REAL TIME TRACKING OF A DYNAMIC OBJECT

Amir Saeed Homainejad
Department of Geomatics
The University of Melbourne
Parkville 3052 AUSTRALIA
Telephone : +61 3 344 6806
Facsimile : +61 3 347 2916
Email : homain@sunburn.sli.unimelb.edu.au

KEY WORDS: Template, Real Time, Tracking, Automatic, Stereo Matching

ABSTRACT

This paper outlines a method for the automatic tracking of a dynamic object in real time. The method is able to position the visual system concurrent with the processing of the tracking. The visual system moves approximately along the Z axis whilst the object moves throughout the scene from the left to the right. A program in C language has been developed to position the visual system and to provide stereo matching in 30 ms. In addition, the program is able to track the object in less than 30 ms. The program is robust, reliable, and able to locate the target with sub-pixel precision. This paper will explain and describe the method and present some experimental results that achieve real time tracking.

1. INTRODUCTION

Computer vision is a technique that includes a wide range of image processing and statistical pattern classifications, and is used for vision perception. Computer vision has had a vast amount of applications; eg, pattern recognition, pattern detection, and pattern extraction. This paper will focus on object tracking which is a process that involves the above mentioned applications as well. Fundamentally, visual sensors are the main sources for this type of tracking because they enable a visual feedback into the processing. In addition, they provide information about the object and the background, and, hence, are a facility for a robot and/or an autonomous vehicle to calibrate and position itself with the background.

Visual systems based on the kind of the sensor are classified to two major groups: passive and active systems. Active systems supply their own source of energy to illuminate features of interest. Two examples are sonar and radar sensors. In contrast, passive systems require external light sources, such as sunlight.

Such systems as tracking a manipulator robot, navigating a mobile robot, self controlling manufacturing require different approaches in finding reliable methods for tracking. The methods vary, based on the use of mono cameras or multi cameras, the use of stable visual systems or dynamic visual systems, and the use of active/passive systems. The major issues that affect the mono camera method are:

- singularity,
- the exiting noise on the images that degrades the computation of the depth of a point,
- a small error in the measured coordinates of a point in the image results in a large error in the depth.

The first problem can be compensated, if at least three targets on the object are detected (Papanikolopoulos et al. 1992). The two latter problems can be overcome by imposing certain constraints. For example, imposing a constraint whereby the camera or object is moving along a defined axis will reduce depth error. In contrast, the main disadvantage of multi camera methods is slow processing. This problem can be overcome by reducing precision. Such the methods are used by Tistarelli et al. (1991), Elfes (1987), and Waxman (1987).

This paper will present a robust and reliable method of tracking a dynamic object that satisfies both aspects of fast processing and precision. In a previous experiment (Homainejad and Shortis 1995c), a stable multi camera setup was used to track automatically a dynamic object. This paper describes a new experiment, in which it is sought to answer to some basic questions. The questions are:

- At what speed can the system track a dynamic object?
- Is the program able to reposition the vision system, when the system is relocated?
- To what extent is the method robust and reliable?

To address the questions, the visual system was set up at a point in front of the test field, and a stereo image was acquired. The test field consisted the templates as explained by Homainejad and Shortis (1995a). The program initialised the processing as described by Homainejad and Shortis (1995c) whereafter; the visual system was moved to a new position. The visual system was moved approximately along the Z axis. The base line was not perpendicular to the Z axis nor coincides with the X axis. Therefore, the program should define the position of each camera separately. In the new position the visual system acquired some stereo images from the test field whilst a dynamic object was moving from the left to the right. Figure 1 illustrates the chart of the visual system and the test field.

The organisation of this paper is based on the description of the system and the explanation of its advantages. Furthermore, the paper will give the ways whereby real time was achieved. The next section will evaluate and explain the system. In addition, different approaches of programming to achieving the goals of the experiment will be assessed in Section 2, and Section 3 presents a conclusion about the test.

