SEGMENTATION OF LINES AND ARCS AND ITS APPLICATION FOR DEPTH RECOVERY

Rüdiger Beß[†] & Dietrich Paulus & Michael Harbeck
Lehrstuhl für Mustererkennung (Informatik 5)
Friedrich-Alexander-Universität Erlangen-Nürnberg
Martensstr. 3, D–91058 Erlangen, Germany
{bess,paulus}@informatik.uni-erlangen.de

Accepted for
ICASSP 1997

Rev.: 2.35

CONTENTS

LIST OF FIGURES

SEGMENTATION OF LINES AND ARCS AND ITS APPLICATION FOR DEPTH RECOVERY

Rüdiger Beß[†] & Dietrich Paulus & Michael Harbeck
Lehrstuhl für Mustererkennung (Informatik 5)
Friedrich-Alexander-Universität Erlangen-Nürnberg
Martensstr. 3, D–91058 Erlangen, Germany
{bess,paulus}@informatik.uni-erlangen.de

ABSTRACT

In this paper we describe an advaced segmentation approach for stereo images improving the computation of depth compared to the commonly used straight line segmentation. Using a straight line – circular arc approximation of chain coded lines, the number of primitives is reduced significantly. This approximation lowers the computational effort as well as the frequency of erroneous matches. Starting with matched pairs of primitives, a disparity image is computed containing the initial disparity values for a subsequent block matching algorithm. The output of this algorithm is the partially dense depth image of one aspect of the object. We describe the result of a parallel implementation using object–oriented programming techniques. In segmentation as well as in matching we evaluate color information to improve accuracy and reliability of the depth values. The algorithms are part of a system computing depth from monocular image sequences. Taking a sequence of different views by a camera mounted to a robot hand, each two consecutive images are considered as a stereo image. The depth images computed from these stereo images are fused to one complete depth map of the object surface. The results show substantial improvements in comparison to a monochrome system with respect to speed, accuracy, and completeness.

1 INTRODUCTION

The system introduced in this paper is designed to compute a dense depth map of an object seen from different viewing angles. For this purpose, a monocular color camera is mounted to the hand of a robot allowing arbitrary positioning within the robot's working space. The relation between camera and robot hand is fixed and calibrated once in the startup phase of the system. This setup enables us to record object views from different angles with known camera positions. The accuracy of the camera position is limited by measurement errors in the robot position which can be up to 4mm. To overcome the calibration error and illumination changes due to camera rotation, a combined feature and correlation based approach is used. The feature based step allows to compute depth at the edges of an object accurately and reliably. The number of matching errors is reduced to a minimum by using a color edge filter, by evaluating color features at the edges, and by approximating detected lines by straight line – circular arc sequences (Sect. 3). While the matching step is done on the rather coarse approximation, the disparity is computed on the chain codes of the lines directly. Thus the number of primitives can be reduced without loss of accuracy. (Sect. 4).

The depth values at the edges are considered as points of support for a subsequent block matching step. Here again the color information is used to improve the accuracy. In this paper we describe the segmentation and its relation to stereo matching techniques. The other modules are explained in [1, 2]. The base of the segmentation and its implementation in an object oriented programming style is the work represented in [7, 14]. We introduce a hierarchical order for image processing operators implemented as classes and discuss the advantages of such a system (Sect. 5). We present results in Sect. 6 and conclude our contribution in Sect. 7 giving an outlook on further work.

[†]This work was funded partially by the German Research Foundation (DFG) under grant number SFB 182. Only the authors are responsible for the contents.

2 RELATED WORK

Several papers have been published about segmentation of images into lines, circular arcs, and vertices e.g. [4, 9]. Rather than only adding a new reference to this list, we *apply* such techniques to feature based stereo and introduce a new segmentation technique.

The evaluation of different color spaces for area based stereo matching showed better results compared to gray–level images [10]; the best result was obtained in RGB color space [8].

While feature based stereo algorithms result in sparse depth maps with high reliability and accuracy [12] block matching algorithms generate dense depth maps but are sensitive to illumination changes.

