IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received December 11, 2017, accepted January 20, 2018, date of publication February 8, 2018, date of current version May 9, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2801028

Robust Tracking Through the Design of High Quality Fiducial Markers: An Optimization Tool for ARToolKit

DAWAR KHAN[1,2,3], SEHAT ULLAH[3], DONG-MING YAN[1,2], IHSAN RABBI[3], PAUL RICHARD[4], THUONG HOANG[4], MARK BILLINGHURST[5], AND XIAOPENG ZHANG[1,2]

[1]National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences, Beijing 100190, China
[2]University of Chinese Academy of Sciences, Beijing 100049, China
[3]Department of Computer Science and IT, University of Malakand, Chakdara 18800, Pakistan
[4]Laboratoire Angevin de Recherche en Ingénierie des Systèmes, Université d'Angers, 49035 Angers, France
[5]School of ITMS, University of South Australia, Adelaide, SA 5095, Australia

Corresponding authors: Sehat Ullah (sehatullah@hotmail.com) and Xiaopeng Zhang (xiaopeng.zhang@ia.ac.cn)

This work was supported by the National Natural Science Foundation of China under Grant 61772523, Grant 61620106003, and Grant 61331018. The work of D. Khan was supported by the Chinese Government Scholarship.

ABSTRACT Fiducial markers are images or landmarks placed in real environment, typically used for pose estimation and camera tracking. Reliable fiducials are strongly desired for many augmented reality (AR) applications, but currently there is no systematic method to design highly reliable fiducials. In this paper, we present fiducial marker optimizer (FMO), a tool to optimize the design attributes of ARToolKit markers, including black to white (B:W) ratio, edge sharpness, and information complexity, and to reduce inter-marker confusion. For these operations, the FMO provides a user friendly interface at the front-end and specialized image processing algorithms at the back-end. We tested manually designed markers and FMO optimized markers in ARToolKit and found that the latter were more robust. The FMO will be used for designing highly reliable fiducials in easy to use fashion. It will improve the application's performance, where it is used.

INDEX TERMS Fiducial markers, ARToolKit, augmented reality, marker tracking, robust recognition.

I. INTRODUCTION

Augmented Reality (AR) is a type of mixed reality where computer generated information is superimposed over the real environment to enrich visual information and/or enhance visibility to user [1]–[3]. AR aims to be widely utilized in various applications. For example, AR-based teaching and learning [4], [5], robot path planning [6], and live conferences [7]. Similarly, AR is used in medical treatments, games, military training, maintenance, manufacturing, repair, and engineering applications [8].

Fiducial markers are real objects or images in a physical scene that are traced and recognized through a video camera and compared with existing markers in a library [9]. These markers are used for camera tracking and pose estimation [10]. A number of indoor computer vision applications are equipped with marker tracking [3]. It is a low-cost method for pose estimation as several AR systems generally use video cameras.

ARToolKit is open sourced, with simple configuration and easy documentation and mostly used in a number of AR applications. It is comparatively faster in execution than ARToolKitPlus and ARTag [11], [12]. Figure 1 shows certain ARToolKit markers with virtual overlying objects.

FIGURE 1. Examples of ARToolKit markers. Virtual teapots are overlaid (far left).

Issues remain regarding ARToolKit in terms of user guidelines to improve marker quality and tracking. A few patterns for fiducial markers are given with ARToolKit. However, clearly defined constraints for designing high quality markers are not in place.

Several alternative design decisions are available. However, a user may become confused on which design to use (for example, the quantitative width of the outer border of a marker and number of internal black objects in the marker body). Quantifying how objects are desired is not simple. Furthermore, a challenging decision for a user is ensuring that the new marker has a certain degree of dissimilarity with each existing marker in the library. To the best of our knowledge, such a marker designing tool that ensures marker distinctions in the library is not available.

