Recognition of Object Contours from Stereo Images: an Edge Combination Approach

Margrit Gelautz and Danijela Markovic
Institute for Software Technology and Interactive Systems, Vienna University of Technology
Favoritenstrasse 9-11/188/2, A-1040 Vienna, Austria
phone: +43 1 58801 18849, fax: +43 1 58801 18898, email: gelautz@ims.tuwien.ac.at

Abstract

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 slightly displaced due to non-perfect stereo reconstruction. Our experiments demonstrate how the proposed edge combination approach can be used in conjunction with an active contours algorithm to achieve better segmentation results.

1. Introduction and Motivation

The recognition of contour edges for object segmentation constitutes a crucial step in many image and video processing tasks. Whereas in many industrial computer vision applications an approximate model of the object to be extracted is available, multimedia applications such as video editing and compositing typically deal with complex objects (e.g., moving humans) in varying environments and illumination conditions (e.g., outdoors), which makes the recognition task particularly challenging.

In the context of image and video object segmentation, stereo-based techniques are currently gaining importance due to the growing market of inexpensive digital cameras that are available for use in multi-camera set-ups or as in-built stereo systems (such as, e.g., Point Grey's Bumblebee camera [5].) The use of stereo analysis for object segmentation is, for example, proposed by [9]. The approach is motivated by the observation that discontinuities in the depth map are often a better indicator of object boundaries than discontinuities in the original intensity/color image, which may also be caused by changing surface reflectance properties inside the object.

The key step in stereo processing is stereo matching, that is the identification of corresponding points between the two stereo partners. The authors of [7] discuss several state-of-the-art matching algorithms and analyze their performance in application to a common set of test data. The output of the stereo matching is a disparity map which can be converted into a depth map by using the known geometric relationships of the camera set-up.

Whereas a perfect depth map might be used to separate the object of interest from the background in a relatively straightforward way, in practical stereo analysis we have to cope with stereo matching errors that lead to erroneous depth values which are particularly present in the absence of texture and along object discontinuities. A thorough discussion of the errors produced by correlation-based matching techniques in the vicinity of object boundaries is given by [2]. The authors explain the so-called "fattening" effect, which makes objects that are closer to the camera appear larger in the stereo map than in the intensity image. The non-perfect edge localization in stereo maps will also be visible in the examples we show in section 3.

In order to suppress the matching-induced errors along object boundaries, we suggest combining the edges derived from the depth map with the original intensity edges for more accurate localization of contour edges. The basic idea of the edge combination approach is sketched in Fig. 1. Natural images usually contain regions with texture, noise, or other effects such as shadowing that can prevent the active contour from converging to object boundaries. An example of an image with background texture is shown in Fig. 1

(a). The corresponding disparity (or depth) map generated by stereo matching is displayed in Fig. 1 (b). It should be noted that in most cases the object contours are not perfectly recovered in the disparity map, due to matching errors along depth discontinuities. Fig. 1 (c) shows the edges derived from the original image overlaid onto the edges from the disparity map. The idea of the edge combination approach is to utilize the disparity edges to filter out those edges of the original intensity/color image that are located along object boundaries. The ideal result of such a combination is sketched in Fig. 1 (d).

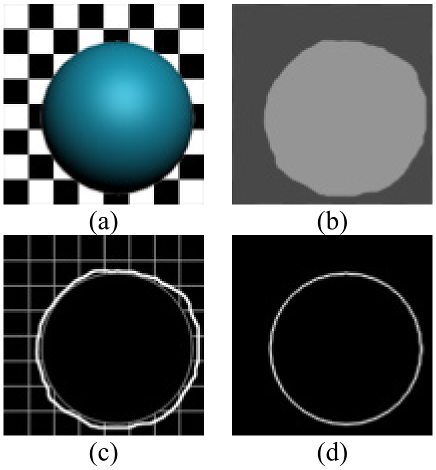


Figure 1: (a) Original image, (b) disparity image, (c) original and depth edges overlaid, (d) edge combination image.