2. THE METHOD EVALUATION

This section presents an evaluation of the method and gives the practical results of the experiment. As mentioned in Section one, in the initial position the visual system acquires a stereo image; whereafter, the program automatically constructs a mathematical stereo model as explained in Homainejad and Shortis (1995b). Then the visual system is relocated in a new position; whereafter, it acquires some stereo images whilst a dynamic object is moving throughout the scene from the left side to the right side.

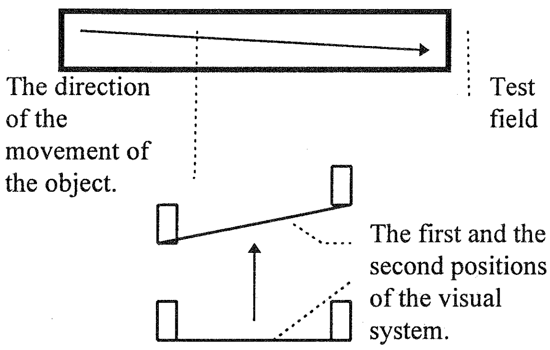

Figure 1: The chart of the movement of the visual system.

2.1 Program Evaluation

At the first step, the program defines the position of both cameras. In this stage, the program uses some constraints that are introduced to the computation. The constraints address the approximate direction of the movement of the visual system. These constraints do not limit the all the movements of the visual system; ie, the base line has some free movement in the visual system. After replacing the visual system, the base line is rotated around the Z and X axis. As a result, the program should define the position of the two cameras separately, rather than defining it only for one camera. The parameters of exterior orientation for the two cameras are presented in table 1.

CAMERA	ω	φ	k	X	Y	Z
Left	57 25' 18"	0 74' 83"	359 32' 14"	5.748	3.975	3.19
Right	359 54' 30"	357 18' 9"	0 31' 57"	6.284	3.969	3.204

Table 1: Parameters of exterior orientation for the two cameras.

If the visual system has no freedom of movement and its baseline is imposed always perpendicular to the Z axis and coincident with the X axis, then, defining the position of one camera is essential and the position of the second camera can simply be defined according to the position of the first camera. Positioning two cameras increases the time delay by at least 10 ms. The program registers the three parts of the image (that includes the three templates) into three arrays. This processing reduces the time delay by 50 ms. The longest time delay in the processing is related to the retrieval of the image into an array. It should be noted that for fast and precise processing only three control points are needed by the program. The constraints enable the program to recognise the approximate position of the three templates in the images. The reduction in delay time depends on the size of the image; ie, a larger image has a longer time delay. The size of the images in this experiment is 768x580 pixels. It should be noted that the processing is carried out on a SUN SPARC 20 work station. The images are in TIFF format, and the TIFF library was used for image processing. Other methods for restoring and reading the images were tested but were slower than the TIFF library. For example, reading and restoring TIFF images using the C library is 150 times slower than the TIFF library; therefore, it is preferred to use the TIFF library to be used in this experiment.

In programming, **Loop** is another characteristic that increases the time delay. The processing of over 20000 characters in the program are common for positioning of the left and the right cameras. In order to process these 20000 common characters, four approaches are tested. As the first approach, a **goto** command was used for the processing, but the time delay was more than 100 ms. The **goto** command was replaced by the **while** command in the second attempt, and the time delay improved by 40 ms. In the third attempt, the **for** command was tested and the time delay improved by 2 ms. In the fourth attempt, the loop was removed and in conclusion, 20000 characters are increased to the program. As a result, the time delay for the positioning of the two cameras and stereo matching was improved to 30 ms in the fourth approach. The process of the positioning and stereo matching is pseudocoded in Figure 2.

```
while ( The condition is true ) {
        for ( i=0;i<n;i++){
          Retrieve the part of image and restore into
          the array according to the constraints.
          Measure the coordinates of the coordinates
          of control points and read their number.
        }
        Compute the position of the camera and the
        external parameters.
        if ( rigth image ) break;
}
Compute stereo matching.
```

Figure 2a: The pseudocode for the second approach.

```
if ( left image ) {
        for ( i=0;i<n;i++){
          Retrieve the part of the image and restore
          into the array according to the constraints.
          Measure the coordinates of the control
          points and read their number.
        }
        Compute the position of the left camera and
        exterior parameters.
}
if ( right image ) {
        for ( i=0;i<n;i++){
          Retrieve the part of the image and restore
          into the array according to the constraints.
          Measure the coordinates of the control
          points and read their number.
        }
        Compute the position of the right camera
        and exterior parameters.
}
Compute stereo matching.
```

Figure 2b: The pseudocode for the fourth approach.

