Random Dot Markers

Hideaki Uchiyama* Hideo Saito†

Keio University

Abstract

This paper presents a novel approach for detecting and tracking markers with randomly scattered dots for augmented reality applications. Compared with traditional markers with square pattern, our random dot markers have several significant advantages for flexible marker design, robustness against occlusion and user interaction. The retrieval and tracking of these markers are based on geometric feature based keypoint matching and tracking. We experimentally demonstrate that the discriminative ability of forty random dots per marker is applicable for retrieving up to one thousand markers.

Index Terms: H.5.1 [Information Interfaces and Presentation]: Multimedia Information Systems—Artificial, augmented, and virtual realities; I.4.8 [Image Processing and Computer Vision]: Scene Analysis—Tracking

1 Introduction

In the studies on augmented reality, several techniques for detecting and tracking objects have been developed in the last decades. Especially, natural feature based approaches are becoming mainstream in recent years. With remarkable progresses that have been made, some of them can work in real time even on a mobile device with low computational power and memory [1, 2]. Even though these approaches are becoming more robust and requiring less computational cost, fiducial marker based approaches are still utilized in the development of prototype and actual systems [3, 4]. Compared to natural texture, the characteristics of these markers are visible explicitness and understandability for users. For example, the users can clearly know the positions on which virtual objects are overlaid by finding specific markers in front of the users. In this paper, we aim to come up with a novel type of a marker which can be applied to several meaningful uses and various applications.

The world's first marker system was developed by Rekimoto [5]. In this system, each marker had square black and white patterns as a barcode. The order of the patterns is decoded as a bit sequence for marker recognition. ARToolKit developed by Kato and Billingusrt is the most famous toolkit for developing augmented reality applications [6]. In this toolkit, users can put any shape of patterns inside a square black frame. Because this toolkit was open-sourced, it was ported into several development environments such as JAVA [7], Pocket PC [8] and mobile devices [9]. Uchiyama et al. developed different type of markers with hexagonal black and white patterns so that the number of different markers is efficiently increased [10]. They also provided the framework for the sensor fusion between a magnetic sensor and a camera equipped on a head-mounted display. ARTag utilizes a signal processing theory for generating patterns inside markers [11]. These markers have a bi-tonal pattern including checksums and forward error correction for robust marker recognition. The common shape of these markers is a square black frame for marker detection. However, this design has following two disadvantages. As is well known, the shape of the marker is only limited to a square with a black frame. Also, markers cannot be detected when a part of the marker is occluded. This becomes a limitation for user interaction because users cannot hold a marker by their fingers.

As another type of fiducial markers, a circular shape has also been considered because the center of the circle can accurately be detected from different viewpoints. For example, Naimark and Foxlin designed a 2D circular barcode with a black frame [12]. In this system, the combination of black and white sectors inside the frame corresponds to a marker ID for marker recognition. Wagner et al. put regular grids of dots on a paper map for incorporating natural features and fiducial markers [13]. These dots served as reference points in the tracking process, and have the better visibility and the robustness against partial occlusion.

In this paper, we present a keypoint based approach for detecting and tracking markers with random dots. The local arrangements of the dots are utilized as descriptors. By putting different distributions of random dots on different markers, multiple markers can be retrieved and tracked. Compared with other circular markers [12, 13], we utilize the random dots for retrieving a large number of different markers from a marker database.

This novel marker system was inspired from map image retrieval based on matching intersections [14]. Uchiyama et al. developed the framework for retrieving maps using the local distribution of intersections as a query. They demonstrated that the distribution could be utilized for distinguishing maps. The development of our random dot markers was motivated from this result and started from replacing intersections by random dots.

For a decade, several types of markers have been developed and their applications are almost becoming matured. However, our random dot markers still have potential possibilities so that the marker design and marker based user interaction will drastically be changed. The development of the novel markers will include wide-ranged important contributions, which are not be limited to the applications in augmented reality and human-computer interaction.

2 How to Make Random Dot Markers

A random dot marker is composed of randomly scattered dots as illustrated in Figure 1. Because the dots are directly utilized for marker retrieval and tracking, this marker does not need to have a square black frame, which is one significant difference compared with traditional square markers. In our method, we utilize the local arrangements of the dots for descriptors in keypoint matching and tracking using LLAH [15]. Because LLAH works well on the unique local arrangements of keypoints, we assume that the randomness of the distribution will lead to the uniqueness.