In a previous research [13], we experimentally evaluated eleven factors and analyzed their effect on marker recognition. We tested these factors in different values and identified their optimal value(s) [13]. Subsequently, we proposed simple algorithms for distinct and sharp-edged markers [14] and a marker classification system to minimize inter-marker confusion [15]. Similarly, we proposed an algorithm for marker complexity computation and border control in a previous study [16]. However, the study has certain limitations, such as lack of detailed experiments. Furthermore, we used simple methods which may fail at a certain point. End users experienced difficulty because the codes lack proper implementation. Furthermore, an easy-to-use GUI tool is unavailable for end users. Lastly, a number of threshold values were used without experimental proof of their optimality.

In this paper, we further studied three factors related to marker design (i.e., sharpness of edges, black to white ratio, and complexity of interior information) and designed an easy-to-use GUI tool called fiducial marker optimizer (FMO). The FMO has an algorithm to ensure marker distinction and reduce inter-marker confusion. Furthermore, it also has a marker classifier module, which helps in reducing inter-marker confusion.

The FMO is developed for designing high quality markers with minimal inter-marker confusion and fast and accurate detection. It is a GUI based tool that enables users to upload each marker to the library and pass it through a step by step procedure. This process optimizes the attributes related to marker design, such as B:W ratio, sharpness of edges, and complexity of information, and reduces inter-marker confusion. The FMO could be used by ARToolKit users to design highly reliable and robust fiducial markers for accurate recognition. In this regard, we will provide online access to the FMO.

The remainder of the paper is structured as follows. Section II describes existing work on ARToolKit and fiducial markers, section. Section III presents the FMO in detail and section. Section IV describes the experimental study and results. In section V, the paper is concluded with limitations and future plans.

II. RELATED WORK

A number of studies on reliability of marker recognition, and their application and use are available. This section will review a few existing studies on reliability, usage, and issues in fiducial markers. We will also discuss ARToolKit and other systems for marker tracking.

A. RATING A FIDUCIAL MARKER'S QUALITY

A marker which is recognized in an easy and reliable way under various circumstances should be rated as a good marker [17]. Various applications have different requirements. Hence, the marker used in each application has its parameters for quality rating [17]. In either case, a systematic designing method to devise a robust fiducial marker is desirable [9]. High quality and reliable markers must meet the following criteria. The marker must be distinct from the context background, a unique entity in the marker library, passive (not covered with electronic materials), fast in detection, robust in low and high light, and detectable in noisy environments using certain robust algorithms for image processing [9]. Meeting these criteria is a challenging task in AR application for fast and accurate recognition with reasonable cost and minor changes to the desired applications [3].

The quality of markers can be rated in terms of high true detection rate and minimal false detection rate. In addition, increasing the library size without inter-marker confusion is also considerable. For example, in a previous work, Sun *et al.* [2] proposed QMarkers and compared its performance with ARToolKit markers. For comparisons they used marker recognition rate and its reliability in tracking. They have improved the detection rate as shown in their experimental study [2]. Similarly, diagonally connected component markers introduced in another study [18], are recognized without unwrapping for fast detection. Neto *et al.* [19] used multi-color markers to develop up to 65,000 distinct markers with considerable accuracy. They mentioned that low-quality cameras may cause miss-detection of the markers. Furthermore, they concluded that marker size should be increased with the increase in library size to ensure their distinction [19]. However they gave no proof experimentally.

Fiala [9] proposed eleven terms for evaluating a marker tracking system. These terms are given as follows. 1. The rate at which a marker is reported as present where it is, in fact, non-existent (false positive rate), 2. the rate of inter-marker confusion (ratio of wrong ID's reported to the total number of ID's reported), 3. the rate at which the system misses the detection of a marker which present in front of camera (the false negative rate), 4. the marker minimal size (the possible minimal size of the marker which is reasonable for detection) 5. the characteristics of vertex jitter (i.e. noise ratio in marker) 6. the library space (the total number of fiducial stored as templates in the library), 7. handling the change in lighting conditions, 8. handling occlusion, 9. the performance speed, 10. perspective support and 11. handling photometric calibration.

