Recognition of a Moving Object in a Stereo Environment Using a Content Based Image Database

Attila Kiss, Tamás Németh, Szabolcs Sergyán, Zoltán Vámossy, László Csink

Budapest Tech, John von Neumann Faculty of Informatics
Institute of Software Technology
Nagyszombat u. 19, H-1034 Budapest, Hungary
Phone: +36-1-368-9840/178, Fax: +36-1-368-9632
Email: vamossy.zoltan@nik.bmf.hu

Abstract: We have developed a two-camera system that is capable to detect a moving object in the workspace. After noise removal and preprocessing the disparity map and the 3D model is produced. This detected moving object and its most important features are forwarded to a content based retrieval system that finds the most similar stored objects in the database.

The found object is surrounded by the smallest rectangle or the convex hull. To do this the feature points had to be found by the Susan and Harris type corner detection algorithms.

As we have used two cameras, the 3D description of the objects are also given by the disparity map. Considering the literature various CBIR systems have been developed, ours is a special one that can fruitfully cooperate with the stereo system.

Keywords: stereo camera system, content based image retrieval

1 Introduction

Our project involves the camera interception of the movement of simple objects, and their retrieval from the object image database.

The subgoals of the project are the development of reliable working algorithms for motion detection, space modeling and object recognition. Motion detection involves the localization of the moving object in the image so that only this image segment would have to be processed. In the space modeling phase we wish to produce the 3D-description of the objects using the images produced by the two cameras. The Content-Based Image Retrieval System (CBIRS) is planned in such a way that it should be able to learn from user-generated feedback.

The individual modules have separate applications. The modeling system, for example, can be applied for 3D-scanning; the motion detection system can be applied in security systems; the CBIRS also has several applications in its own right. Therefore, we have planned these systems so that they could be used separately as well. The idea of combining these programs in one system may have the advantage of enhancing each other's performance, e.g. the retrieval of the selected objects by motion detection in the image database.

2 Project Description

From a functional point of view the system has three parts. The first module handles the cameras, image extraction, noise filtering, preprocessing, motion detection and object segmentation. In short, this is the preprocessing module. The output of this module is forwarded to the second module – 3D stereo modeling and visualization – and to the third module, the CBIRS.

In the first module, after noise filtering, the motion detection algorithm determines the difference image, more precisely the smallest rectangle or convex polygon spanned by the difference points. The feature points in the image part are also extracted. The algorithms of this module are described in [3] and [4]. The coordinates of detected corners and other feature points and the corresponding image parts are forwarded to the 3D modeling module that finds the corresponding points in the left and right image, respectively, determines depth and builds the model. The disparity map is also produced for the CBIRS. The user may decide what information should be passed from the preprocessor to the CBIRS. The actual camera image, as well as the image part determined by the smallest rectangle or the convex hull can also be passed to the CBIRS apart from the feature points.

The system has been designed in such a way that the preprocessing module – containing the camera module and segmentation – can be considered as an independent stereo system as well; and the CBIRS can also take input from other environments, too. However; the coordination of these two systems enhances the effectivity of the CBIRS.

3 Experiments

The algorithms have been tested in Windows XP, on an AMD Athlon XP-M 2400 processor with 512 MB DDR RAM.

3.1 Motion Detection and Segmentation

Figure 1(a) shows the background we used for testing, in Figure 1(b) tan object is placed in the scene. Both have resolution 320×240 with 24-bit color. The implemented algorithms have been tested with several parameterization settings.

Figure 1
(a) Background without (b) with object

The tests showed that too much noise reduction prevents the system running in real time, and it may happen that some parts of the object are considered to be noise. Too little noise reduction, however, may result in bad performance.

3.2 Extraction of Feature Points

To extract feature points we have implemented the SUSAN [3] and Harris [4] corner detection algorithms. To compare the algorithms we use the image from [1], with resolution 240×200 with 8-bit color.

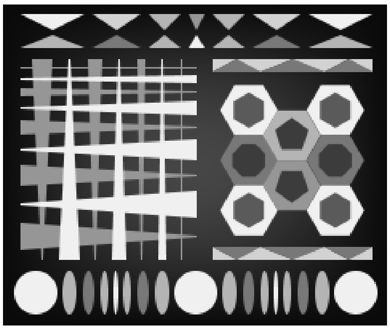


Figure 2
The test image from [1]

Method	threshold	Running time	Found corners
SUSAN	-	59 ms	6949
Harris	1000	214 ms	428
Harris	3000	229 ms	387
Harris	5000	205 ms	350

SUSAN finds too many corner points. The Harris-detector yields better results with good parameters.