As opposed to the development of specific matching techniques that incorporate edges (for example, [1] or [3]), the proposed edge combination approach constitutes a post-processing step that provides high flexibility; in principle, the procedure can be applied to the output of any edge detection and stereo matching algorithm. The free choice of the matching algorithm constitutes a major difference between our study and previous work on the combined exploitation of 2D and 3D edge information published by [6]. The authors of [6] perform stereo matching on the extracted intensity edges, which results in a sparse disparity map with disparity values computed only for the edge pixels. In our work, the identification of corresponding intensity and disparity edges requires a major effort, because the two types of edges are usually slightly displaced. Since our approach is not bound to a specific matching technique, we can incorporate recent developments in dense stereo matching algorithms as provided by [7].

A major motivation for our study on edge extraction is the development of image-based non-photorealistic rendering (NPR) techniques that require the precise localization of contour edges in order to generate high-quality rendering results [8]. We describe the edge combination algorithm in section 2. In section 3, we apply the technique to stereo video frames delivered by two Dragonfly color cameras [5] and show that the edge combination image delivers more precise contour information and segmentation results than the intensity image or depth map alone.

2. Algorithm

2.1. Algorithmic Outline

A summary of the involved processing steps can be seen in Fig. 2. 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. For stereo matching, we use the technique developed by [1]. The algorithm matches individual scan lines of stereo pairs in epipolar geometry using dynamic programming.

As the next step, an edge detector (e.g., Canny) is applied to the intensity image and depth map, resulting in a set of intensity edges and depth edges. 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 idea is to combine the edges from the stereo map, which suggest the location of scene discontinuities, with the higher positional accuracy of the intensity edges for more accurate recognition of contour edges.

The edge combination algorithm can be divided into three major steps: (a) edge search, (b) edge linking, and (c) cleaning. During step (a), a pixel-by-pixel search is performed in the surrounding of each intensity edge to find a corresponding depth edge. The search takes into account the orientation of the edge to reduce the possibility of mismatches. The result of step (a) is further refined during step (b). The edge linking process bridges minor gaps in the computed combination edges based on a labeling of the connected edge components in the original intensity image. The linking procedure helps to suppress the effects of broken depth edges caused by matching errors along object boundaries. Additional cleaning functions are applied afterwards to eliminate superfluous pixels. Steps (a), (b), and (c) are described in more detail in subsections 2.3, 2.4, and 2.5, respectively. The result of the edge combination step is a set of edges that are located for the most part along object contours.

The final step of the processing chain from Fig. 2 is the application of an active contours (snake) algorithm to the edge combination image. Generally, snakes serve to extract an object of interest from the background using the gradient information inherent in the image. For our experiments in section 3, we employ a snake implementation which follows the Gradient Vector Flow (GVF) algorithm suggested by [10].

2.2. 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 slightly to extract edges from color images. Also, for every edge pixel we gather information about the orientation of the corresponding edge at this location.

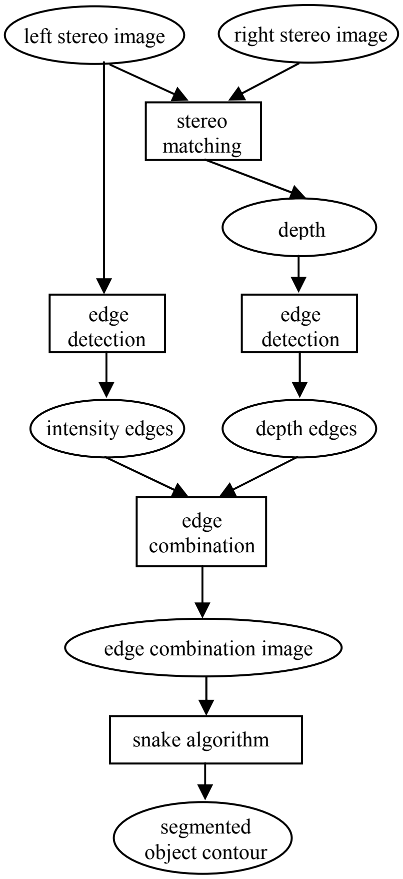


Figure 2: Overview of processing chain.