In order to generate random patterns on markers, we create random values on x and y coordinates within the marker size. The overlap of dots is not desirable and degrades the robustness of the recognition because they cannot separately be extracted from a captured image. To avoid this, we reject dots that are too close to other existing dots. Note that the number of dots in each marker does not need to be same.

In the preparation of the markers, the selection of the size of the dots is an issue. The size influences the valid range between a camera and markers for extracting the dots . If the size is large, the

*e-mail: uchiyama@hvrl.ics.keio.ac.jp
†e-mail: saito@hvrl.ics.keio.ac.jp

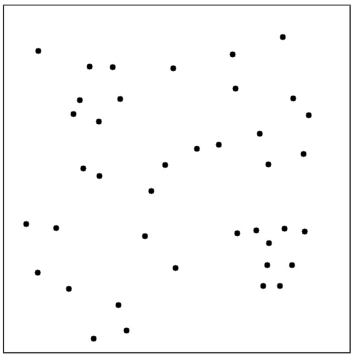


Figure 1: Example of random dot marker. A marker is composed of only random dots, and does not need a square black frame. An important element for making robust markers is the uniqueness of each local arrangement of neighbor dots. We assume that the uniqueness can be guaranteed by the randomness.

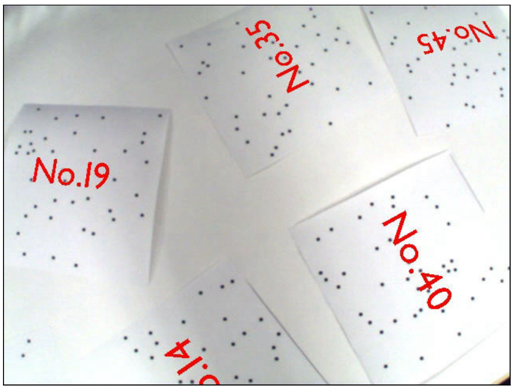


Figure 2: Multiple marker retrieval and tracking. The retrieved and tracked marker numbers are shown. In every frame, marker retrieval is performed to detect new markers. After the markers are retrieved, they are tracked from the next frame.

dots can be extracted from an image captured far from the marker. However, the visibility of the marker will be degraded in this case because the occupation of the dots in the marker is large. Because this is the trade-off between visibility and robustness, the size can be selected according to target applications. Also, the selection of the number of dots per marker correlates with the number of retrievable markers. This involves the same trade-off as mentioned above.

3 Multiple Marker Retrieval and Tracking

Our random dot marker system is based on multiple marker retrieval and tracking, which is the same approach as [16]. For continuous augmentation of pictures in this approach, frame by frame tracking is getting more important and works better compared with the case which uses only marker retrieval as tracking by detection.

First, users need to create the database of multiple markers. In the on-line marker recognition, the marker retrieval is performed in every captured image to detect new markers stored in the database. Once the markers are retrieved in a frame, they will be continuously tracked from the next frame. If a marker fails to be tracked, it will be detected again by the marker retrieval. By performing both retrieval and tracking, multiple markers can stably be recognized as illustrated in Figure 2.

3.1 Keypoint Extraction

For a captured image, keypoint extraction is first performed because the marker retrieval and tracking are based on keypoint matching. In order to utilize the center of each dot as a keypoint, we extract dot regions in the image.

In Figure 1, circular dots are black while their background is white. In this case, the dots can simply be extracted by thresholding brightness. Because the color of dots is not limited to black as red used in [14], other color extraction such as thresholding hue or saturation in HSV space is also applicable. After the binarization, connected regions are extracted and each center is computed as a keypoint.

3.2 Keypoint Matching by LLAH

For each extracted keypoint, its corresponding point is retrieved from the database. Because a dot itself does not have discriminative texture, we need to use geometric features computed from the dots instead. Geometric descriptors are developed in the studies of document image retrieval [15] and aerospace for tracking stars [17]. In our method, we use the local arrangements of the dots developed in LLAH [15].

The procedure for computing the descriptors is as follows. For each extracted keypoint, n neighbor points are selected. Then, m points are selected from n points. The number of all combinations ($_nC_m = \frac{n!}{m!(n-m)!}$) represents the number of descriptors per keypoint. In order to compute a descriptor from m points, 4 points are selected. The number of all combinations ($_mC_4$) corresponds to the number of dimensions per descriptor. From 4 points, a ratio of two bordering triangles is computed. For fast searching, $_mC_4$ dimensional descriptor is converted into a 1D index by a hash function.

When we create the database, each keypoint with its keypoint ID and marker ID is stored at computed indices in an inverted file.