B. SOME MARKER TRACKING SYSTEMS

Several systems are available for marker tracking, such as CyberCode [20], Cantag [21], ARTag [12], Vuforia [22], AprilTag [23], AprilCal [24], ArUco [25],

ARToolKit [26], [27] and ARToolKitPlus [28]. ARToolKit was demonstrated publicly for the first time at SIGGRAPH in 1999 [27]. CyberCode [20] was available before ARToolKit, whereas ARTag and ARToolKitPlus were later introduced [29].

The main advantages with ARTag are low rate of inter-marker confusion, low rate of false negativity, and robustness in various lighting and partial occlusion [29]. The disadvantage with ARTag is computational complexity which making it incompatible with mobile phones and mini devices [17]. Compared to ARTag, ARToolKit is faster in terms of marker recognition [12]. ARToolKitPlus suffers from slow processing [11]. Meanwhile, the ARToolKit source code is freely available. Although other toolkits provide free access (to executables), they are not open source [11]. Olson presented an efficient digital coding system called AprilTag which can work robustly in occlusion, warping, and lens distortion. Although it is also open source like ARToolKit, AprilTag remains comparatively expensive in terms of computation [23].

ARToolKit is widely used in several AR and other interactive applications due to its source code availability [9]. It is compatible with various operating systems, such as PC Linux, SGI IRIX, Windows, Alao Mac, Android, and iOS [26]. Furthermore, ARToolKit is fast in processing. Hence, it is feasible for real-time interaction [11]. Although other toolkits also came into existence and may have their advantages, ARToolKit remains a wide-spread system. Recent research on ARToolKit markers [13], [30]–[34] and their use in different AR applications [35]–[39] highlights the ARToolKit's significance.

Various attempts have been made to design robust markers. In this regard, Vuforia [22] provided 512 frame markers which allowed users to scale the markers according to the desired application. Similarly, ARToolKit gives a number of sample patterns for marker designers [26]. In addition, online marker generating tools [40]–[42], are available. However, they cannot enhance tracking reliability. In a previous study [40], users were able to store the pattern of the printed markers using a camera. Similarly, [41] and [42] provided limited editing in marker images, web-based text areas, and submit buttons. Recently, Rabbi *et al.* [43] proposed a method to automatically create layered markers. The generated layered markers extended the marker tracking range to long range AR applications. However, their method is limited to layered markers only. To reduce inter-marker confusion, classifying markers is also helpful. For example, Badeche and Benmohammed classified Latin characters for ARToolKit markers to reduce inter-marker confusion [31]. However, they did not propose a generalized classification algorithm.

C. EXISTING PROBLEMS WITH MARKER DESIGNING

Although ARToolKit is easy to use and has proper documentation, the design and recognition of markers continue to hold a number of issues and challenges. Making decisions while designing a marker lacks a certain level of assistance. For example, the size of an outside border can be thick or thin or black objects in the marker body can be increased or decreased. These decisions alter the marker complexity (see Figure 2), which creates difficulty in designing high quality markers. A key challenge for a user in marker designing is to ensure that the new marker has a shape distinct from all existing markers in the library. Handling this issue is very challenging without any automatic module. To the best of our knowledge, a mechanism to ensure marker distinction in the library and reduce possible inter-marker confusion does not currently exist.

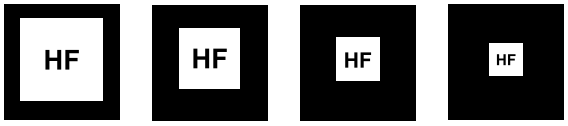


FIGURE 2. Example of markers with varying black border sizes.

Similarly, there are very limited guidelines [26] without any quantitative measurements for designing reliable markers. The ARToolKit suffers from inter-marker confusion, false positivity, and false negativity due to these issues. The reliability of the ARToolKit marker tracking varies from application to application and depends on various factors (such as the design of markers, library size, and light intensity) [13].

