A Coded Visual Marker for Video Tracking System Based on Structured Image Analysis

Takeshi Kawano
Graduate School of
Science and Technology,
Kobe University
Nada, Kobe, Japan
takeshi@ai.cs.scitec.kobe-u.ac.jp

Yoshihiro Ban
Information Science
and Technology Center,
Kobe University
Nada, Kobe, Japan
ban@kobe-u.ac.jp

Kuniaki Uehara
Graduate School of
Science and Technology,
Kobe University
Nada, Kobe, Japan
uehara@kobe-u.ac.jp

Abstract

This paper proposes a new kind of visual marker for a wearable computer based Augmented Reality (AR) system. Various marker systems have already been developed for using with an AR system. However, the improvement is needed to use these marker systems on a wearable AR system in terms of the amount of information or the width of the recognizable area. The main feature of our visual marker is a large amount of information that it can possess. Our marker consists of 32 bits data area with error detection code. This feature enables a wearable AR system to be used in various ways.

1. Introduction

Recently, as the computer technology advances, a wearable computer has been becoming widely available. Wearable computers are often deployed in Augmented Reality (AR) systems. In such systems, markers are often used for rendering virtual objects into the real space. There are various marker tracking systems already exist in the market. In [1], the following systems are evaluated and compared. These are ARToolKit[2], HOM marker system by Hoffman at Siemens AG, IGD marker system used by ARVIKA [3] and SCR marker system developed at Siemens Corporate Research. Another system is the 2D matrix code system developed by J.Rekimoto[4]. These marker systems contain enough amount of information and possess good recognition accuracy. These systems are suitable for using on an AR system. However, we observe that a marker system may have various usages if the system possesses large amount of information.

In this paper, we describe a newly developed marker system which possesses large amount of information. Our system also possesses good recognition accuracy, long recognizable distance and enough processing speed for a real application.

2 The proposed marker

An example of the marker proposed in this paper is shown in Fig.1(a). Fig.1 (b) shows the marker divided into 50 areas. 5×5 blocks are divided diagonally. The resulted 50 triangles areas can be either black or white as in a binary code, so the length of this marker's data code is 50 bits. The orientation recognition is assigned to 3 blocks (6 bits) out of the 4 blocks of the corners. 12 bits between these blocks are used for the error detection. The rest of 32 bits is the data area. The main feature of our marker system is the large amount of information. Our marker can store 32 bits of information with a parity check function.

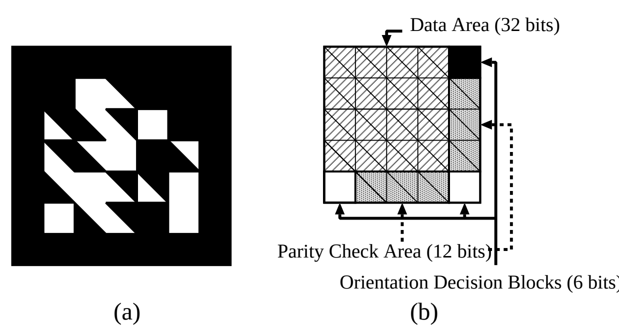

Figure 1. an example of the proposed marker and the distribution of the marker

3 Recognition algorithm

3.1 Preprocessing capturing image, binarization and labeling

Our system adapts the Pyramid method that uses a quarter compressed image for improving the processing

Proceedings of the Second IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR '03)
0-7695-2006-5/03 $17.00 © 2003 IEEE

speed of our system. So, first we capture the video images both of VGA (640×480 pixels) and QVGA (320×240 pixels) in the YUV format. Then, by picking up only the Y element (brightness) from these images, gray scale images are obtained.

Next, the edge process is performed on the QVGA gray scale image by using the Sobel operator. At the same time, the image is binarized by a fixed threshold. Next, labeling is performed on the binary image. Then, we reject label groups whose size is too large or too small or those touching the outside frame.

3.2 Marker detection

We treat as markers, the label groups whose shape is a rectangle and those in which it is possible to recognize the orientation of the label group. Therefore, first the corner detection is performed and rectangle label groups are detected. Next, orientation recognition is performed.

In the corner detection, we track outlines of label groups. Curvatures are calculated in all points on the outline. In doing so, we obtain some groups in which curvature is over a threshold. We consider points in the local maximal value as corners. And, we consider the label groups that have 4 corners as rectangles.

In the orientation recognition, we check the 4 blocks on the corners inside the code area of a rectangle label group. In our marker, for the orientation recognition, the bottom left and right blocks are represented by white blocks and the upper right block is represented by a black block. Then, if we can detect the order of $\{*, white, white, black\}$, we can recognize the orientation of the marker on the video image. And, we can consider the label group as a marker.

3.3 Code identification

For the code identification, we use the VGA image. So, we convert the coordinates of the 4 vertexes of the marker from the QVGA coordinate system to the VGA one. And, the VGA gray scale image is binarized by the threshold calculated by the discrimination analysis method. Here, as preprocessing, we derivate accurate grid lines dividing the code area. First, the edge process is performed on the VGA binary image. For each 12 grid lines, a line passing through most of the edge elements in the vicinity is newly determined as a grid line.

When we identify the code of the marker, we acquire 2 bits of data from each of the 25 quadrangles sectioned by the grid lines. And, a total of 50 bits data is acquired. In the code identification, in each quadrangles, a 2 bits data is determined by rates and positions of gravity points both of black and white elements. We call this 50 bits data, the marker's "code value".

3.4 Error Detection

The error detection is performed in order to check whether the code value is correct, or not. Here, only an error detection is performed, and an error correction is not performed. The parity check is performed by using 12 out of 50 bits.

4 Performance Evaluation

Here, we describe the performance evaluation of our marker system. The evaluation environment is following wearable AR system. The processing device is a wearable PC (Xybernaut MA V: CPU Intel Mobile Celeron 500MHz, SDRAM 256MB). The image capture device is an IEEE camera (Sony C102T: VGA, 30fps). The OS is Linux, and libdc1394 library is used for controlling IEEE1394 camera. Our system works from 14 fps (no marker) to 6 fps (10 markers). Our system can recognize markers of the size of 50×50 pixels on the captured image. In the case of using 5×5 cm size marker, the recognizable distance is 75 cm.

5 Conclusion

In this paper, we proposed a marker tracking system which is suitable for using with a wearable AR system. We introduce the marker system with a satisfactory amount of information of 32 bits. The recognizable distance is enough to be used in real applications. We observe that the processing speed is also within the required range for the actual application. However, the stability recognized by the markers is unsatisfactory. In the future work, we aim to further improve the recognition stability, the recognizable distance and the processing efficiency of our marker tracking system.

References

[1] Xiang Zhang, Stephan Fronz, Nassir Navab: Visual Marker Detection and Decoding in AR Systems: A Comparative Study, Proc. of ISMAR'02, 2002.

[2] ARToolKit: http://www.hitl.washington.edu/research/shared_space/download/

[3] ARVIKA: http://www.arvika.de/www/index.htm

[4] Jun Rekimoto: Matrix: A Realtime Object Identification and Registration Method for Augmented Reality, Proc. of APCHI'98, 1998.