3.3 Marker Retrieval

After keypoint matching is performed in a captured image, each keypoint has a corresponding keypoint in the database. Because each keypoint has its retrieved marker ID, all keypoints can be clustered by the marker ID. The clusters represent the candidates of detected markers in the image. For each cluster, homography is computed with RANSAC [18] as geometric verification. When the homographies in several clusters are computed such that the number of the correspondences is more than a threshold (we set 8), multiple markers are simultaneously retrieved.

3.4 Marker Tracking

After markers are retrieved, they are tracked until failing the tracking. When we use geometric descriptors for keypoint matching, the tracking has an important aspect for handling wide range of viewpoints as discussed in [14]. Thanks to the tracking, augmentation of markers is possible even when a camera is tilted as illustrated in Figure 3. In our work, instead of updating the descriptor database as in [14], we keep the descriptors of only previous frame because the updating procedure may produce an adverse affect such that the discriminative ability of the descriptors is degraded as the number of updated descriptors increases.

In $t-1$ th frame, the keypoints in the retrieved or tracked references are projected onto the captured image by using each homography to find more correspondences between keypoints in the image and keypoints projected from the references. For t th frame, keypoint matching by LLAH is performed with $t-1$ th frame. Because the keypoints in $t-1$ th frame have already had the correspondences with those in the reference, the keypoints in t th frame can also have the correspondences with those in the reference. For the correspondences of each marker, the homography between t th frame and the references in the database is computed as in the retrieval.

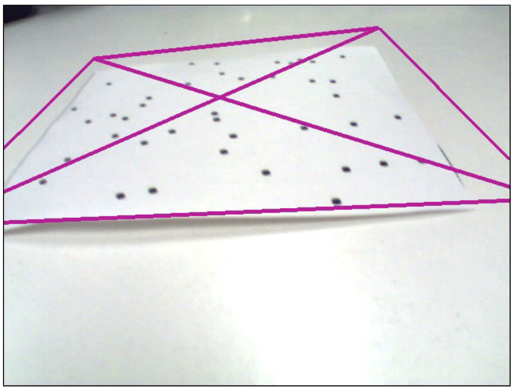


Figure 3: Tracking at tilted views. The initial camera pose should be almost top view for marker retrieval. After the retrieval is successfully done, a camera can be tilted thanks to frame by frame descriptor matching.

In every frame, the tracking is performed at first. In the retrieval, tracked markers are removed from the candidates of retrieved markers.

4 Advantages

Compared with traditional several markers, our random dot markers have good properties in the following aspects.

4.1 Marker Design

One of the characteristics of random dot markers is flexible marker design as illustrated in Figure 4. Because the random distribution of the dots can be embedded into any shape, the shape of the marker is not limited to the square shape. For example, we can make star like or heart shape in Figure 4, which could not be achieved by traditional square marker techniques.

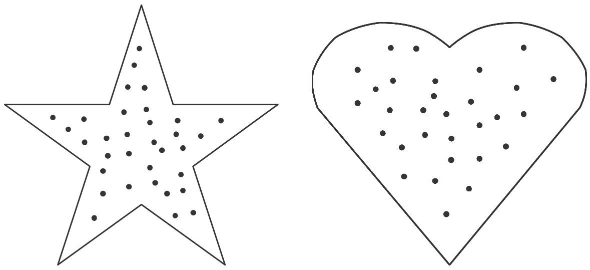
The shape of dots is not also limited to circular. The circular shape is desirable because the center of the circle may stably be extracted from various viewpoints as discussed in [12]. However, other shapes such as square or triangle can be replaced with the circle. In addition, the shape of all dots is not always same. In a marker, some of dots are circles and the others can be triangles. Depending on the design, the shapes of the markers and dots are flexibly selectable.

Thanks to the dots, the visibility of the marker is better than that of the markers with a black frame. In Figure 1, the dots occupy a small percent of the whole marker, and may not be obstacles on normal texture.

4.2 Robustness against Occlusion

Because our random dot markers are based on a keypoint based approach, the robustness against occlusion is better than that of the square markers. In Figure 5, overlaid squares with a cross represent the whole shape of each marker. Even in the presence of occlusion, the markers can be retrieved and tracked.

In ARTag [11], markers can be recognized even when a part of the black frame is occluded. However, they cannot be recognized when one side of the black frame is completely occluded because four corners should be detected. The retrieval and tracking of the random dot markers are not much influenced by such occlusion. In Figure 5, a marker at left lower part is occluded by three markers. Even though this huge occlusion occurs, the marker can be retrieved.