III. FIDUCIAL MARKER OPTIMIZER (FMO)

The FMO is a GUI-based tool developed in MATLAB to optimize the design attributes of markers, increase detection rate, and reduce inter-marker confusion. Its design attributes are as follows: black to white ratio, which is the ratio of the number of black pixels in the border to that of the interior white region (Sec. III-A); information complexity, which represents the complexity of black objects in the interior white region (Sec. III-B); and edge sharpness, which is the abruptness of the intensity of edge changes (Sec. III-C). The FMO also removes noise from the marker and checks for similarity among the markers of the library to predict inter-marker confusion. Figure 3 shows a model of the FMO, which represents the overall procedure of marker quality enhancement and its submodules.

First, when a marker is uploaded, the FMO checks the optimality of the marker's design attributes and performs the necessary editing operations, such as border optimization (B:W ratio), edge sharpness, and information complexity. The three operations may be performed in any order. Before a marker is added to the library, its similarity is checked against all existing markers to predict inter-marker confusion. Image correlation is used to provide a quantitative measure of the similarity between any two markers. This module assigns a degree of similarity to IMC (inter-marker confusion) variable ranging from [0, 1], where 0 means no similarity and 1 means exactly the same marker. If the IMC is greater than or equal to a given threshold, then conflict with a similar marker

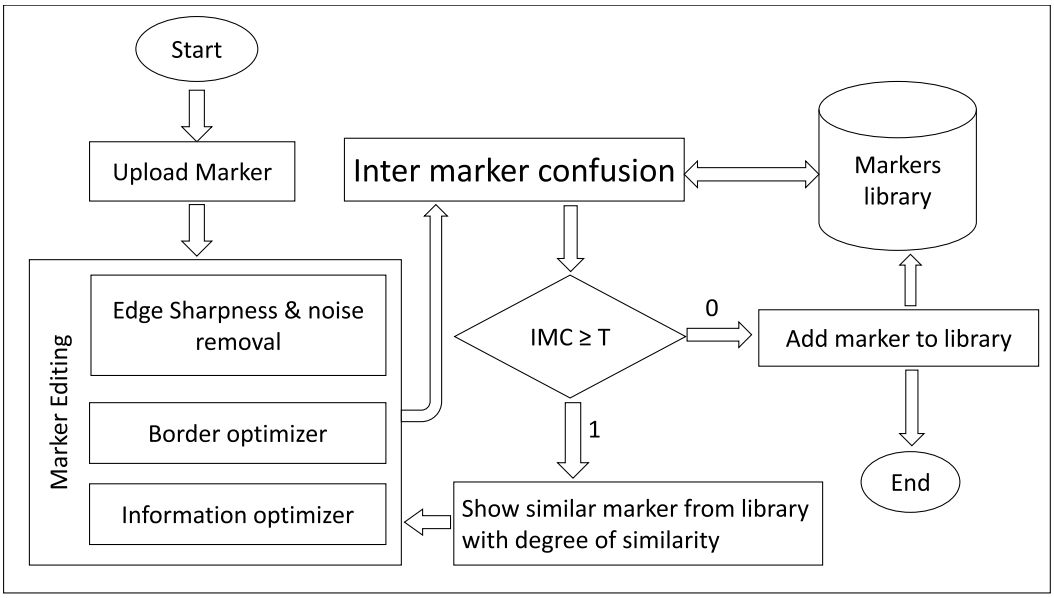


FIGURE 3. FMO model representing the overall procedure of marker quality enhancement and its submodules.

is prompted. These steps comprise the prediction of inter-marker confusion (Sec. III-D). Consequently, the marker shape must be changed to ensure that it is distinct from all existing markers. In the case of no confusion (IMC < T), the user is allowed to add the marker into the library, where it is ready to be used by ARToolKit. These steps are discussed in more detail in the following subsections.

