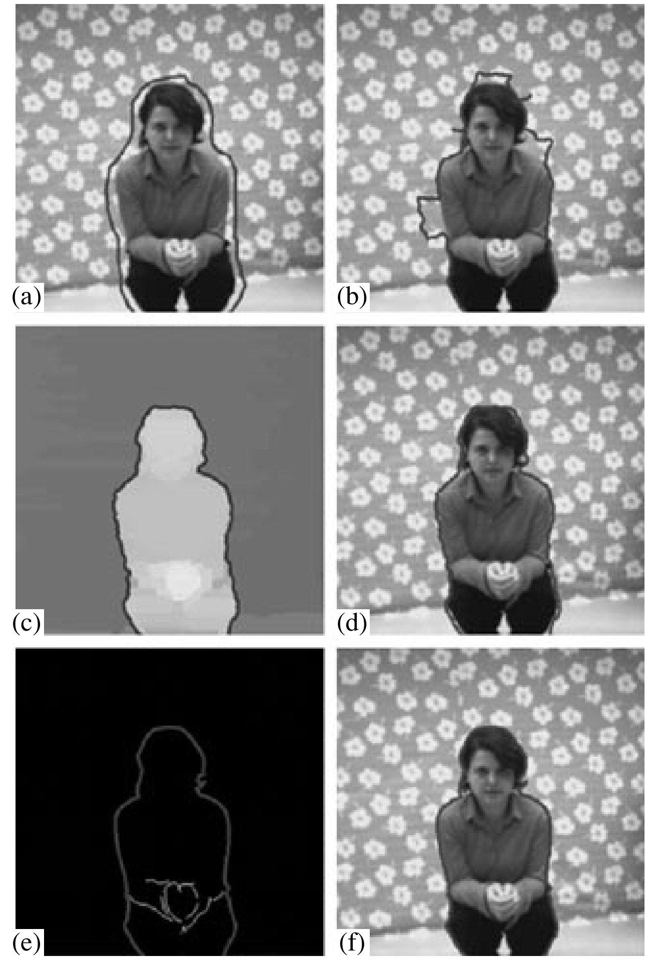

Fig. 6. Experimental results with GVF snake: (a) original image with snake initialization, (b) final snake on original image, (c) final snake on disparity image, (d) original image with snake from (c) overlaid, (e) final snake on edge-combination image, (f) original image with snake from (e) overlaid.

Figure 6 illustrates the results produced by applying the GVF snake to the original, disparity, and edge-combination images. The snake initialization, shown in Fig. 6a, and the parameters used for the snake computation were the same in all three cases. The results can be compared in Figs. 6b, 6d, and 6f. The active contour computed on the original image in Fig. 6b shows obvious errors caused by the background texture. Clearly, the background pattern pulls the snake away 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 Fig. 6b. These errors are no longer present in the disparity-derived snake result in Fig. 6d. However, because of the imperfection of the stereo-matching results, the final position of the GVF snake in Fig. 6d does not coincide exactly with the boundaries of the object. The errors in Figs. 6b and 6d are largely suppressed by the edge-combination approach, as demonstrated by the almost perfect fit of the snake in Fig. 6f.

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 disparity map alone, which demonstrates the usefulness of the combined approach.

9. 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 an efficient spline representation for subsequent image-based computer graphics rendering applications.

ACKNOWLEDGMENTS

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 **35** (3), 269–293 (1999).
2. T. Darrell, G. Gordon, M. Harville, and J. Woodfill, "Integrated Person Tracking Using Stereo, Color, and Pattern Detection," International Journal of Computer Vision **37** (2), 175–185 (2000).
3. http://www.intel.com/research/mrl/research/opencv/ (Intel Open Source Computer Vision Library, version 3.1, 2003).
4. E. Izquierdo, "Disparity/Segmentation Analysis: Matching with an Adaptive Window and Depth-Driven Segmentation," IEEE Transactions on Circuits and Systems for Video Technology **9** (4), 589–607 (1999).
5. M. Kass, A. Witkin, and D. Terzopoulos, "Snakes: Active Contour Models," International Journal of Computer Vision **1** (4), 321–331 (1987).
6. B. Nayfeh, "Cellular Automata for Solving Mazes," Doctor Dobb's Journal **18** (2), 32–38 (1993).
7. http://www.ptgrey.com (Point Grey Research Inc., 2003).
8. W. Woo, N. Kim, and Y. Iwadate, "Object Segmentation for Z-keying Using Stereo Techniques," in *Proceedings of ICSP, 2000*, pp. 1249–1254.
9. C. Xu and J.L. Prince, "Gradient Vector Flow: A New External Force for Snakes," in *Proceedings of IEEE Conf. on Comp. Vis. Patt. Recog. (CVPR), Los Alamitos, 1997* (Comp. Soc. Press., 1997), pp. 66–71.

Danijela Markovic graduated from the Faculty of Electronic Engineering, University of Nis, Serbia in 1997. She is currently a PhD student at the Institute for Software Technology and Interactive Systems, Vienna University of Technology. Her research interests are in computer vision and computer graphics, including stereo vision and curve/surface modeling. Particularly, she is interested in object segmentation, feature extraction, and tracking.

Margrit Gelautz received her PhD degree in computer science from Graz University of Technology, Austria. She worked on stereo and interferometric image processing for radar remote sensing applications during a postdoctoral stay at Stanford University. Her current research interests include image and video processing for multimedia applications, with a focus on 3D vision and rendering techniques.