Several types of algorithms to compute dense depth of the complete surface of an object can be found in the literature, some of them with impressive results, but they necessitate to influence the scene; thus they are not suitable for unknown environments. Either a calibration rig has to be placed in the scene to determine the camera position (compare [11] for an overview of photogrammetric approaches) or the object has to be placed on a turn table as in [18].

Object–oriented programming has become common for image analysis by the dissemination of the IUE ideas [5]. Data representation is widely implemented in classes. However, little has been published yet on hierarchies of operators for image processing.

3 SEGMENTATION INTO SEQUENCES OF LINES AND ARCS

Using the robot, a sequence of color images is recorded showing the object from different viewing angles. Two adjacent images are treated as a stereo pair which is normalized so that epipolar lines correspond with the scan lines. This step is used additionally to change the image size to a power of two. This transformation from PAL format to 512×512 pixel simplifies the use of a resolution hierarchy.

After these preprocessing steps, color edges are detected by a Nevatia–Babu edge filter. Each of the three channels of the RGB color space is filtered independently resulting in three different values of edge strength and edge orientation of each pixel. These values are transformed to one by choosing the maximal edge strength and the corresponding edge orientation. Combining the edge information previous to line detection saves the fusing step which would be necessary to combine lines detected at slightly different positions in the three color channels.

The single edge–image is subject to line detection by a hysteresis algorithm which produces a set of continuous digital lines represented as chain codes; this set is part of a so called "segmentation object" which is the common data structure for the representation of complex segmentation results (Sect. 5).

The set of chain codes computed by the line following algorithm are now passed to a split–and–merge procedure to approximate the chains as sequences of straight line segments and circular arcs. The result is again stored in a segmentation object. An example illustrating the power of the approach is shown in Fig. 1. Some implementation details will be given in Sect. 5

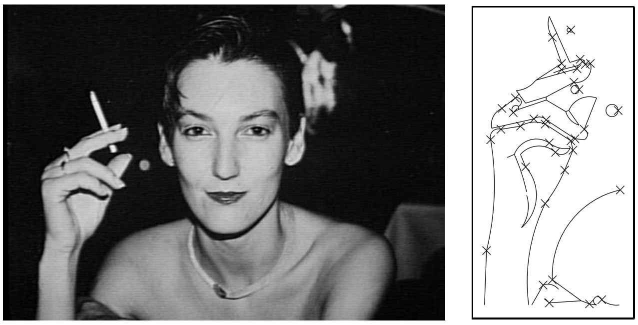

Fig. 1. Gray–level image and detail of segmentation result showing hand and shoulder. Corners and vertices are marked as crosses.

In the case of the stereo system, this approximation of chain codes as straight lines and arcs greatly reduces the number of primitives to be matched. Compared to an approximation by straight lines only the average number of primitives is lessened to 58.4% measured on a sample of 850 images [7], while the average approximation error does not increase for the lines.

4 FEATURE BASED STEREO USING LINES AND ARCS

The feature based stereo approach matches line segments and circular arcs. To support the matching step, a number of color attributes is taken into consideration to separate different groups of lines and arcs. Only if these attributes are similar for a pair of segmentation objects, this pair can be matched. The initial set of potential matches is formed by all pairs of segmentation objects with the following characteristics: One object is from the left, one from the right stereo image and the attributes of both objects are similar.

By a relaxation like algorithm [12] the set of preferred matches is computed from the initial set. Changes compared to the original algorithm are reported in [16] especially the score function for the initial potential matching is altered and the algorithm is fitted to a resolution hierarchy.

To be able to match straight line segments and circular arcs, we extended the score function by a comparison between circular arcs and straight line segments and by a comparison between pairs of circular arcs. The parameters defined for gradient and color contrast had to be defined for circular arcs. For this purpose circular arcs with more than one vertical branch are split into single vertical branches. Horizontal segments are erased since a reliable depth computation for horizontal primitives is not possible [2] using epipolar geometry.