A. BORDER OPTIMIZER

The border optimizer is used to optimize the black to white ratio. The B:W ratio is the number of black area pixels to that of the white area of the marker. For a marker having length of one side S and border b, the B:W ratio can be calculated as in equation 1.

$$\frac{B}{W} = \frac{4b(S-b)}{(S-2b)^2}, \tag{1}$$

By default, the ARToolKit assumes the border size to be 25% of its width (Figure 4) [27]. However, variation in border size affects tracking [13]. The optimal size for borders should range from 17% to 34% as identified in a previous study [13].

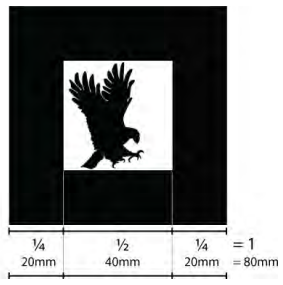


FIGURE 4. Structure of an ARToolKit marker, an example of the 25% default border [27].

The border is changed via a slider in the FMO. Translating the slider toward the left or right causes changes in the border length. When a border is set using the slider, the user presses the Confirm Border button of the FMO to apply the changes along the length of the marker (see Figure 5). To optimize border setting, Algorithm 1 is implemented at the back-end of the FMO which will be discussed below.

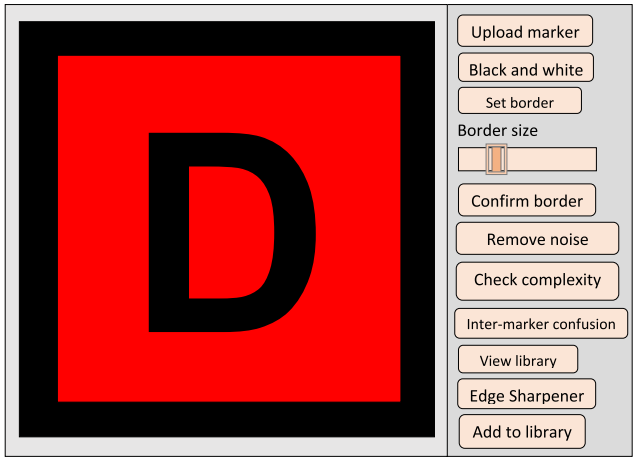


FIGURE 5. Front-end GUI of the FMO. The red color of the interior region of the marker indicates that the current width of the border is not optimal.

Initialization takes place in the first step, where the slider position is mapped into the border length. The marker is stored in binary version and the color for the interior region of the marker is set. A white color denotes an optimal border while red shows a non-optimal one. The border length is temporarily applied to the marker in the second step. Similarly, the marker preview is shown in the third step. Figure 5 presents a preview of the marker (with no optimal border). In this case, the user will change the border to

Algorithm 1 Optimizeborder(CM, slidervalue)

1: *Initialization* :
(i) $L_{border} \leftarrow \lfloor \frac{slidervalue}{100} * CM's\ length \rfloor$
(ii) $m_0 \leftarrow Convert2binary(CM)$
(iii) **if**(slidervalue ∈ [17 34])
Set font color: White
else
Set font color: Red
end if
2: $m_0 \leftarrow$ setborder(m_0, L_{border})
3: Preview($m_0, color$)
4: *if* (*Pressed confirm*)
(i) *if* (*color is red*)
Show message : "*Your border is not optimal*"
else
Show message : "*Your border is optimal*"
end if
(ii) $CM \leftarrow m_0$
end if
5: *End*

reach an optimal border, and apply the border permanently by pressing the FMO Confirm Border button. In the fourth step, a checker is used to validate whether the black to white ratio is optimal or not and show a corresponding message. Furthermore warning/guiding messages come in three types: (1) border width is smaller than threshold, (2) border width is optimal, and (3) border width is greater than threshold.

B. INFORMATION COMPLEXITY CONTROL

The interior objects in the marker body can vary in number or size (see Figure 6). With less information, creating distinct markers becomes difficult if the library size is large. Meanwhile, complex information may cause slow detection. To solve this problem, the FMO can check for information complexity. For this purpose, we developed and implemented a threshold-based algorithm (i.e., algorithm 2), which works as described below.