(a) (b)

Figure 4: Various type of marker design. Because only constraint of the design is to include random dots, the shape of the marker does not have any limitation. New types of markers such as star in (a) and heart in (b) can be possible.

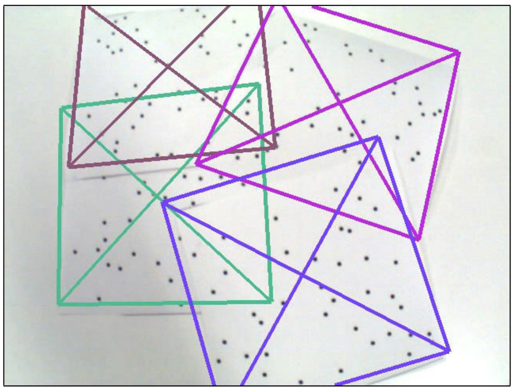


Figure 5: Robustness against occlusion. Each colored square with a cross represents the whole shape of each marker. Even though one side of the marker is completely occluded by other markers, the marker can still be retrieved and tracked due to a keypoint based approach.

4.3 User Interaction

As illustrated in Figure 6, a marker occluded by a hand can be retrieved and tracked as described in the previous section. Normally, a square marker cannot be recognized when users touch the marker because the occlusion is not acceptable. However, the random dot markers have a possibility toward user interaction with touching markers. For example, the region occluded by a hand on a marker can be computed as follows. The viewpoint of the marker in the image is rectified into that of its reference using the homography. Then, the hand region can be computed by subtracting the rectified marker and the reference. From this procedure, the occluded dots can be recognized and utilized as a query for the interaction. Also, dots themselves can be utilized such that the users can touch them to access the information embedded in them.

5 Evaluation

In order to measure the scalability of our random dot markers, we have tested marker retrieval for one thousand markers with forty dots. The markers are generated by following the rule described in Section 2. The parameters in LLAH [15] are selected as follows: $n = 7$, $m = 5$ and $k = 32$. The number of descriptors per keypoint is ${}_7C_5 = 21$ and the total number of keypoint numbers in the inverted file is $1,000 \times 40 \times 21 = 840,000$.

Each marker is captured from nearly top view because the inverted file for marker retrieval is generated from the dot distribu-

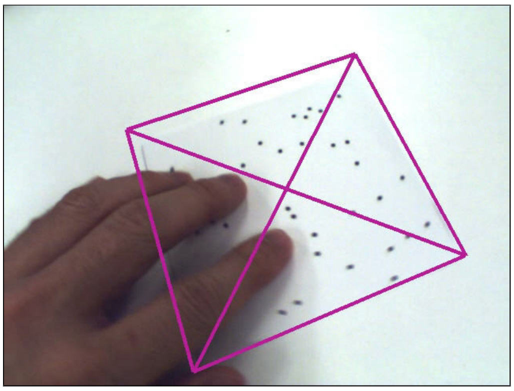


Figure 6: Toward user interaction. The occluded hand region can be computed by rectifying the image with a homography and subtracting the rectified marker and the reference. The size or location of occluded region can be utilized as a query in user interaction.

tions of the top views. For each marker, the retrieval was succeeded with less than a few milliseconds. From this result, the forty dots with random distribution may be discriminative for at least one thousand of markers. Because the computational cost is affected by only the number of extracted keypoints in the image, the retrieval of one marker will not need more costs as the number of the markers increases.

For multiple marker retrieval, the computational cost increased such as more than 100 millisecond in the presence of six markers. If there are many outliers in keypoint matching, lots of iterations occurs in RANSAC based homography computation. Because the retrieval needs much more costs than the tracking, the limitation of the computational costs in the retrieval is necessary for guaranteeing the frame rate. One solution is to limit the maximum number of iterations in RANSAC.

6 Conclusion and Future Works

This paper presented a novel technique for detecting and tracking markers with randomly scattered dots. These markers have significant advantages compared with traditional markers with square pattern. In the random dot marker system, the design of the markers was flexible such that the shape can be heart and star like. Thanks to the LLAH based keypoint matching, the markers were retrieved and tracked in real-time against the presence of occlusion. This is also helpful for user interaction because the occlusion always occurs during the interaction. As an approach for keypoint matching, the local geometrical relationship of the dots was utilized. When markers are retrieved, they are next tracked by frame by frame descriptor matching. From the experiment, forty random dots are utilized for retrieving up to one thousand markers.

