Target Marker: A Visual Marker for Long Distances and Detection in Realtime on Mobile Devices

Oliver Christen, Edwin Naroska, Alexander Micheel
Niederrhein University of Applied Sciences, Institute for Pattern Recognition
Reinarzstr. 49, 47805 Krefeld, Germany
oliver.christen@hs-niederrhein.de; edwin.naroska@hs-niederrhein.de;
alexander.micheel@hs-niederrhein.de

Shanq-Jang Ruan
National Taiwan University, Dept. of Electronic and Computer Engineering
No. 1, Sec. 4, Roosevelt Rd., Taipei 10617, Taiwan (R.O.C.)
sjruan@mail.ntust.edu.tw

Abstract- This paper deals with the design, detection and recognition of a visual marker which can be detected and recognized over a long distance in realtime on a mobile device. The solution is based on a rotationally symmetric marker, a detection process based on estimated circle and ellipse parameters and a Hidden Markov Model based approach to verify and classify possible marker candidates. The solution can be used in augmented reality and localization applications.

Keywords: visual marker detection, visual marker decoding, augmented reality, hidden markov model, mobile devices

1 Introduction

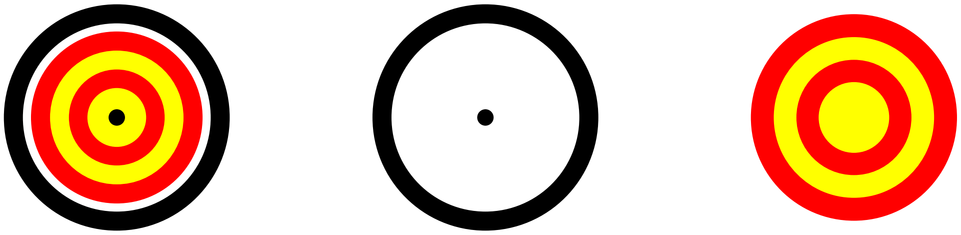
For augmented reality applications and other similar problems in computer vision, detection of visual markers within a scene is important. Unlike other scenarios we are dealing with applications where objects in the real world are equipped with markers and should be detected from a far distance to display navigation information to the user using a normal smartphone or tablet camera sensor.

Several fiducial marker systems have been proposed in the literature, most of them are based on square markers for use in the augmented reality community (Zhang, Fronz & Navab 2002). The markers usually encode an unique identification pattern that also can include information for error detection and correction. They also allow to extract the camera pose from their four corners.

A circular marker for close-range photogrammetry was proposed by V. A. Knyaz (1998) where the data is modeled as points which are located on the inner side of the outer ring. The detection algorithm works well in a controlled environment but would led to false positive results in a natural environment. A Circular Data Matrix Fiducial System proposed by Naimark & Foxlin (2002) is also using a black and white circular marker. The algorithm is not suitable for us because the image needs to be scaled down for speed and marker candidates are filtered based on their size. The multi-ring color fiducial marker approach by Youngkwan Cho Jongweon & Neumann (1998) is also using subsampling to reduce the runtime. Because of the proportional ring width the marker gets considerably larger for storing more information.

The problem of all these solutions is, that the detection and recognition algorithms do not work well if the marker is far away from the camera or the marker is very small regarding the whole image size respectively. To detect such a marker it is necessary to process the image on the highest possible resolution so that the marker information is not lost by subsampling. At the same time, we require the detection and recognition algorithm to be fast and efficient so that it can be used on mobile devices with limited resources like smartphones, tablets and smart glasses without draining too much energy. Also the detection process should filter out false positive marker candidates in complex outdoor scenes. For other applications it should also be possible to detect the marker at a very close range so that the algorithm can not make assumptions about the size or position of the markers to find.

Our solution is based on a circular marker, where the identification is encoded in concentric rings which can be recognised by an approach based on Hidden Markov Models and a detection process based on the estimation of circle and ellipse parameters to reduce the image data and filter out false positives very efficiently.



