Barcode-Assisted Planar Object Tracking Method for Mobile Augmented Reality

Nohyoung Park, Wonwoo Lee, Woontack Woo.
GIST U-VR Lab.
500-712, Gwangju, S.Korea
{npark, wlee, wwoo}@gist.ac.kr

***Abstract*— In this paper, we propose a planar target tracking method that exploits a barcode containing information about the target. Our method combines both barcode detection and natural feature tracking methods to track a planar object efficiently on mobile devices. A planar target is detected by recognizing the barcode located near the target, and the target's keypoints are tracked in video sequences. We embed the information related to a planar object into the barcode, and the information is used to limit image regions to perform keypoint matching between consecutive frames. We show how to detect a barcode robustly and what information is embedded for efficient tracking. Our detection method runs at 30 fps on modern mobile devices, and it can be used for mobile augmented reality applications using planar targets.**

Keywords-QRCode; QRCode Detection; QRCode Tracking; Planar Object Tracking; Augmented Reality; 2D Barcode Tracking;

I. INTRODUCTION

QRCode [1] is a two-dimensional barcode. Recently, the use of QRCode has increased rapidly, owing to the widespread use of mobile devices. Typically, a QRCode has been used to get embedded text information from code readers. With mobile terminals, users can get text information such as URLs, e-mail addresses, phone numbers and so on. Instead of limiting the use of QRCodes exclusively to text information, there exist several applications that adapt QRCodes in diverse fields.

In order to adapt the use of QRCodes in the field of Augmented Reality (AR), vision-based object tracking methods are generally used. Currently, we can classify planar target tracking methods under two approaches: marker tracking and markerless tracking. ARToolkit [2] is widely used as a representative method for marker-based tracking. Such a method uses black-and-white rectangle patterns as markers or rectangular Simple Frame Marker [3]. Because of the formalized marker pattern, the detection process is simplified and requires only small computational power. The markerless tracking method has a more complex procedure than marker-based tracking. In order to perform markerless tracking, especially feature-based tracking [4], heavy computational power is needed. However, marker-based tracking is more robust with regard to changes in tracking conditions, such as light changes, occlusion, and complexity of image.

Nevertheless, both methods have a certain weak point: The tracking module should have an image pattern or pre-trained reference image information. In marker-based tracking, such as that offered by ARToolkit, we should know the binary pattern images that are located inside the fiducial marker. In the case of markerless tracking, we should prepare reference images or pre-trained information about an object. To cover the drawback of needing reference image information, our proposed method suggests a non-pre-trained object detection method by using standard shapes of a QRCode. Moreover, a QRCode has the advantage of encoding text information as its two-dimensional binary pattern. Thus, in this study, we propose a tracking method using auxiliary text information, which is embedded in a QRCode pattern. In order to define auxiliary text information for use as tracking data, we introduce a data representation of a QRCode that defines the additional tracking information. The overall procedure of our approach is shown in Fig. 1.

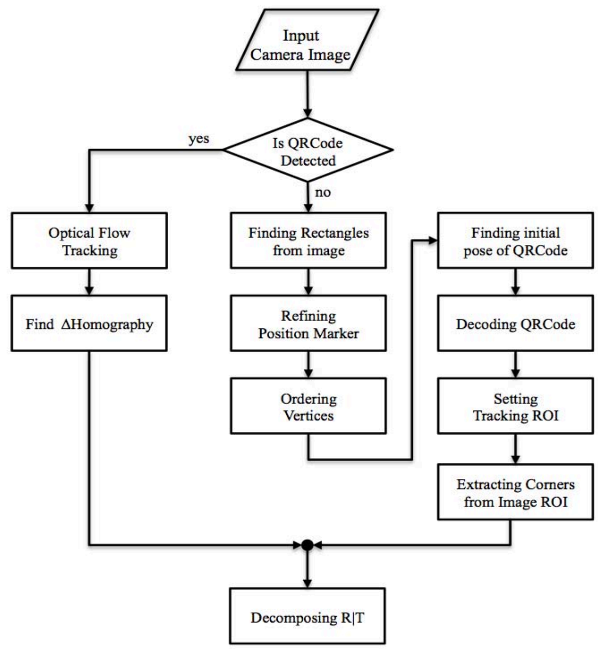

Figure 1. Flow chart of overal procedure.

