I_053

Object Detection by Using Marker Assistance

Qi Zhang†, Jun Ohya†, Shunichi Yonemura‡

ABSTRACT

This paper proposes a solution for PC troubleshooting which can be used in the visual call center. The operator and user communicate not only by audio channel, but also by visual channel. To protect users' privacy, a method for eliminating objects except the users' PC from each frame of the video stream is needed. First, the marker that is to be attached to the PC is designed so that the marker is rotation-invariant and well-contrasted. After detecting candidate regions for the markers in each frame, a modified Chamfer Matching Algorithm is applied so that the markers are detected. Experimental results demonstrate the effectiveness of the proposed method.

1. Introduction

Nowadays, PC producers provide their services via visual call centers to solve troubles from the users. This service is useful for the users, but a serious problem is that most of novice users cannot describe their problems correctly when they are talking with the operators. To deal with that problem, we propose a new solution that uses video data obtained by the handy camera at the user's hand. By watching the video images sent from the user's camera on the monitor, the operator can know what is happening. However, it also brings us a new issue. When an operator looks at the screen, he/she can see everything within the image including the privacy of the user.

We propose a method that can protect users' privacy; more specifically, the video stream is processed so that the privacy is not visible to the operator. In previous literatures, various approaches have been proposed, but most of these works focused on the effectiveness, but not efficiency. To conquer this difficulty, a marker-based region extraction is discussed in this work.

Marker recognition is frequently applied in various issues. However, for our application, to achieve the desirable computational efficiency, the markers should be well-designed. In addition, we must find an effective algorithm which can detect the markers quickly and accurately. The experimental results prove that our algorithm is fast and accurate enough so that it is not necessary to use another method for tracking. What we need is just to find the object in each frame.

The rest of this paper is organized as follows. In section 2, we discuss the design of the marker. Section 3 describes the algorithm of detecting and tracking. Experimental results are presented in section 4, followed by conclusions in section 5.

2. Marker Design

Implementing the marker-based algorithm to the application of visual call center, the pattern and color of markers should be considered. The pattern of the markers should be well-designed so that it can be recognized quickly. On the other hand, the markers should also be unique so as to be distinguished from other objects.

We defined the background color as yellow (255, 255, 0), cyan (0, 255, 255) or pink (255, 0, 255). These colors have high brightness and special constitution so that it is easy for us to build color filters for them.

We have to consider how to deal with the rotation and scale because the pose of the camera is unknown. The marker is designed that there is a round area at the center of it, because this kind of pattern is rotation-invariant. The area is with dark color, so that the circle can be easily detected because of the strong contract between that area and the background. The designed marker is shown in Fig. 2(f). These markers are attached to some corners of the PC's body.

3. The Proposed Algorithm

We define a color filter to find out candidate regions and use a method called "chamfer matching algorithm" to match the pattern of the markers in each region.

3.1 Color Filtering

The detection of the markers should be performed by a template matching. However, the calculation of template matching is time consuming. For accurate computation, the matching should be done only in several small areas. The detection of candidate marker regions is calculated by a traditional method, as follows.

A combined color filter is used to search the candidate areas. First of all, we search the regions whose color is similar to the background color of the marker. If the center of a region is with the similar dark color as the marker, the region is chosen as a candidate. Fig. 2(b) shows an example after color filtering.

3.2 Chamfer Matching Algorithm

To decide whether a candidate is a marker or not, we use chamfer matching algorithm to search the pattern of the markers.

Chamfer matching algorithm (CMA), which was first developed in 1977[1], is a technique for finding the best fit of edge points from two different images, by minimizing a generalized distance between them. The original version of the chamfer matching algorithm is useful only in a limited number of applications. It is a fine-matching method, which needs a good start hypothesis of the transformation that brings the edges into correspondence. That method was evolved to a universally useful edge matching algorithm in later years. The most famous one called "Hierarchical Chamfer Matching Algorithm" (HCMA) was developed in 1988[2]. Recently, some researchers utilized this method for human detecting [3].

HCMA is a stable matching method which has the ability to handle imperfect (noisy, distorted, etc.) data. We propose a modified version of CMA to fit the application of visual call center.

3.2.1 Distance Transform

CMA is a matching algorithm based on distance transform (DT). The template is a binary image called polygon, which is composed of feature points. A special version of distance

† Waseda University

‡ NTT

transform called 3-4DT is conducted in the original image. Distance transform is conducted in the target image. The computation of 3-4DT can be expressed as follows:

$$v_{i,j}^{k} = \min(v_{i-1,j-1}^{k-1}+4, v_{i-1,j}^{k-1}+3, v_{i-1,j+1}^{k-1}+4, v_{i,j-1}^{k-1}+3, v_{i,j}^{k-1}, v_{i,j+1}^{k-1}+3, v_{i+1,j-1}^{k-1}+4, v_{i+1,j}^{k-1}+3, v_{i+1,j+1}^{k-1}+4)$$

where $v_{i,j}^{k}$ is the value of the pixel in position (i, j) at iteration k. The iteration continues until no value changes. Fig. 2(c) is the image before distance transform and Fig. 2(d) is the result of 3-4DT.