3.3 Matching Feature Points

To match the feature points they first have to be found on the left and right camera images, then their neighborhoods are pair wise compared (see Figure 3).

Figure 3
Found feature points

Figure 4 shows the success of of matching described by arrows. Matching is based on the corresponding data described in Table 2.

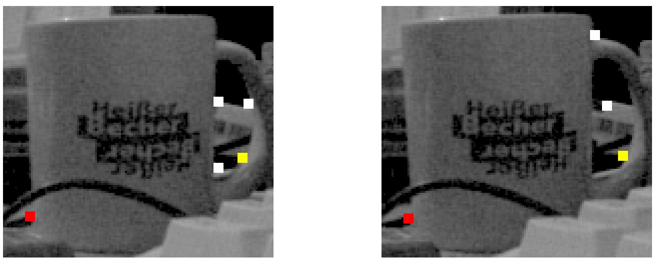


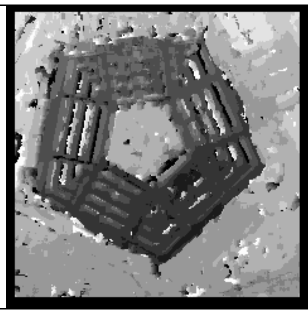
Figure 4
Corresponding points

-179229,6734	221100,6122	172082,7551	52866,73469	-234818,6326	-105583,7755
707742,3673	237053,9387	489928,2244	88376,32653	-56035,83673	-100290,1224
398228,8775	315839,0204	278842,5918	140134,2244	48111,61224	-92191,95918
141621,4285	-63785,57142	-131816,5714	175003,7142	254262,8571	32772,85714
-323333,4489	-61447,59183	186103,8367	-239630,5102	-482033,7551	-33426,18367
-151839,1836	-103672,4693	-162267,6122	-12652,16326	47502,91836	51774,06122

Table 2
Correlation values of matching

3.4 Correlation Based Modeling

In this subsection we present the results of correlation based stereo modeling. The test images can be seen at Figure 5. Darker points are closer, lighter points are further.

Test image	Disparity values	Running time	Result image
Pentagon (240x240)	-5..5	3,2 sec	

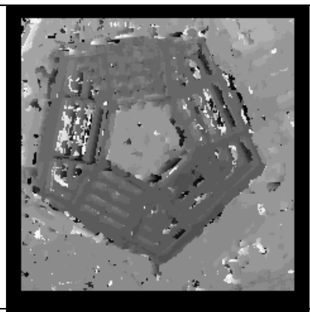
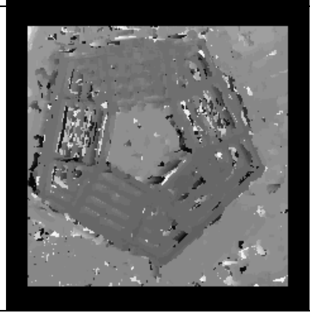
Pentagon (240x240)	-10..10	5,7 sec	
Pentagon (240x240)	-15..15	7,3 sec	

Figure 5
Pentagon stereo images [2]

With stereo images it is important to know how far the cameras are from each other and what angle their axes make. This information is necessary to find the correct disparity interval. The -5..5 interval is too small, therefore the matching of points is erroneous.

3.5 Content Based Image Extraction

While setting up the database we have tried to keep in mind that it should contain various types of images but very similar ones as well. The objects have black background and constant illumination. Some of the objects are partially occluded or are taken from different viewpoints.