II. PROPOSED METHOD

A. QRCode Detection

In order to detect a region and estimate the initial pose of a QRCode, the proposed approach uses vertices of position-markers. Originally, a QRCode has an alignment-marker that plays a role to scan inside the data pattern accurately. Unfortunately, it is difficult to detect a small-scale alignment marker from a camera image. Hence, we propose a method

978-0-7695-4420-5/11 $26.00 © 2011 IEEE
DOI 10.1109/ISUVR.2011.20

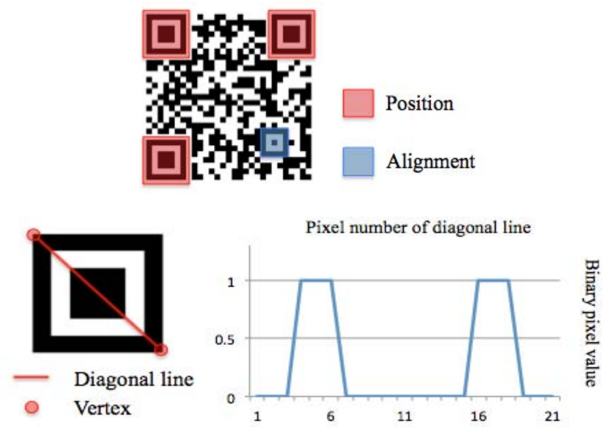

Figure 2. Structure of a QRCode, highlighting functional elements, and explanation of binary pixel values of position-markers.

that detects the QRCode using only position-markers from the acquired camera image.

The binary histogram of the QRCode can be used to find out position-markers. From a camera image, we can find several candidate regions which are produced by closed-contour detection. Those regions have various patterns that simplify binary pattern matching to distinguish whether each candidate region is a position-marker or not. In case of position-markers, a hierarchical black-and-white pattern represents its own identity. Comparing binary histogram patterns between candidate region and position-markers makes it possible to refine three position-markers from several candidate regions.

B. Initial Pose Estimation

To calculate an initial pose, it is necessary that position-markers and their vertices should be in a certain order, such that they are independent with directional changes. But without using alignment-markers, there exists a directional ambiguity between the three position-markers. To avoid that ambiguity, the proposed approach uses a cross-product between two vectors. Suppose that we have two different geometric vectors $\mathcal{u}$ and $\mathcal{v}$ that are started from one of position-markers and spread to other position-markers. From vectors $\mathcal{u}$ and $\mathcal{v}$, we can calculate the angle θ between $\mathcal{u}$ and $\mathcal{v}$. From a value of θ, a centered position-marker could be refined.

Furthermore, the proposed method can determine the order of the other two vectors by calculating a cross-product. After refining a centered position-marker, we could define geometric vectors $\mathcal{u}$ and $\mathcal{v}$ that start from the centroid of the centered position-maker and with their magnitudes ending at the other position-markers, as shown in Fig. 3. The calculation results of the cross-product $\mathcal{u}\times\mathcal{v}$ also compose a vector that has orthogonal direction from $\mathcal{u}$ and $\mathcal{v}$. The directional sign, which was produced by $\mathcal{u}\times\mathcal{v}$, represents whether vectors $\mathcal{u}$ and $\mathcal{v}$ are ordered correctly or not. So, comparing the direction of $\mathcal{u}\times\mathcal{v}$ makes it possible to order the other position-markers correctly. From that procedure, we could refine three position-markers from among several

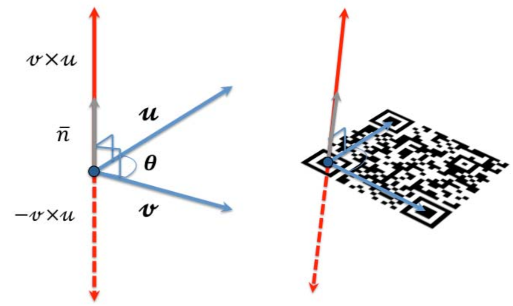

Figure 3. Result of direction produced by vector $\mathcal{u}$ and $\mathcal{v}$.

candidates of position-markers; moreover, we can determine the correct order of position-markers and their vertices.