(a) Marker (b) Components used for detection (c) Components used for recognition

Fig. 1: Components of the “target” marker

2 Marker design

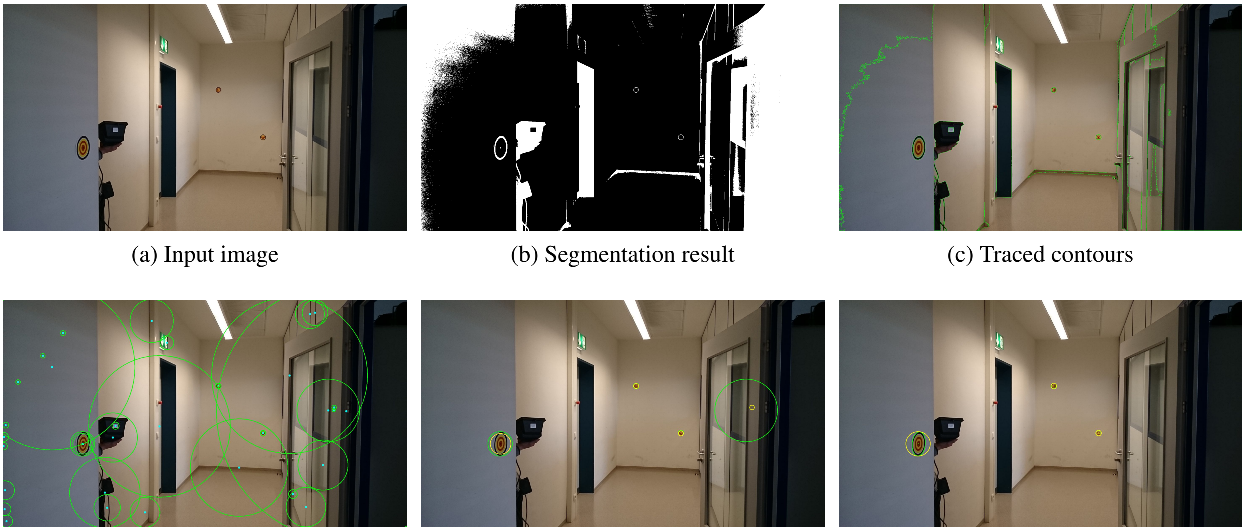
The proposed marker is a n-fold rotational symmetrical circular object which can be divided into two parts (see Fig. 1). The border consists of a black colored ring which is used in the detection phase to find marker candidates within the image. The inside of the marker consists of multiple color rings. These rings, where each ring can be of a different color, is used for holding information. This gives the possibility to distinguish between different marker objects which is used during the detection phase for further verification of the marker candidates and in the recognition phase to classify the color pattern.

3 Marker detection

The detection phase consists of multiple steps. First, the image is segmented into black and non-black areas. Then a contour tracing is performed on the resulting binary image. In the next step circle and ellipse parameters are calculated based on the contours found in the last step. Based on these parameters a filtering for false positives is performed. After that the color information from the color rings are extracted and further filtering is performed using this information. The result of each segmentation step is shown in Fig. 2 and is explained more detailed in the following sections.

3.1 Segmentation

The goal of the segmentation step is to segment the black border ring of the marker. The used algorithm is based on the color space quantization method by Pun & Wong (2008). They transform the cylindrical HSV space to a conical HS’V space via $S' = S \cdot V$ and then quantize the color space. Because only the black areas are interesting in this case, a modified version of the algorithm is used. The color space can be divided



(a) Input image

(b) Segmentation result

(c) Traced contours

(d) Guessed circles from contours

(e) Circles after clustering

(f) Circles and ellipses after filtering

Fig. 2: Steps of the segmentation phase

into color and greyscale regions by putting a threshold on the S' channel. The black regions can further be extracted by a threshold on the V channel.

In summary, a pixel x of the image I (in HSV color space) is classified in the resulting image J as foreground pixel c_f or background pixel c_b given the following formula:

$$J(x) = \begin{cases} c_f \,\text{if}\, \overbrace{I(x)_s \cdot I(x)_v < T_1}^{color/greyscale} \wedge \overbrace{I(x)_v < T_2}^{greyscale/black} \\ c_b \,\text{else} \end{cases} \tag{1}$$

On the binary image J the contour tracing algorithm by Suzuki & Abe (1985) is performed, resulting in the outer and inner contour of each component consisting of connected c_f pixels.

3.2 Circle and ellipse estimation

During the contour tracing step the assumption is made that every contour describes an circle c or is a part of a circle due to a broken contour. Therefore the corresponding parameters (radius c_r and center c_x, c_y) are guessed from the contour points utilizing the following prodecure.