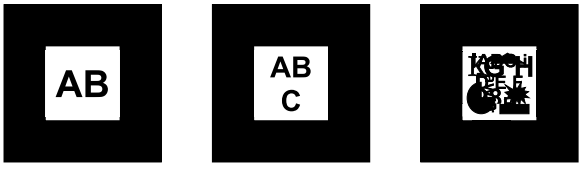


FIGURE 6. Example markers of information complexity. Information complexity is increasing from left to right.

First, the white region of the current marker (interior region) is stored in A variable. Then, the number of zeros and ones of the binary marker are counted which shows black and white pixels respectively. In the fourth step, marker complexity is computed as the ratio of pixels, i.e., number of zeros (black) to the total number of pixels in the interior region. In the fifth step, the number of intensity changes is counted because certain markers may have less black pixels but are considered complex because of the large number of small objects in the interior white region. In intensity changes,

Algorithm 2 ComplexityChecker(CM, T_0, T_1)

1: $A \leftarrow$ RemoveBorder(CM)
2: *black area* ← No-of-zeros(A)
3: *white area* ← No-of-ones(A)
4: *complexity* $\leftarrow \frac{black\ area}{total\ area}$
5: *intensitychanges* ← CountIntensitychanges(A)
6: **if** ((*complexity* $\geq T_0$) OR (*intensitychanges* $\geq T_1$)) **then**
Print "Marker complexity is greater than optimal"
else
Print "The marker complexity is optimal"
end if
7: *End*

we count the occasions when adjacent pixels change from 1 to 0 or 0 to 1. We divided the number of changes by the total number of pixels in the interior region of the marker (excluding the border). The process is repeated row-wise as well as column-wise. Lastly, the maximum value is selected. In the next step, these computed values are checked with the given thresholds. A warning message is displayed if the complexity of the marker is not optimal, which prompts the user that it may cause problems in detection. The threshold values are identified in Section IV.

C. EDGE SHARPNESS AND NOISE REMOVAL

Edge sharpness and noise removal are two of the important operations of the FMO. Noise refers to the small white spot(s) in the black area or small black object(s) in the white area of a marker. It may be caused by marker editing in the FMO or during a prior design of the marker. In this step, the current marker is first converted into a binary version. Then, we use morphological operators (as used in another study [44]) to remove small objects from the binary image. For this purpose, we used algorithm 3.

Algorithm 3 SmallObjectsRemoval(CM)

1: $m_0 \leftarrow$ im2bw(CM)
2: $m_0 \leftarrow$ bwareaopen($m_0, 30^0, 4c$)
3: $m_0 \leftarrow$ bwareaopen($\sim m_0, 30^0, 4c$)
4: $m_0 \leftarrow$ bwareaopen($m_0, 60^0, 8c$)
5: $CM \leftarrow$ bwareaopen($\sim m_0, 60^0, 8c$)
6: *End*

In the first step, the gray pixels are removed by converting the current marker CM into binary (black and white). It also make the edges sharper as it makes abrupt change in intensities. In 2^{nd} step, all spots of with pixel count ≤ 30 in 4-connected are removed. Then, the marker is inverted pixel-wise (i.e., pixels are converted from 0 to 1 and from 1 to 0). Noise is extracted again in the same manner (regions with ≤ 30 pixels 4-connected). In the next step, regions with a pixel count ≤ 60 in 8-connected are also removed. are also removed. In the fifth step, the marker's pixels are inverted again to retain their original shape, whereas regions with pixels ≤ 60 in 8-connected are removed again. The second and fifth steps remove white objects in black areas of the marker, whereas the third and fourth steps remove black objects from

the white region. Figure 7 illustrates the procedure step-by-step. To better understand the steps, we passed a noisy marker to the algorithm.