2.3. 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 window 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.

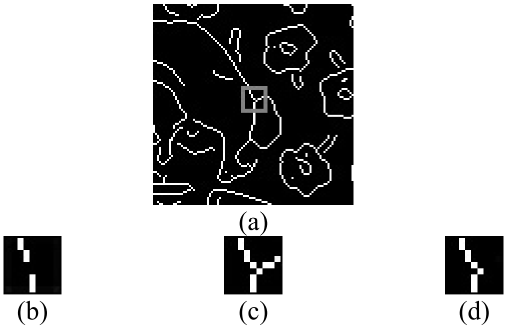
2.4. Edge Linking

In many cases, we encounter missing pixels in the reconstructed contour lines from C that are mostly due to imperfect disparity information. 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 suggests 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.

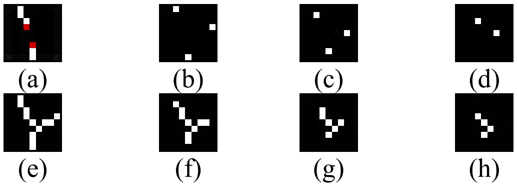
2.5. Cleaning

The effects of the edge linking and cleaning steps are illustrated in Fig. 3. The cleaning process, which relies on a maze solving strategy [4], 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. Fig. 4 (a) shows the subwindow from Fig. 3 (b) with the end points that should be connected marked red. Fig. 4 (e) gives the corresponding subwindow from A that we want to insert (compare Fig. 3 (c)). We remove the unneeded parts of the edge pattern in (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. 3 (d). The user interface of the software package we developed for edge combination is shown in Fig. 5.


(a)

(b) (c) (d)

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


(a) (b) (c) (d)

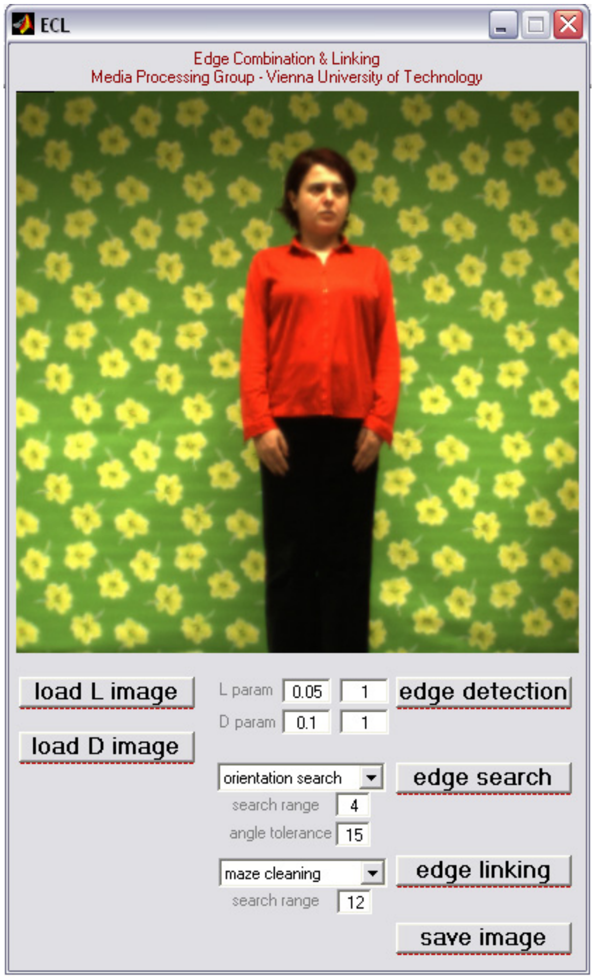
(e) (f) (g) (h)

Figure 4: Subwindow cleaning using the maze solving technique as explained in subsection 2.5.