We assume that the pixels of the contour lie on the circumference of the circle. From each contour three points are choosen. The point with the minimum y-value, the point with the maximum y-value and the point between these two points. The center point is determined by solving the equations for c_x and c_y:

$$(x_1 - c_x)^2 + (y_1 - c_y)^2 = r^2 \tag{2}$$

$$(x_2 - c_x)^2 + (y_2 - c_y)^2 = r^2 \tag{3}$$

$$(x_3 - c_x)^2 + (y_3 - c_y)^2 = r^2 \tag{4}$$

$$c_x = \frac{y_2\left({y_3}^2-{y_1}^2+{x_3}^2-{x_1}^2\right)+y_1\left(-{y_3}^2-{x_3}^2+{x_2}^2\right)+{y_1}^2 y3-{x_2}^2 y_3+{x_1}^2 y_3+{y_2}^2\left(y_1-y_3\right)}{2x_2 y_3-2x_1 y_3+\left(2x_1-2x_3\right)y_2+\left(2x_3-2x_2\right)y_1} \tag{5}$$

$$c_y = \frac{x_2\left({y_3}^2+{x_3}^2-{x_1}^2\right)+x_1\left(-{y_3}^2-{x_3}^2\right)+\left(x_1-x_3\right){y_2}^2+\left(x_3-x_2\right){y_1}^2+{x_1}^2 x_3+{x_2}^2\left(x_1-x_3\right)}{2x_2 y_3-2x_1 y_3+\left(2x_1-2x_3\right)y_2+\left(2x_3-2x_2\right)y_1} \tag{6}$$

$$r = \sqrt{(x_1-c_x)^2+(y_1-c_y)^2} \tag{7}$$

Each fitted circle is treated as a marker candidate. This results in many false positives because not every contour corresponds with an circular shape in the image. These are filtered out in the following steps.

It can be noticed that a true marker candidate results in at least two circles where the circles have almost the same center point. This is due to the circles guessed from the inner and outer contour of a complete segmented circle or because a similar circle was guessed from different broken contour fragments. Given this observation the image is partitioned into cells and the guessed circles within each cell are clustered based on their center point. Only cells with at least two clustered circles are further investigated and only the circle with the biggest radius in each cell is kept.

In real scenes the approximation of the marker with a circle is not exact. A perspective view shot results in a perspective distortion so that the marker is not a perfect circular object but rather an ellipse. Because of that, the parameters of the ellipse e with e_{cx}, e_{cy} (center point), major and minor axis e_a, e_b and the rotation angle e_α are also guessed from the corresponding contour points of the circle using the method by Chang, Chen & Lu (2004).

Further filter criterions are derived by comparing the circle and ellipse parameters. It can be noticed for a true marker candidate, that the center point of the circle and the center of the corresponding ellipse are very close together (see Fig. 3a) and for false positive marker candidates further apart:

$$\sqrt{(c_x-e_{cx})\cdot(c_x-e_{cx})+(c_y-e_{cy})\cdot(c_y-e_{cy})} > T_3\cdot(c_r)^{T_4} \tag{8}$$

The ratio between the circle radius and the ellipse axis is used as another filter criterion because it is unlikely to be very high or low even under perspective distortion of the marker.

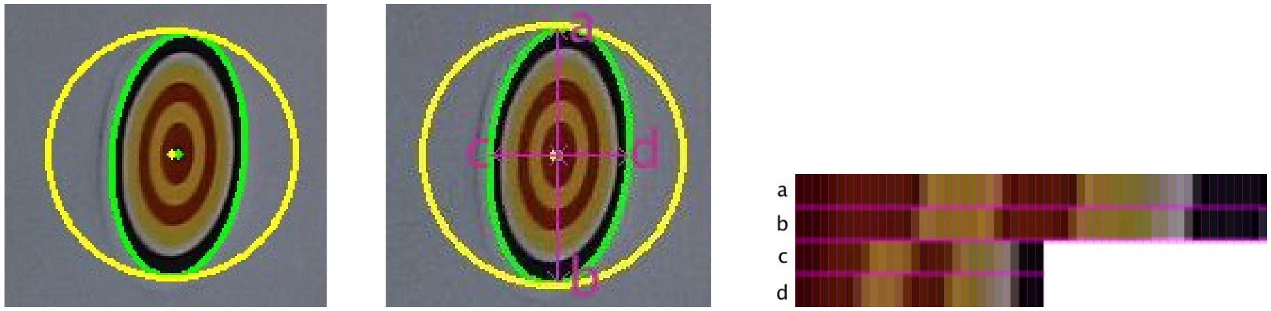
$$\frac{c_r\cdot 2}{e_a} > T_5 \tag{9}$$

3.3 Slicing

The next steps are using the eclipse parameters and color information of the original image. Not every pixel in the marker area has to be considered by exploiting the rotational symmetrical property of the marker. Instead, just four “slices” from the ellipse center to the border in four directions are extracted from the image (see Fig 3b and 3c).