After the distance transform, the template is superimposed on the distance image. An average of the pixel values that the polygon hits is the measure of correspondence between the edges, called the edge distance. Fig. 1 shows the principle of the computation.

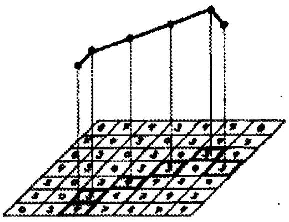

Fig. 1 Computation of the edge distance

The algorithm is computed by minimizing the edge distance. To make this minimization as simple as possible, the edge distance function should be smooth and convex. There are many variances of matching measure averages, e.g. arithmetic, root mean square and median. In our system, the root mean square average is used as matching measure (See [4] for other matching measures.)

$$\frac{1}{3}\sqrt{\frac{1}{n}\sum_{i=1}^{n} v_i^2}$$

where v_i are the distance values and n is the number of points in the polygon. The average is divided by 3 to compensate the unit distance 3 in the 3-4DT.

In fact, CMA contains also other computations, but for our case, this is robust enough.

3.3 Template Generation

It is difficult to use a certain template for the pattern matching, if no a priori knowledge of the distance between the camera and the object is obtained. Templates are generated automatically and each of them is used for matching.

The template is easy to be generated since the pattern of marker is designed to be simple, here a circle. The size and location of every candidate region is obtained by the previous computation, so the center of each circle can be calculated. The maximum radius of each circle can be decided according to the size of the region. In practice, instead of all the points, only some pixels of the circle are generated.

3.4 Noise Elimination

Even though we search the pattern in every candidate region; it is probable to result in false matching at non-marker regions. To identify the markers, the variance of all the regions is calculated. Since the size and pattern of each marker are the same, we can get the marker regions by minimizing the variance.

4. Experimental Results

Experiments were carried out to testify the computational efficiency and the effectiveness. The video images whose size is 320 by 240 pixels are used to conduct the experiments. Note that the algorithm gets slower when the noise increases, but even under the worst condition, all the calculation for each frame can be done within 32 milliseconds. It is fast enough for this application.

As for the accuracy, it remains above 80% even if there are a lot of noises within the image. Fig. 2(a) is the input image; the result is demonstrated in Fig. 2(e). As can be seen, in order to protect users' privacy, mosaic was overlaid to non-PC areas in the original image.

5. Conclusion

In this paper, we proposed a new solution for PC troubleshooting via visual call center in means of the protection of users' privacy. The computational efficiency and the effectiveness are proved through the experimental results.

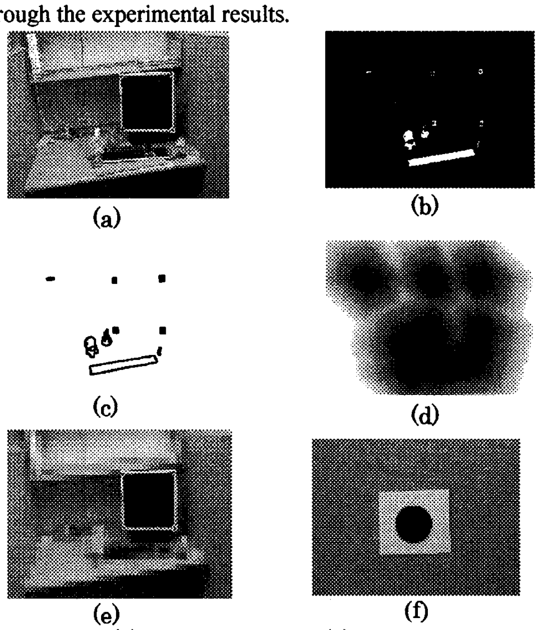

Fig. 2 (a) original image (b) image after color filtering (c) edge extracted (d) 3-4DT distance image (e) final result (f) marker

References:

[1] H.G. Barrow, J.M. Tenenbaum, R.C. Bolles and H.C. Wolf, "Parametric correspondence and chamfer matching: Two new techniques for image matching", in Proc 5th Int. Joint Conf. Artificial Intelligence, Cambridge, MA 1977, pp. 659-663

[2]Gunnila Borgefores, "Hierarchical chamfer matching: A parametric edge matching algorithm", IEEE transaction on Pattern Analysis and Machine Intelligence, Vol. 10, no. 6, pp. 849-856, 1988

[3] Liang Zhao, "Closely Coupled Object Detection and Segmentation", ICCV, pp. 454-461, Tenth IEEE International Conference on Computer Vision (ICCV'05) Volume 1, 2005.

[4] Abdul Ghafoor, "Robust Image Matching Algorithm", EC-VIP-MC 2003, 4th EURASIP Conference focused on Video / Image Processing and Multimedia Communications, 2-5 July 2003, Zagreb, Croatia