3. Tests and Results

3.1. Test Data

For our tests, we employed a stereo configuration of two Dragonfly IEEE-1394 video cameras as delivered by Point Grey [5]. The frames are provided to the user in 24 bit RGB format. The camera set-up was calibrated using the calibration routines provided by Intel's Open Source Computer Vision Library. We then transformed the image pairs into epipolar geometry to facilitate the subsequent stereo analysis.


ECL

Edge Combination & Linking
Media Processing Group - Vienna University of Technology

L param 0.05 1

D param 0.1 1

orientation search

search range 4

angle tolerance 15

maze cleaning

search range 12

Figure 5: Graphical User Interface of the edge combination package implemented in Matlab.

3.2. Experimental Results

The application of the method to our test data and the results obtained are illustrated in Figs. 6 through 9. Figs. 6 (a) and (b) show a pair of stereo video frames in epipolar geometry. The size of the images is 400 x 400 pixels. The disparity map derived from stereo matching is displayed in Fig. 6 (c). Figs. 6 (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).

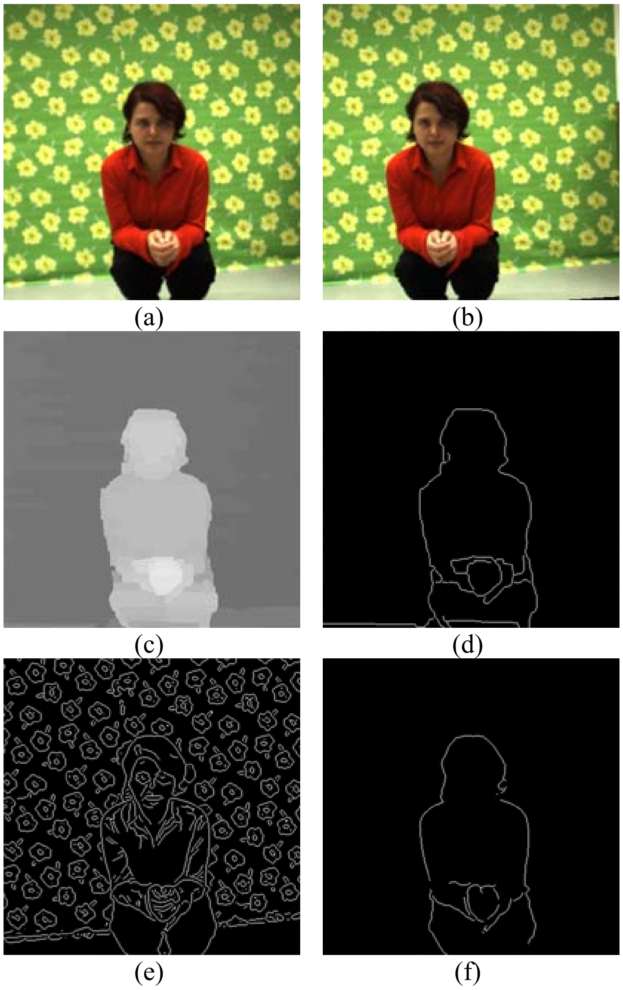


Figure 6: (a) Left camera image, (b) right camera image, (c) depth image, (d) disparity edge image, (e) original edge image, (f) edge combination image.

The application of the snake algorithm is illustrated in Fig. 7. Subfigure (a) shows the manual initialization of the snake. Identical initialization points and default parameters were used for all tests with the same data set. The snake result obtained from the original intensity image from Fig. 6 (a) can be seen in Fig. 7 (b). 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).