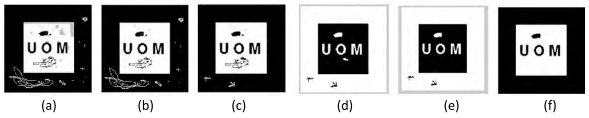


FIGURE 7. Steps in the edge sharpness and noise removal algorithm. (a) Input marker. (b) Edge sharpness. (c) Noise removal from border (4-connected pixels). (d) Noise Removal from interior region (4-connected pixels). (e) Noise removal from interior region (8-connected pixels). (f) Noise removal from border (8-connected pixels). Note: in (d) and (e), the light colored borders have no meaning. They are used only for visibility in the paper.

D. CHECKING INTER-MARKER CONFUSION

Ensuring that the current marker is unique in shape and distinct from other markers in the library is a very challenging task. The FMO has an algorithm that computes the degree of similarity between the current and individual existing markers. When the similarity (with each marker) is smaller than the given threshold (T = 0.46), a message with no confusion is printed. Otherwise, if similarity is above or equal to the given threshold, then the most similar marker (MSM) along with the computed value for similarity is printed (see algorithm 4).

Algorithm 4 InterMarkerConfusion(CM, Lib(n), T)

1: *Initialize variables* :
   (i) $key1 \leftarrow$ Convert2binary(CM)
   (ii) $Corr_{max} \leftarrow 0$
   (iii) $MSM \leftarrow$ NULL
2: **for** $i \leftarrow 1$ *to* $|n|$ **do**
   $CR \leftarrow$ CheckConfusion($key1$, $Lib(i)$)
   // Call to algorithm 5
   if (CR $>$ $Corr_{max}$) **then**
   $Corr_{max} \leftarrow CR$
   end if
   if(CR $\geq$ T) **then**
   $MSM \leftarrow$ Lib(i)
   end if
   end for loop
3: **if**($Corr_{max} \geq$ T) **then**
   (i) Showmarker(MSM)
   (ii) Print "Degree of similarity =". $Corr_{max}$
   end if
4: *End*

Current marker *CM*, marker library *Lib(n)*, and the recommended similarity threshold *T* are given as input to the algorithm. The current marker is checked against each existing marker in the library, then the *MSM* is displayed (if it exists). In the first step, the variables *key*, *MSM*, and $Corr_{max}$ (maximum correlation) are initialized with the binary version of the marker, *NULL* value, and 0, respectively. In the second step, the *key marker* (i.e. current marker in binary version) is checked and compared against each existing marker (in the library) via *CheckConfusion()* algorithm (i.e. algorithm 5, which will be described in detail). Here *Lib(i)* means the i^{th} marker in library, whereas *CR* shows the similarity (correlation) between two markers. Here, a marker having a high similarity with the current marker is found and its degree of similarity, i.e., maximum correlation, is stored in $Corr_{max}$. In the third step, if the condition ($Corr_{max} \geq T$) is true, then the most similar marker with corresponding value of $Corr_{max}$ are printed. In other words, the algorithm will prompt that the current marker (key) has the most similar marker in the existing library with a value of degree of similarity equal to $Corr_{max}$. For example, adding the two markers shown in Figure 8 causes inter-marker confusion in the library. Therefore, the FMO will show both markers with their degree of similarity (0.64533) to prompt the user and avoid inter-marker confusion.

Algorithm 5 CheckConfusion(key, marker)

1: Rotating the marker:
   (i) $A \leftarrow$ marker
   (ii) $B \leftarrow$ Rotate($marker$, 90^0) // Rotated by 90^o
   (iii) $C \leftarrow$ Rotate($marker$, 180^0) // Rotated by 180^0
   (iv) $D \leftarrow$ Rotate($marker$, 270^0) // Rotated by 270^0
2: Similarity computation:
   (i) $cr1 \leftarrow$ corr2(key, A)
   (ii) $cr2 \leftarrow$ corr2(key, B)
   (iii) $cr3 \leftarrow$ corr2(key, C)
   (iv) $cr4 \leftarrow$ corr2(key, D)
   (v) $cr \leftarrow$ max($cr1$, $cr2$, $cr3$, $cr4$)
3: *Return cr*
4: *Theend*

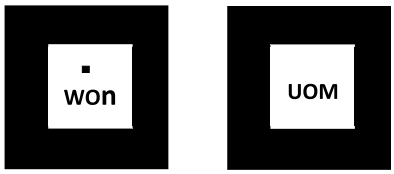


FIGURE 8. Example of two similar markers having a 0.64533 degree of similarity with each other (as calculated in FMO). If one of these markers already exists in the library, the FMO will not allow user to add the second one.