A simple test to further verify the marker candidates is to check if the pixels of the extracted slices are sufficiently colorful or are different from a gray-scale value respectively:

$$(\sum_{r=1}^{4}\frac{1}{||row_r||}\sum_{c\in row_r}|c_R-c_G|+|c_R-c_B|+|c_G-c_B|) > T_6 \tag{10}$$



(a) Fitted circle (yellow), fitted ellipse (green) and their corresponding center points

(b) slices (magenta)

(c) Slice image: extracted slices displayed as rows

Fig. 3: Process of slicing a marker candidate which is represented by an ellipse

4 Marker recognition

The slicing process basically results in four one-dimensional signals with 3 channels each. To classify the marker slices, a hidden markov model for each marker is trained where each pixel of a sliced row provides the R,G and B values as features (emitted visible states) which are mapped during the training process to the color rings (hidden states). Each extracted slice can then be classified at runtime with the Viterbi algorithm using the trained models.

If the slices belong to a true marker candidate it can be assumed that the Viterbi algorithm will predict the same hidden markov model for all four slices. This kind of HMM agreement is also used to filter out false positives which made it through to this step.

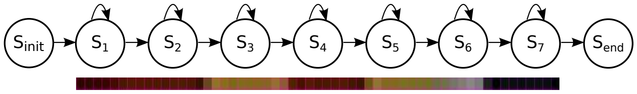


Fig. 4: Training of a hidden markov model from a marker slice

5 Experiments and results

The presented algorithm is implemented as an Android application with the image processing component written as native C++ code utilizing the OpenCV library. The image segmentation step is implemented in OpenCL. The application was tested on a Sony Xperia Z3 Compact Tablet which is able to process the image frames with a resolution of 1920 × 1080 at an average of 60ms per frame. The breakdown into the individual steps can be seen from Fig. 5.

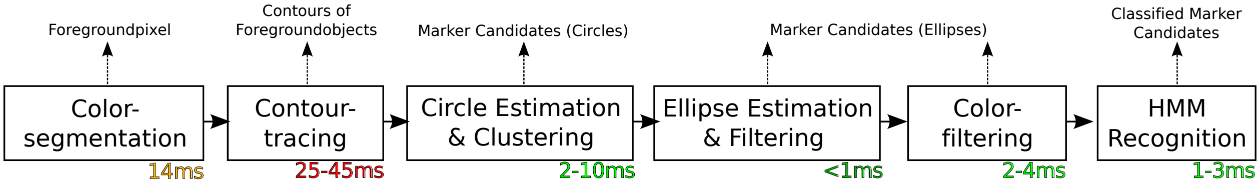

Foregroundpixel
Contours of Foregroundobjects
Marker Candidates (Circles)
Marker Candidates (Ellipses)
Classified Marker Candidates
Color-segmentation
14ms
Contour-tracing
25-45ms
Circle Estimation & Clustering
2-10ms
Ellipse Estimation & Filtering
<1ms
Color-filtering
2-4ms
HMM Recognition
1-3ms

Fig. 5: Approximate runtime of the individual steps

The filter parameters for a Full-HD resolution were chosen as shown in Table 1.

Parameter	Value
T_1	0.16
T_2	85
T_3	0.045
T_4	0.911
T_5	2.5
T_6	1.5
Cluster Width	64 px
Cluster Height	60 px

Table 1: Chosen parameters

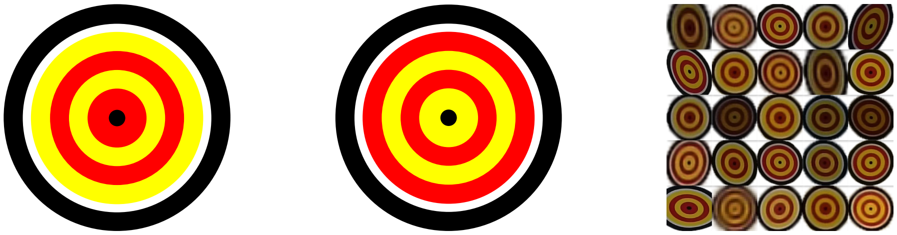
As shown in Fig. 6, two different markers were used. The training dataset was build from real images where 88 markers were extracted manually. For each marker sample four slices were extracted and used in the training process.