The snake computed for the depth map from Fig. 6 (c) is displayed in Fig. 7 (c). For better comparison, the same snake is overlaid onto the original image in Fig. 7 (d). One can recognize that the depth-derived snake in (d) follows the object contours more closely than the intensity-derived snake in (b). However, minor deviations between the computed snake and the actual object contours are still visible in (d). Finally, Figs. 7 (e) and (f) show the snake computed for the edge combination image. The almost perfect fit of the snake in (f), when compared to the results in (b) and (d), confirms the improvement achieved by the edge combination approach.

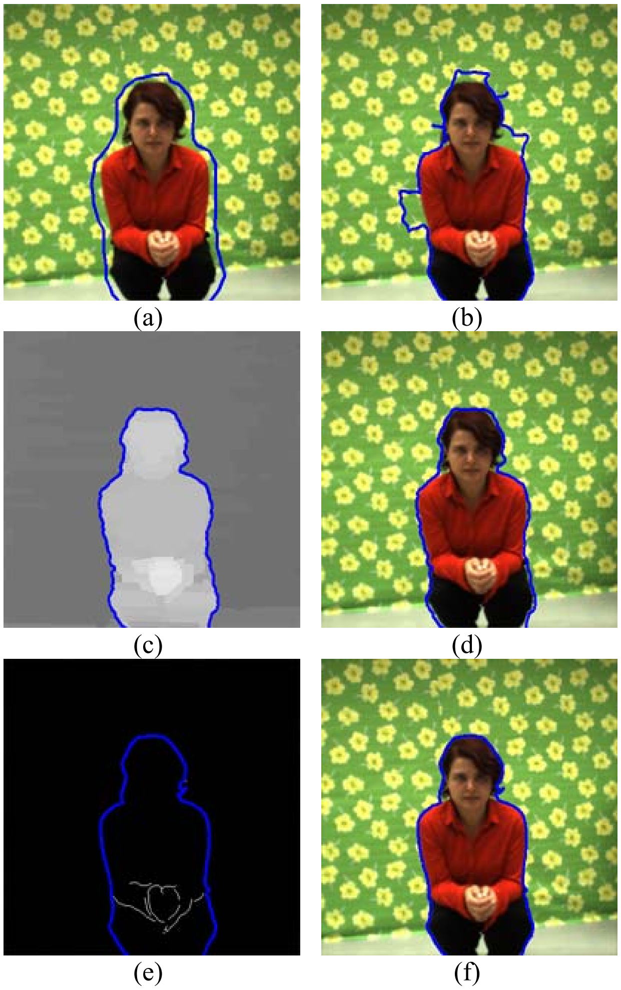


Figure 7: Experimental results with GVF snake: (a) original image with snake initialization, (b) final snake on original image, (c) final snake on depth image, (d) original image with snake from (c) overlaid, (e) final snake on edge combination image, (f) original image with snake from (e) overlaid.

Similar results were obtained from the data set shown in Figs. 8 and 9. The unsatisfactory snake result derived from the original intensity image can be inspected in Fig. 9 (b). The snake computed for the depth image delivered already much better results. However, a remaining misfit between the stereo-derived snake and the real object contour is apparent from the crinkly contour line in Fig. 9 (d). Again, the best snake result was achieved using the edge combination technique, as demonstrated in Fig. 9 (f). Besides the better adjustment of the final snake, we also observed that the snake results obtained from the edge combination image were more robust to changes in the snake parameters than the snakes computed from the intensity image.

Fig. 10 gives another view of the extracted edge information. With an eye to future applications in non-photorealistic rendering (NPR), we display the intensity edges from Fig. 6 (e) with varying gray values according to their significance and distance to the viewer. The principal shape of the object is captured by the contour edges from the edge combination approach, which are drawn in a dark gray tone. We display the other edges using lighter gray tones according to their distance to the viewer so that the viewer's attention focuses on foreground elements.