All together the introduction of color attributes cuts the number of potential matches to less than 50%. This number is further reduced by the direct computation of the disparity on chain coded lines. Thus the matching can be computed on a rather coarse

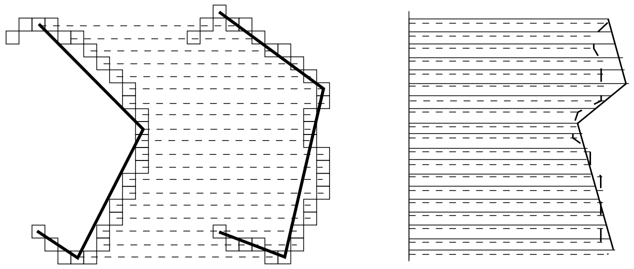

Fig. 2. Disparity computed from chain coded lines (dashed lines) and from segmentation primitives (thin solid lines).

approximation of the lines without loss of accuracy. The following algorithm was chosen:

1. Chain coded lines in the stereo image pair are computed.
2. The lines are approximated coarsely by straight line segments and circular arcs.
3. The matching is computed on the coarse approximation.
4. The common rows for each two matched segmentation primitives are determined.
5. Every primitive represents one chain coded line section. In each common row the positions of the chain coded line sections belonging to each two matched primitives are identified.
6. The disparity is computed directly from the chain code positions. In case of horizontal chain code links, the mean position of these codes is used.

An example for this algorithm is shown in Fig. 2. Thick lines show three pairs of matched segmentation primitives, each box shows a pixel of the associated line segment. The disparity computed for line segments is shown as dashed lines; the disparity computed from segmentation objects is depicted as thin continuous lines.

A result of the procedure on approximately straight lines is shown in Fig. 3; even there the number of primitives is reduced, although the real object has no curved edges.

The feature based stereo approach, described above, results in a sparse depth map with data only for the matched lines. To obtain a dense depth map, the sparse map is used to initialize the subsequent block matching algorithm working on a resolution hierarchy [2].

5 HIERARCHY OF IMAGE PROCESSING OPERATORS

The algorithms are implemented using ἵππος (HIPPOS) [14, 15], an object oriented class library designed for image analysis which is based on the commonly used NIHCL C++ class library [6]. In [14], the data interfaces were defined as classes for the *representation* of segmentation results. The *segmentation object* plays the central role here. In [7] we extend this system to a hierarchical structure of *image processing and analysis classes* and *objects*. Image segmentation, in particular line detection, is considered here as a sequence of (possibly repeated) processing steps shown in Fig. 4; the data classes in boxes are taken from [14].

The basic structure of the class hierarchy for image processing operations is shown in Fig. 6. On a coarse level, operators for line based segmentation can be divided into edge detection (arrow 1 in Fig. 4), line following (with hysteresis threshold, arrow 2), gap closing (arrow 3), and corner and vertex detection (arrow 4). For each processing step which is implemented here as a class, there exists a large variety of choices in the literature. When the whole sequence of operations is subject to optimization, either manually or automatically as in [13], it is crucial to have similar interfaces to exchangeable single processing steps, such as several corner detection algorithms. This is greatly simplified by object–oriented programming and polymorphic interfaces as shown here. Only the

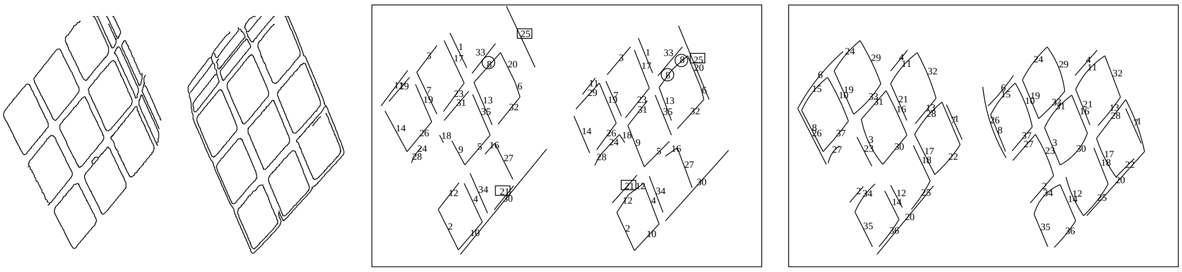

Fig. 3. Chain coded lines (*left*). Matched lines with straight line approximation (*center*), with straight line and circular arc approximation (*right*), each number referring to a segmentation primitive, equal numbers indicate matched primitives. Matching errors are marked by a box.