For the training process the Hidden Markov Toolkit (HTK) was used. An HMM prototyp with 9 states (including begin and end state) without allowing skips was trained with the provided `HInit` tool and further refined with `HRest` using a single mixture gaussian model. HResults reports an resubstitution error rate of 1.7% using 352 training samples. The rates per class can be seen in Table 2.

		Predicted Class		
		Marker 1	Marker 2	Total
Actual Class	Marker 1	167	5	Σ172
	Marker 2	1	179	Σ180
	Total	Σ168	Σ184	Σ352

Table 2: Confusion matrix of the classification results from HResults

The current implementation can detect and correctly classify a marker with a radius of 6 cm from a distance up to 9 m. A marker with a radius of 4.5 cm can be detected and classified up to a distance of 6 m. This corresponds closely to a radius of 14 pixels in the image frame for both cases.



(a) Marker 1

(b) Marker 2

(c) Training data subset

Fig. 6: Training data

6 Conclusions

The methods proposed in this paper provide a practical solution for a visual marker which is also suitable for long distances and recognition in realtime. The experiments have shown that the approach is working well regarding runtime and recognition. The proposed marker can be detected and recognised over a wide range of distances without triggering almost none false positive candidates.

Nevertheless, there remain interesting areas of further research. Currently the contour tracing step is taking the most of the overall computation time which is also depending on the number of foreground pixels returned by the segmentation step. An idea to speed up the process would be to exclude areas which were already classified as false positives in a previous frame.

As already mentioned the training data was created by hand from real images. The goal would be to achieve the same or better results with automatically created synthetic training data.

Currently only the raw RGB color values are used for the hidden markov model training process. To further improve the recognition result, methods used in speech recognition like an windowing approach and frequency features could be adapted for the marker recognition.

To further improve the use of the marker for augmented reality use cases, the augmented reality objects need to be placed based on the camera pose regarding to the marker. Therefore a mapping between the 2D ellipse and a 3D circle can be estimated using the approach by Hutter & Brewer (2009).

Acknowledgments

The work is partly founded by the German Federal Ministry for Economic Affairs and Energy.

References

Chang, F., Chen, C.-J. & Lu, C.-J. (2004), ‘A linear-time component-labeling algorithm using contour tracing technique’, *Computer Vision and Image Understanding* **93**(2), 206 – 220.

Hutter, M. & Brewer, N. (2009), ‘Matching 2-d ellipses to 3-d circles with application to vehicle pose identification’.

Naimark, L. & Foxlin, E. (2002), Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker., *in* ‘ISMAR’, IEEE Computer Society, pp. 27–36.

Pun, C.-M. & Wong, C.-F. (2008), Image retrieval using a novel color quantization approach, *in* ‘Signal Processing, 2008. ICSP 2008. 9th International Conference on’, pp. 773–776.

Suzuki, S. & Abe, K. (1985), ‘Topological structural analysis of digitized binary images by border following’, *Computer Vision, Graphics, and Image Processing* **30**(1), 32 – 46.

V. A. Knyaz, R. V. S. (1998), The development of new coded targets for automated point identification and non-on-contact surface measurements, *in* ‘3D Surface Measurements, International Archives of Photogrammetry and Remote Sensing, Vol. XXXII, part 5’, pp. 80–85.

Youngkwan Cho Jongweon, Youngkwan Cho, J. L. & Neumann, U. (1998), A multi-ring color fiducial system and a rule-based detection method for scalable fiducial-tracking augmented reality, *in* ‘Proceedings of International Workshop on Augmented Reality’.

Zhang, X., Fronz, S. & Navab, N. (2002), Visual marker detection and decoding in ar systems: a comparative study, *in* ‘Mixed and Augmented Reality, 2002. ISMAR 2002. Proceedings. International Symposium on’, pp. 97–106.