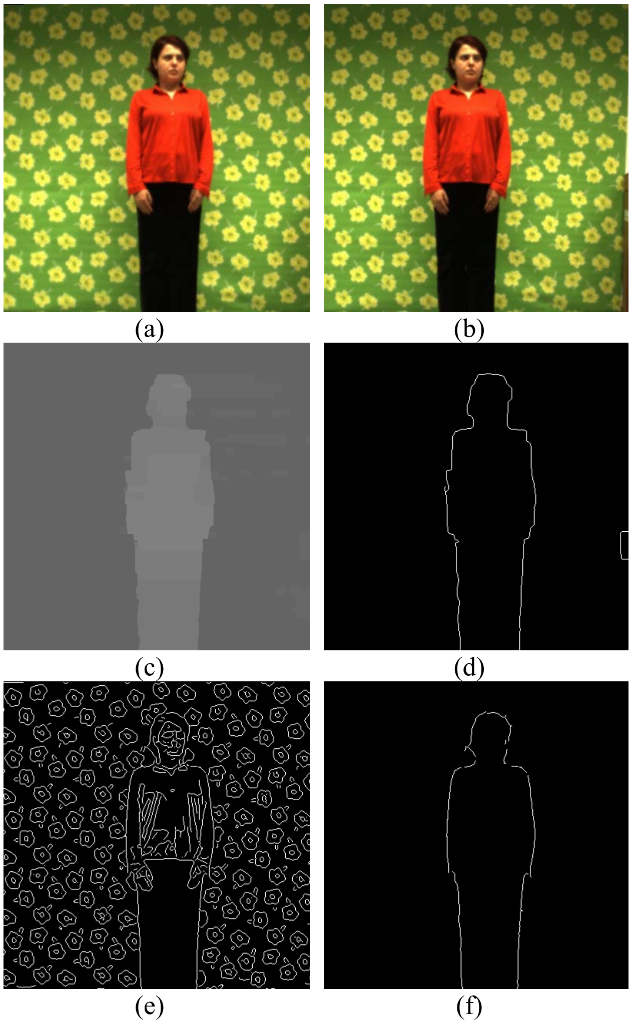


Figure 8: Example 2: (a) left stereo image, (b) right stereo image, (c) stereo-derived depth map, (d) depth edges, (e) intensity edges, and (f) edge combination image.

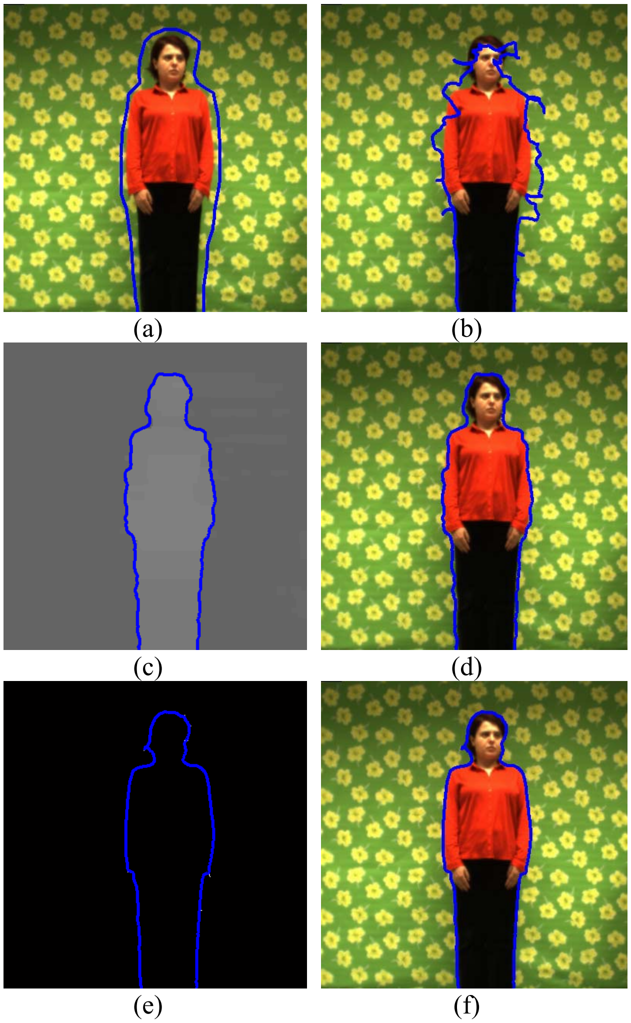


Figure 9: Snake results from example 2: (a) snake initialization, (b) intensity-derived snake, (c) depth-derived snake, (d) depth-derived snake overlaid onto original image, (e) edge combination snake, and (f) edge combination snake overlaid onto original image.