In the process of pose estimation, the ordered three position markers and their vertices are used to calculate the initial pose of a QRCode. According to the result of position-marker ordering, the proposed method can determine twelve vertices from various directional changes. Given that the QRCode exists on a planar object, the initial pose of the QRCode can be calculated by using homography; the numerical formulation shown in Equation (1) represents homography ($\mathcal{H}_{init}$), which transforms the camera coordinates $(\alpha_n, \beta_n, \gamma_\mathrm{n})$ into image coordinates $(\mu_\mathrm{n}, \nu_\mathrm{n}, \omega_\mathrm{n})$. As already described above, both image coordinates and camera coordinates of a QRCode could be detected or be known, so an unknown matrix $\mathcal{H}_{init}$ can easily be calculated by solving an eigenvalue problem.

$$\begin{pmatrix} \mu_1 & \mu_2 & \cdots & \mu_{12} \\ \nu_1 & \nu_2 & \cdots & \nu_{12} \\ \omega_1 & \omega_2 & \cdots & \omega_{12} \end{pmatrix} = \mathcal{H}_{init} \begin{pmatrix} \alpha_1 & \alpha_2 & \cdots & \alpha_{12} \\ \beta_1 & \beta_2 & \cdots & \beta_{12} \\ \gamma_1 & \gamma_2 & \cdots & \gamma_{12} \end{pmatrix} \quad (1)$$

The result of a QRCode detection and initial pose estimation are shown in Fig. 7 at the end of this paper.

C. Tracking Metadata for f2f tracking

Primarily, a QRCode are widely used for obtaining embedded text information by a specific decoder on mobile devices. Instead of being limited to obtaining only text information from a QRCode, we propose to extend the use of a QRCode to Augmented Reality as well. The study "QRCode Data Representation for Mobile Augmented Reality" [5] applied a QRCode to the field of Augmented Reality. The data representation describes a data format (e.g., 'Code Metadata,' 'Content Metadata' and 'Tracking Metadata') that will be embedded in a QRCode for Augmented Reality.

From the data representation of a QRCode, the proposed method applies Tracking Metadata, which contains information about tracking a QRCode. This tracking metadata can include the following fields: tracking ROI (Region of Interest), tracking method, multiple tracking flags, and ROI types. Thus, once a QRCode has been detected and decoded, the tracking metadata can be used for robust and fast tracking performance. For example, the information about tracking ROI can specify a planar tracking area. As such, the tracking points can only be on the inside of

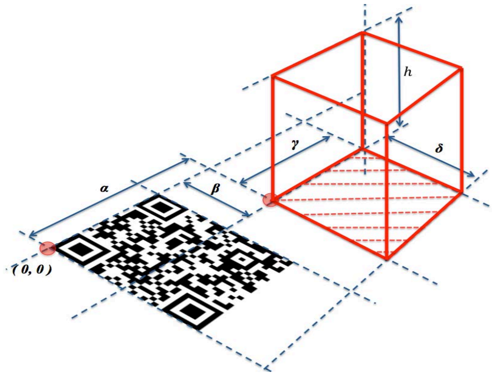

Figure 4. Explanation of Tracking Region of Interest.

a tracking region, and are only calculated for the specified region to save computational power for tracking; moreover, tracking points in a bounded area help to reduce re-projection errors, because tracking points are bounded in a specific plane. It is possible to identify an image region to perform tracking from the metadata. Tracking Metadata, and a data format of ROI information can be denoted as float point $\alpha, \beta, \gamma, \delta,$ and h. The values of α and β represent a two-dimensional coordinate of starting points, and the values of γ and δ represent the width and height of the tracking boundary. In order to expand our method into 3D-object tracking for a future work, we retain data point h, which represents the height of the tracking object. Besides, detection result $\mathcal{H}_{init}$ enables calculation of the relative image coordinates of those values. Fig. 4 shows the specific tracking region, which was composed by $\alpha, \beta, \gamma, \delta,$ and h.