As further works, we have following topics. In the current implementation, the distribution of the dots is set as random. However, there might be better distributions than random ones for markers. In other words, we need to seek the methodology for finding and generating the optimal distributions of dots depending on the number of markers and the trade-off between visibility, robustness and computational costs. Because this may also be related with the optimization of parameters in LLAH, the further research is necessary from the theoretical point of view.

The other interesting extension of our random dot markers includes developing non-rigid markers. In the traditional markers, they should be rigid and could not be deformed because they were detected by extracting straight lines of the black frame. However, normal papers can always be deformed and their shape is non-rigid. In the studies on deformable shape analysis, Pilet et al. developed a method for detecting non-rigid surface from keypoint correspondences [19]. This method can be applied to our random dot markers toward non-rigid markers, which will really enhance the functionality of fiducial markers.

Novel user interaction and 3D interfaces using random dots will also be sought. Because occlusion is not a problem, pinching or covering a part of the marker can be possible. These manipulations will be utilized as queries in various applications.

Acknowledgements

This work is supported in part by a Grant-in-Aid for the Global Center of Excellence for high-Level Global Cooperation for Leading Edge Platform on Access Spaces from the Ministry of Education, Culture, Sport, Science, and Technology and Grant-in-Aid for JSPS Fellows in Japan.

References

[1] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg. Pose tracking from natural features on mobile phones. In *Proc. ISMAR*, pages 125–134, 2008.

[2] G. Klein and D. Murray. Parallel tracking and mapping on a camera phone. In *Proc. ISMAR*, pages 83–86, 2009.

[3] K. Pentenrieder, C. Bade, F. Doil, and P. Meier. Augmented reality-based factory planning - an application tailored to industrial needs. In *Proc. ISMAR*, pages 1–9, 2007.

[4] S. Zhao, K. Nakamura, K. Ishii, and T. Igarashi. Magic cards: a paper tag interface for implicit robot control. In *Proc. CHI*, pages 173–182, 2009.

[5] J. Rekimoto. Matrix: a realtime object identification and registration method for augmented reality. In *Proc. APCHI*, pages 63–68, 1998.

[6] Hirokazu Kato and Mark Billinghurst. Marker tracking and HMD calibration for a video-based augmented reality conferencing system. In *Proc. IWAR*, pages 85–94, 1999.

[7] C. Geiger, C. Reimann, J. Sticklein, and V. Paelke. JARToolKit - a java binding for ARToolKit. In *Proc. ART*, 2002.

[8] D. Wagner and D. Schmalstieg. ARToolKit on the PocketPC platform. In *Proc. ART*, pages 14–15, 2003.

[9] D. Wagner and D. Schmalstieg. ARToolKitPlus for pose tracking on mobile devices. In *Proc. CVWW*, pages 139–146, 2007.

[10] S. Uchiyama, K. Takemoto, K. Satoh, H. Yamamoto, and H. Tamura. MR platform: a basic body on which mixed reality applications are built. In *Proc. ISMAR*, pages 246–253, 2002.

[11] M. Fiala. ARTag, a fiducial marker system using digital techniques. In *Proc. CVPR*, pages 590–596.

[12] L. Naimark and E. Foxlin. Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. In *Proc. ISMAR*, pages 27–36, 2002.

[13] D. Wagner, T. Langlotz, and D. Schmalstieg. Robust and unobtrusive marker tracking on mobile phones. In *Proc. ISMAR*, pages 121–124, 2008.

[14] H. Uchiyama, H. Saito, M. Servieres, and G. Moreau. AR GIS on a physical map based on map image retrieval using LLAH tracking. In *Proc. MVA*, pages 382–385, 2009.

[15] T. Nakai, K. Kise, and M. Iwamura. Use of affine invariants in locally likely arrangement hashing for camera-based document image retrieval. In *Proc. DAS*, pages 541–552, 2006.

[16] J. Pilet and H. Saito. Virtually augmenting hundreds of real pictures: An approach based on learning, retrieval, and tracking. In *Proc. IEEE VR*, pages 71–78, 2010.

[17] M. Kolomenkin, S. Pollak, I. Shimshoni, and M. Lindenbaum. Geometric voting algorithm for star trackers. *AES*, 44(2):441–456, 2008.

[18] Martin A. Fischler and Robert C. Bolles. Random sample consensus: a paradigm for model fitting with applications to image analysis and automated cartography. *C. of the ACM*, 24:381–395, 1981.

[19] J. Pilet, V. Lepetit, and P. Fua. Augmenting deformable objects in real-time. In *Proc. ISMAR*, pages 134–137, 2005.