4. Conclusions

We have presented a method to improve the recognition of object contours from stereo images by combining the robustness of stereo-derived edges with the higher precision of the original intensity edges. We implemented an edge combination algorithm that can suppress the effects of stereo matching errors, especially along object boundaries, provided that the erroneous depth edges do not coincide with edges in the intensity image. Our tests with stereo video frames have confirmed that the developed technique can improve the performance of an active contours algorithm.

As part of an ongoing project, we are currently exploring possibilities to utilize the extracted contour edges along with the stereo-derived depth information for computer-generated drawing and the creation of artistic work from real images and videos scenes.

Acknowledgement

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

5. References

[1] S. Birchfield and C. Tomasi, “Depth Discontinuities by Pixel-to-Pixel Stereo”, *International Journal of Computer Vision*, vol. 35 (3), pp. 269-293, 1999.

[2] H. Hirschmueller, P. Innocent, and J. Garibaldi, “Real-time Correlation-based Stereo Vision with Reduced Border Errors”, *International Journal of Computer Vision*, vol. 47 (1/2/3), pp. 229-246, 2002.

[3] E. Izquierdo, “Disparity/Segmentation Analysis: Matching with an Adaptive Window and Depth-driven Segmentation”, *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 9 (4), pp. 589-607, 1999.

[4] B. Nayfeh, “Cellular Automata for Solving Mazes”, *Doctor Dobb’s Journal*, pp. 32-35, February 1993.

[5] Point Grey Research Inc., http://www.ptgrey.com, 2003.

[6] T.P. Pridmore, J.E.W. Mayhew, and J.P. Frisby, “Exploiting Image-Plane Data in the Interpretation of Edge-Based Binocular Disparity”, *CVGIP*, vol. 52 (1), pp. 1-25, October 1990.

[7] D. Scharstein and R. Szeliski, “A Taxonomy and Evaluation of Dense Two-frame Stereo Correspondence Algorithms”, *International Journal of Computer Vision*, vol. 47 (1/2/3), pp. 7-42, 2002.
http://www.middlebury.edu/stereo/

[8] E. Stavrakis and M. Gelautz, “Image-Based Stereoscopic Painterly Rendering”, to appear in *Proceedings of the Eurographics Symposium on Rendering*, Norrkoeping, June 2004.

[9] W. Woo, N. Kim, and Y. Iwadate, “Object Segmentation for Z-keying Using Stereo Images”, in *Proceedings of ICSP 2000*, pp. 1249-1254, August 2000.

[10] C. Xu and J.L. Prince, “Gradient Vector Flow: A New External Force for Snakes”, in *Proc. IEEE Conf. on Comp. Vis. Patt. Recog. (CVPR)*, Los Alamitos: Comp. Soc. Press, pp. 66-71, June 1997.

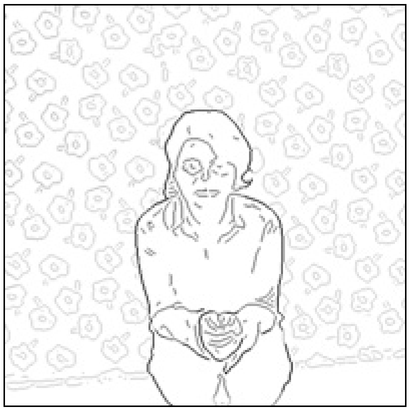


Figure 10: Edges extracted from the original image from Fig. 6 (a) with contour edges and foreground elements emphasized by darker gray values.