Table 2 compares the time delay for the four approaches. The result is reliable, robust and precise. It is reliable and robust because the program without any mistakes and errors recognises a control point in the right image that matches with the control point in the left image. The processing is precise because the program measures the coordinates of the control points with sub-pixel accuracy. In addition, by using the intersection method the coordinates of the control points in the object system are measured with accuracy less than 1 mm. The RMS error of the computing coordinates on the images is sub-pixel. Table 3 shows the differences between the measured and computed control points in the left and right images.

APPROACH	*TIME DELAY*
1 (GOTO)	about 100 ms
2 (while)	70 ms
3 (for)	68 ms
4 (without loop)	30 ms

Table 2: Comparison the time delays of four efforts.

PK	*Left image*		*Right image*	
	x	*y*	*x*	*y*
10	1.30E-06	(-1E-06)	-5.80E-06	(-2.1E-06)
11	(-4.7E-06)	(-6E-06)	(-7E-06)	-4.80E-06
14	(-5.81E-05)	-1.50E-06	0	(-2E-07)

Table 3: The differences between the measured and computed coordinates of control points for both the left and the right images. The unit is pixel.

Once the processing of the camera position and the stereo matching are completed, the program continue to track the dynamic object. The object is considered a rigid body. According to Hunt (1987), a rigid body does not change its shape or its size; it does not stretch, compress, twist, bend, or deform. The object in this experiment is a plastic bottle with a waist. The waist depth is about 2 cm. Figure 3 shows the object in the test field.

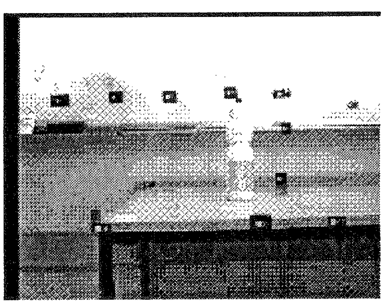

Figure 3: Illustration of the object in the test field.

The strategy of Homainejad and Shortis (1995c) is used in this experiment. A very important issue for the program is to define a point upon the object that is recognisable in both the left and the right images. The colour of the object that is used in this experiment is white and light grey and is very similar to the background colour. The program for recognising a common point in the two images should apply a sharpness mask to the images and this processing increases the time delay. For extracting the pattern from the right image that matches the pattern in the left image, an empirical threshold is used by the program. The program defines a weight based on the histogram of a common area in the left and the right images. According to the weight and the area that already is extracted from the left image, the area of interest in the right image is recognised and extracted. Figure 4 demonstrates two areas of interest in the left image and its matched area in the right image. The processing is very precise and reliable; however, recognising the area of interest is very difficult. The ability of the program for recognising different patterns is tested by selecting different points in the waist and other area of the object. Program recognises and extracts two different points in the waist area and other area upon the bottle. Table 4 presents the coordinates of the two points in the waist and a pattern upper than waist area. These points are in a very difficult areas that even an expert operator can not simply extract and recognise those points. But the weight enables the program to recognise and extract even that difficult points without applying any mask and human supervision. The weight is not constant for a sequences stereo images because lighting is not constant for all stereo images. In other words, each stereo image has a weight. Table 5 demonstrates the weights for three stereo images. When the program defines a weight for each stereo image, the time delay is increased for each stereo image by 30 ms. The time delay for extracting a common area from a stereo image is between 30 ms to 60 ms. This delay of time could be reduced if the weight is constant for all stereo images. When considering table 5, the point in the waist area has 10.6 mm difference in depth with the other point.