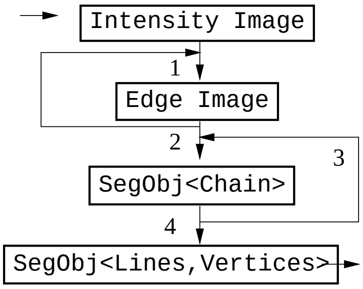

Fig. 4. Segmentation sequence

FOR	Start point TO end point				
	compute curvature of actual point				
	IF	curvature is below lower hysteris threshold			
	THEN	IF	already found possible corner		
		THEN	output this corner		
	ELSE	IF	curvature is above upper hysteris threshold		
		THEN	IF	already found possible corner	
			THEN	IF	curvature greater than curvature in potential corner
				THEN	actual point is potential corner
			ELSE	potential corner found	

Fig. 5. Hysteresis Corner Detection

type of the input and output data has to remain fixed for such a system.

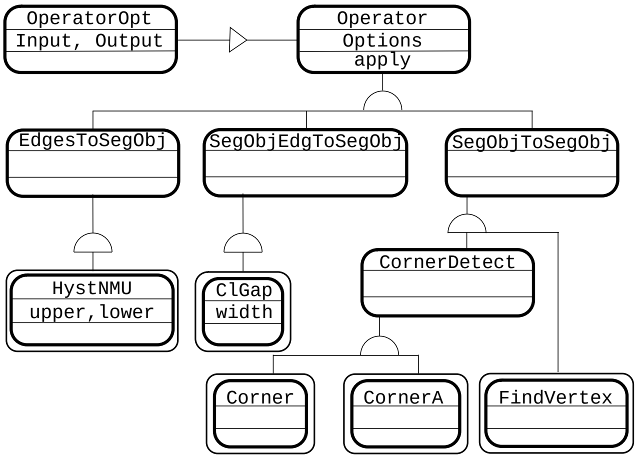

Fig. 6. Class hierarchy for image operators [7]

One particular algorithm applied here is the corner detection which is required for the approximation of chain codes by straight lines and arcs. The principle is shown in Fig. 5. The curvature to be computed is left unspecified in the class `CornerDetect` (i.e. it is a pure virtual function). The derived classes, e.g. `CornerA` or `Corner`, implement this function using the k–curvature [17] or the H–curvature which is introduced in [7].

The idea of the H–curvature is to treat the line as a parametric curve $K(t)$. The curvature in the point P_M on the line uses two points P_L and P_R on the curve. The path along the line from P_L to P_M has to be equal to the path P_M to P_R; this length is a parameter k of the approximation algorithm. The distance H between P_M and the middle point P_C of a secant through two points P_L and P_R is used as a measure for the curvature in the point P_M. This filter will be applied several times with varying parameter k to the curve to detect weak corners and stronger corners with small distances between them. This is sketched in Fig. 7; more details can be found in [7].

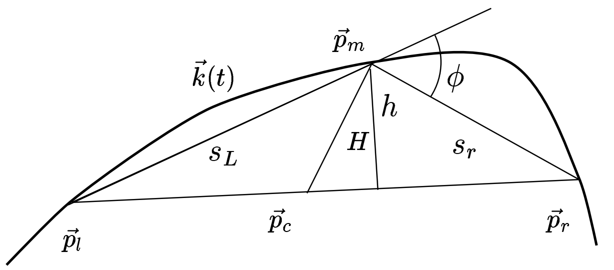

Fig. 7. Corner Detection

Of course, the split–and–merge procedure also works in a general way on either straight lines, circular arcs, or the combination of both, and leaves the actual approximation to derived approximation classes [7].