In Table 5 we give a selection of algorithms with various parameters. The last two columns show the results, namely the percentage of cases when there was a similar image in the same category, or there was a similar image or in the first three places there was an image of the same category.

object 1

object 2

object 3

object 4

object 5

object 6

object 7

object 8

object 9

object 10

Table 4
Image objects

algorithm	Number of histogram cells	Color space and normalization	First	First three
DCT			41,57	54,12
Corner_HWC5*			42,16	61,76
Corner_HWC10*			33,33	53,92
Minkowski distance	4x4x4	HSV	83,33	94,12
	8x8x8	HSV	81,37	93,14
	4x4x4	LAB	82,35	92,16
	8x8x8	LAB	87,25	91,18
	4x4x4	LUV	51,96	76,47
	8x8x8	LUV	73,53	88,24
	4x4x4	RGB-not normalized	72,54	80,39
	4x4x4	RGB normalized	52,94	71,57
	8x8x8	RGB-not normalized	78,43	94,12
	8x8x8	RGB normalized	52,94	75,49
	4x4x4	XYZ	68,62	81,37
	8x8x8	XYZ	67,65	88,24
	4x4x4	YIQ	77,45	92,16
	8x8x8	YIQ	76,47	89,22

Histogram Intersection (HI)	4x2x4	HSV	95,1	98,04
	4x4x4	HSV	91,18	98,04
	8x8x8	HSV	92,16	98,04
	4x4x4	LAB	83,33	96,08
	8x8x8	LAB	91,18	97,06
	4x4x4	LUV	57,84	81,37
	8x8x8	LUV	80,39	91,18
	4x4x4	RGB	78,43	84,31
	8x8x8	RGB	85,29	93,14
	4x4x4	XYZ	75,49	83,33
	8x8x8	XYZ	77,45	91,18
	4x4x4	YIQ	79,41	95,1
	8x8x8	YIQ	85,29	94,12
	16x16x4	YIQ	45,1	73,52
	16x16x2	YIQ	89,22	96,08

Table 5

Test results with identical background and illumination

The best results were achieved by the Histogram Intersection 4x2x4 HSV. It is, of course, possible that with other images other parameters should be selected.

Table 6 shows the results of the depth test.

First	First three
71,21	86,36

Table 6

Depth test

The above results are not really dependent on illumination changes, but they rare sensible to scaling or change of background.

We have also found some typical errors. It cannot be avoided that the object and the background share some colors. This discrepancy causes 54,2% of the errors. The other errors are due to the fact there is no one-to-one map between the objects and their color histograms.

4 Results and Conclusions

We have developed a two-camera system that is capable to detect a moving object in the workspace. After noise removal and preprocessing the disparity map and the 3D model is produced. This detected moving object and its most important features are forwarded to a content based retrieval system that finds the most similar stored objects in the database.

The found object is surrounded by the smallest rectangle or the convex hull. To do this the feature points had to be found by the Susan and Harris type corner detection algorithms.

As we have used two cameras, the 3D description of the objects are also given by the disparity map. Considering the literature various CBIR systems have been developed, ours is a special one that can fruitfully cooperate with the stereo system.

We have developed a CBIRS which can automatically produce content-based indices. Using these indices, search in the hierarchically built image database is possible. Textual descriptions can also be attached to the elements of the database, and the images are categorized in a hierarchical structure; these can also be used in a search.

We have developed four color-based, two texture-based and one depth-based feature extraction techniques. Each of these can be parameterized in several ways, for example we can select the color space or the scaling of color histograms.

Semantic interpretations may be attached to the images. The position of the image in the image hierarchy also has some semantic content. This may help in narrowing down the search in some cases.

We can find nearest neighbors, or images that are more similar than a given threshold.

We have developed a weighting algorithm based on user feedback to earlier search results, which helps future search.

Each feature detection technique has been thoroughly tested with each possible parameterization.

We think that our system can be used for the tasks determined in the introduction and we hope that previous experiments have made it more stable and reliable.

References

[1] Sojka, E.: A New Algorithm for Detecting Corners in Digital Images. 2004.12.18, http://www.cs.vsb.cz/sojka/cordet/presentation.html

[2] Confidently Stable Monotonic Matching for Standard Stereo Pairs. 2004.12.18. http://cmp.felk.cvut.cz/demos/Stereo/New/Matching/smm-standard.html

[3] Smith, S. M, Brady, J. M.: SUSAN – a new approach to low level image processing. Int. Journal of Computer Vision, Vol. 23, Nr. 1, 1997, pp. 45-78

[4] Harris, C., Stephens, M.: A combined corner and edge detector. Fourth Alvey Vision Conference, 1988, pp. 147-151