D. Frame-to-Frame Tracking by using Tracking Metadata

In order to enable robust QRCode tracking from various changes of camera direction, we propose a barcode-assisted tracking method that uses auxiliary tracking information from Tracking Metadata, which was already introduced in the previous section. In addition, we will introduce details of a tracking method that traces corner points using Optical Flow [6], and a routine for updating the current camera pose.

The proposed frame-to-frame tracking method keeps up with tracing corners to enable robust tracking. First, we extract corners from the region of interest defined in the metadata. The proposed method uses Optical Flow tracking to trace the position of corner points. With Optical Flow between the image frame of $\mathcal{F}_n$ and $\mathcal{F}_{n+1}$, we could find slight distance changes of corner points, and the difference between $\mathcal{F}_n$ and $\mathcal{F}_{n+1}$ be considered as a transformation of homography between $\mathcal{F}_n$ and $\mathcal{F}_{n+1}$; we denote this as $\Delta\mathcal{H}$. Finally, to calculate current homography ($\mathcal{H}_n$) in the tracking process, we use multiplication of matrices. Equation (2) represents a method to find out $\mathcal{H}_n$ in frame-to-frame procedure.

$$\mathcal{H}_n = \mathcal{H}_{init}\, \Delta\mathcal{H}_0^1\, \Delta\mathcal{H}_1^2\, \Delta\mathcal{H}_2^3 \cdots \Delta\mathcal{H}_{n-2}^{n-1} \qquad (2)$$

The results of the barcode-assisted tracking are shown in Fig. 8 at the end of this paper.

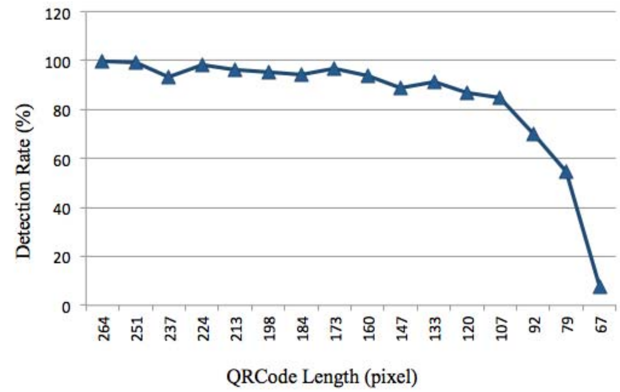

Figure 5. Detection rate of a QRCode

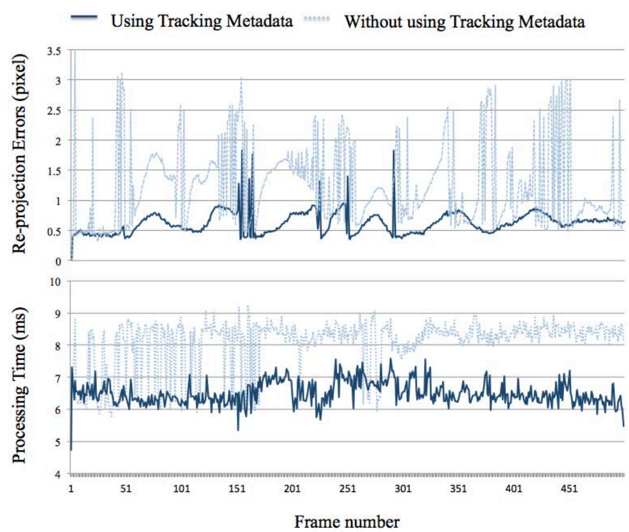

Figure 6. Comparison of re-projection errors and processing time in frame-to-frame tracking

III. EXPERIMENTAL RESULT

To verify the effectiveness of our method, we performed several experiments and analyzed the experimental results. The experiments were performed with a 2.93GHz Intel Core i7 CPU. In order to adapt our approach to the mobile environment, we implemented our detection module on an Apple iPhone 4 with 1GHz CPU. The image resolution used for experiment was 640×480.