6 RESULTS

The mean disparity error for the combined stereo approach is 1.93 pixel [2]. This error is measured by matching of planar surfaces by hand and comparing this to the disparity values computed for these surfaces. An example for the disparity measured for a Rubic's cube is shown in Fig. 8.

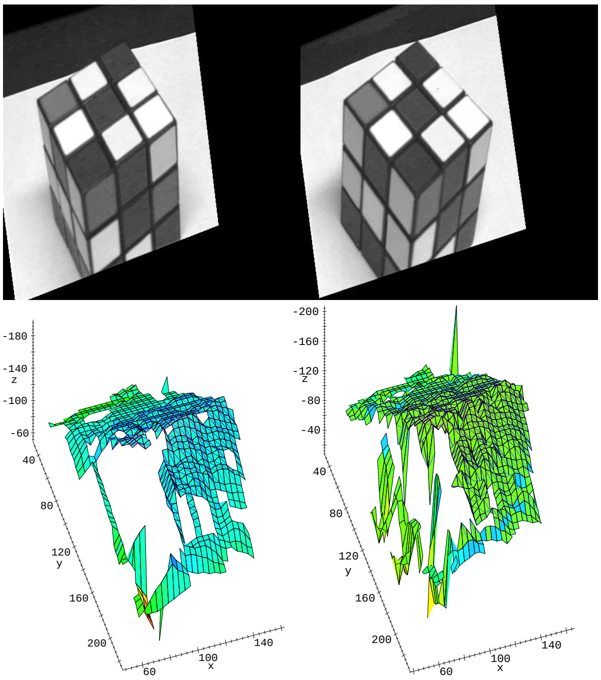

Fig. 8. Combined matching: normalized input images (top), initial depth map from line based matching (bottom left), result of combined matching (bottom right).

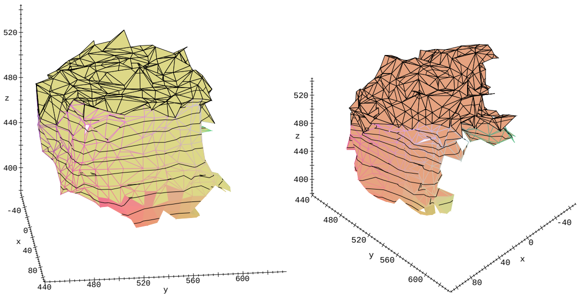

Fig. 9. Resulting depth data of the cube.

The computational time needed for the stereo algorithm to proceed one stereo image pair has been measured on a HP 735 (99 MHz) with 124 MIPS. For the Nevatia–Babu–edge filter in three color channels 3.9s are needed, for the line detection 0.7s, for the straight line – circular arc – approximation 0.9s, for the feature based matching and interpolation 7.0s, for the area based matching 38.0s, which adds to 50.5s on this single processor machine. The area based matching step was parallelized on the *M*odular *E*xpandable *M*ultiprocessor *SY*stem *MEMSY* [3]. Since data partition allows an independent computation of disparity in each line we got a nearly linear speedup for up to 16 processors.

An example for the final result of the registration and fusion of several views [1] is shown in Fig. 9. The missing side surface (left picture) was not visible in the image sequence. No surface fitting is applied here; only the unprocessed range data is visualized.

7 CONCLUSION AND FURTHER WORK

We showed a segmentation algorithm reducing the number of primitives significantly, leading to improved performance and reliability in feature based stereo matching. Furthermore the disparity is computed on chain coded lines directly thus allowing to use rather coarse approximation for matching without loss of accuracy. We explained the advantages of our object–oriented approach for developing an hierarchy of operations for image processing. Results of the complete system show that our segmentation approach overcomes the problem of inaccurate calibration and intensity changes.

Further work will concentrate on judgement and path planning to complete parts of the surface which can be seen only from certain viewing directions and on classification of the examined object.