POINT	X	Y	Z
The waist area	6.128	3.968	4.576
The second area	6.123	3.972	4.566

Table 4: The coordinates of two points upon the object.

	1st stereo	*2nd stereo*	*3rd stereo*
weight	1.2	1.26	1.33

Table 5: A list of three weights for three stereo images.

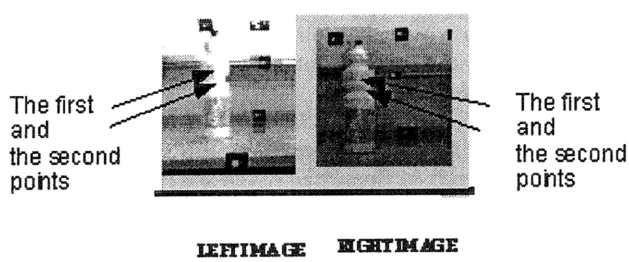

Figure 4: The left and the right images of a stereo image with areas of interest for matching.

3. CONCLUSION

This paper presents a method for tracking of a dynamic object when the stereo vision is relocated. The method is reliable, robust and precise. A program in C language is developed for automatic stereo matching and positioning. In addition, program can automatically define and track a dynamic object. The lighting is the main issue in the processing, but the applied weight reduces significantly the lighting issue. The time delay for each processing is, at best, in the order of 30 ms. The majority of time delay is related to the image registration in an array. It seems that a special image library is necessary for digital photogrammetry that it reduces the time delay. The image library should satisfy the application of real time processing for digital photogrammetry. The TIFF library is very reliable and trustable. Table 6 presents the time delay for different processes.

Retrieval image into the array	Positioning and stereo matching	Tracking
Using TIFF library 10 ms.	Only positioning for one camera 10 ms.	Tracking without considering the weight 30 ms.
Using C library 1.5 s.	All of processes for two cameras 30 ms.	tracking with considering the weight 60 ms.

Table 6: Time delays for different processes.

Further development of the program will enable the position of the visual system without the need to apply the constraints used in this experiment. The templates are very important for automatic stereo matching and positioning. The templates enable the program to carry out the positioning, the stereo matching and tracking precisely. The weighting for a stereo image improves automatic pattern recognition; and it is successfully tested for several stereo images. The outputs are very reliable, robust and precise.

4. REFERENCES

Elfes, A., "Sonar-based real-world mapping and navigation", IEEE Journal of Robotics and Automation, Vol. RA-3, No. 3. pp 249-265, June 1987.

Homainejad, A. S. and Shortis, M. R., "Development of a Template for Automatic Stereo Matching." ISPRS Intercommision Workshop. From Pixels to Sequences, Zurich, Switzerland, March, pp 318-322, 1995a.

Homainejad, A. S. and Shortis, M. R., "The Implementation of an Automatic Stereo Matching for Digital Photogrammetry" 21st Australian National Survey Conference, September, pp 220-231, 1995b.

Homainejad, A. S. and Shortis, M. R., "Stereo Vision System for Tracking a Dynamic Object" SPIE, Videometrics IV, 22-26 October, pp 264-271, 1995c.

Hunt, K. H., "Kinematic Geometry of Mechanisms", Clarendon Press, Oxford, p 3, 1978.

Papanikolopoulos, N. p., Nelson, B., Khosla, P. K., "Full 3-D tracking using the controlled active vision paradigm" IEEE int. Symp. on Intelligent Control (ISIC-92), pp. 267-274, August 1992.

Tistarelli, M., Grosso, E., Sandini, G., "Dynamic stereo in visual navigation" Proceedings IEEE Computer Society Conference on Computer Vision and Pattern Recognition, pp 186-193, June 1991.

Waxman, A. M., LEMoigne, J. J., Davis, L. S., Srinivasan, B., Kushner, T. R., Liang, E., Siddalingaiah, T., " A visual navigation system for autonomous land vehicles", IEEE Journal of Robotics and Automation, Vol. RA-3, No. 3. pp 124-141, April 1987