To evaluate our detection method, we varied the length of a QRCode from an input image determined as a parameter of our experiment. In addition, to prove the effectiveness of our method for directional changes, we used images which rotate input images by 1° up to 360°. Fig. 5 shows the results of the detection rate. From these results, we can observe that the proposed method provides over 90% successful detection when we had a greater than 100-pixel length for a QRCode.

We measured the time regulated for a QRCode detection and f2f tracking, and the result are shown in Table 1.

TABLE I. DETECTION AND TRACKING PERFORMANCE

	Operating Environment (CPU / OS)	*Processing Time (milliseconds)*
Detection	Intel Core i7 / Mac	0.98
	A4 1GHz / ios	32.345
Tracking	Intel Core i7 / Mac	3.26
	A4 1GHz / ios	87.46

In order to prove the effectiveness of our approach, we compared mean values of re-projection distances and processing time between two methods; one method adopted Tracking Metadata and the other did not. Fig. 6 shows the result of the comparison. Our method has two less re-projection errors in most image frame sections, and from the second graph we can recognize that overall processing time was reduced by about two milliseconds.

IV. CONCLUSION AND FUTURE WORK

In this study, we proposed a barcode-assisted planar target tracking method for real-time mobile Augmented Reality application. Our method can efficiently tracks a planar target by exploiting the information embedded in a barcode, and we confirmed that the proposed detection was ported fast enough to a mobile device with 30 fps performance. We expect that our method can be useful for mobile Augmented Reality application, such as Context-aware AR authoring tool[7] or AR tour guide system[8].

ACKNOWLEDGMENT

This research is supported by Ministry of culture, Sports and Tourism (MCST) and Korea Creative Content Agency (KOCCA), under the Culture Technology(CT) Research & Development Program 2011.

REFERENCES

[1] BS ISO/IEC 18004:2006, Information technology. “Automatic identification and data capture techniques. Bar code symbology, QR code”.

[2] Hirokazu Kato, Mark Billinghurst, "Marker Tracking and HMD Calibration for a Video-Based Augmented Reality Conferencing System," iwar, pp.85, 2nd IEEE and ACM International Workshop on Augmented Reality, 1999.

[3] Hyejin Kim, Woontack Woo, “Simple Frame Marker for Image and Character Recognition”, *in Proc. of International Symposium on Ubiquitous VR 2008, pp. 43-46*, 2008.

[4] V. Lepetit and P. Fua, “Monocular Model-Based 3D Tracking of Rigid Objects: A Survey”, Foundations and Trends in Computer Graphics and Vision, 2005.

[5] H. Yoon, N. Park, W. Lee, Y. Jang, and W. Woo. QR Code Data Representation for Mobile Augmented Reality, The International AR Standards Meeting 2011, February 17-19, Barcelona, Spain.

[6] J. L. Barron, D. J. Fleet and S. S. Beauchemin “Performance of optical flow techniques”, International Journal of Computer Vision, 1994.

[7] Jonghee Park, Changgu Kang, Sejin Oh, Hyeongmook Lee, and Woontack Woo, Context-aware Augmented Reality Authoring Tool in Digital Ecosystem, ISUVR 2010, pp. 16-19, 2010.

[8] Choonsung Shin, Hyejin Kim, Changgu Kang, Youngkyoon Jang, Ahyoung Choi, Woontack Woo, Unified Context-aware Augmented Application Framework for Supporting User-Driven Mobile Tour Guide,*ISUVR2010, pp. 52-55*, 2010.

Figure 7. Result of QRCode detection performed on mobile device.

Figure 8. Result of QRCode tracking method.