REFERENCES

[1] R. Beß. Registering depth maps from multiple views recorded by color image sequences. In *The 4th Open Russian–German Workshop on Pattern Recognition and Image Analysis*, pages 19–24, Valday, The Russian Federation, 1996.

[2] R. Beß, D. Paulus, and H. Niemann. 3d recovery using calibrated active camera. In *Proceedings of the International Conference on Image Processing (ICIP)*, volume II, pages 855 – 858. IEEE Computer Society Press, Lausanne, Schweiz, September 1996.

[3] M. Dal Cin et al. Architecture and realization of the modular expandable multiprocessor system memsy. In *Proc. of the conf. on Massively Parallel Computer Systems — MCPS '94*, Ischia, 1994.

[4] P. Damaschke. The linear time recognition of digital arcs. *Pattern Recognition Letters*, 16(5):543–548, May 1995.

[5] J. Mundy et al. The image understanding environments program. In *Proc. of the DARPA Image Understanding Workshop*, pages 185–214, Hawaii, Jan. 1992.

[6] K. E. Gorlen, S. Orlow, and P. S. Plexico. *Data Abstraction and Object–Oriented Programming in C++*. John Wiley and Sons, Chichester, 1990.

[7] M. Harbeck. *Objektorientierte linienbasierte Segmentierung von Bildern* "Object–oriented line–based image segmentation" . Dissertation, Technische Fakultät, Universität Erlangen–Nürnberg, Erlangen, 1996.

[8] P. Koller. Computation of range images from color–stereo–images by 'simulated annealing'. In N. Pavešić, H. Niemann, and D. Paulus, editors, *Proceedings of the German–Slowenian Workshop on Image Processing and Stereo–Analysis*, pages 119–130. Arbeitsberichte des IMMD der Universität Erlangen–Nürnberg, Band 26/1, Erlangen, 1993.

[9] J. Koplowitz and S. Plante. Corner detection for chain coded curves. *Pattern Recognition*, 28(6):843–852, Juni 1995.

[10] A. Koschan. Chromatic block matching for dense stereo correspondence. In S. Impedovo, editor, *Progress in Image Analysis and Processing III (Proceedings of the International Conference on Image Analysis and Processing)*, pages 641 – 648, Singapore, 1993. World Scientific.

[11] F. Leberl, M. Gruber, P. Uray, and F. Madritsch. Trade–offs in the reconstruction and rendering of 3–D objects. In W. G. Kropatsch and H. Bischof, editors, *Tagungsband Mustererkennung 1994*, volume 5 of *Informatik Xpress*, pages 58 – 73, Berlin, 1994. Springer.

[12] G. Medioni and R. Nevatia. Segment-based stereo matching. *Computer Vision, Graphics and Image Processing (CVGIP)*, 31:2–18, 1985.

[13] H. Niemann, V. Fischer, D. Paulus, and J. Fischer. Knowledge based image understanding by iterative optimization. In St. Hölldobler G. Görz, editor, *KI–96: Advances in Artificial Intelligence*, volume 1137 (Lecture Notes in Artificial Intelligence), pages 287–301. Springer, Berlin, 1996.

[14] D. Paulus. *Objektorientierte und wissensbasierte Bildverarbeitung*. Vieweg, Braunschweig, 1992.

[15] D. Paulus and J. Hornegger. *Pattern Recognition and Image Processing in C++*. Advanced Studies in Computer Science. Vieweg, Braunschweig, 1995. second edition to appear 1997.

[16] Stefan Posch. *Automatische Tiefenbestimmung aus Grauwert-Stereobildern*. Dissertation, Deutscher Universitäts Verlag, Wiesbaden, 1990.

[17] A. Rosenfeld and M. Thurston. Edge and curve detection for visual scene analysis. *IEEE Transaction on Computers*, 20:562–569, 1971.

[18] W. B. Seales and O. D. Faugeras. Building three–dimensional object models from image sequences. *Computer Vision and Image Understanding*, 61(3):308–324, May 1995.