The algorithm for computing the confusion between markers is given as algorithm 5, which is described below.

First, the *marker* (from library) is stored in variable A. Then the marker is rotated with 90^0, 180^0 and 270^0 (see Figure 9) and stored in variables *B, C* and *D* respectively.

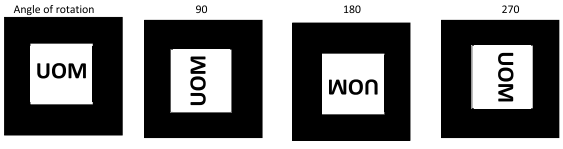


FIGURE 9. Marker from the library is rotated in different angles to compute its similarity with the current marker.

Then, we use image correlation to explore the similarity of the *key* (i.e., binary version of the current marker) with the abovementioned rotations of a marker. Finally, the algorithm displays the computed maximum similarity

E. ARToolKit MARKERS CLASSIFIER

Marker classification also helps to minimize inter-marker confusion [31]. The ARToolKit marker classifier is an additional capability in the FMO to ensure marker distinction and minimize the rate of confusion which occurs in the library.

In FMO, we implemented an algorithm which can be used to group the markers of the library into a specific number of distinct groups (i.e., classes). Each group will contain markers that have similar shapes, whereas markers from two different groups will be sufficiently distinct from each other. Each marker in the library is individually compared with all markers. The quantitative value for the similarity between two markers is stored. These quantitative values are stored in an $N \times N$ table (see Table 4). Then the markers in the table are grouped into k distinct groups by calling the *k-means* algorithm. We recommend picking markers from different groups to minimize the rate of expected confusion. A marker from a small-sized group is comparatively distinct in the library.

1) CLASSIFICATION ALGORITHM

The classification algorithm can be used to classify the markers of the ARToolKit into a given number of groups (see algorithm 6).

Algorithm 6 Classify (Library[n], k)

```
1: N ← CountMarkers(Library[n])
2: for i ← 1 to N1 do
       (i) key.marker ← Library[a]
       (ii) for j ← i to N do
            (*) marker2 ← Library[j]
            (*) cr ← CheckConfusion(key, marker2)
            (*) Results[i][j] ← cr
            (*) Results[j][i] ← cr
          end inner for loop
   end outer for loop
3: List ← k-means(Results, k)
4: return List
5: end
```

This algorithm takes all the markers in the library (*Library*[*n*]) and the k value (i.e. number of classes) as input. IIn the initial step, the algorithm counts the total number of markers in the ARToolKit library. In the second step, all markers along with their numerical values for similarity between any two markers are added to the $N \times N$ Table i.e. *Results* (see example in Table 4). Here similarity is computed by calling algorithm 5), i.e. *CheckConfusion()* algorithm. In the third step, the markers are grouped into a given number (k) of distinct groups. Lastly, the grouped markers are sent as output in step 4.

Table 4 depicts the results for the classification of seven markers into four groups. The marker names are listed in the top row as well as in the left column, whereas the number of groups is placed in the bottom row. The remaining cells of the table are filled with the numerical values of similarities between each pair of markers. For every *(i, j)* and *(j, i)* cell, the numeric values of the similarity between the i^{th} and j^{th} marker are added.

IV. EXPERIMENTAL STUDY

We evaluated several aspects of the FMO in seven sets of experiments. During the experiments, the frame rate, light intensity, and other factors which affect tracking were kept constant in their optimal ranges as identified in our previous study [13]. The marker has to be moved continuously in front of the camera during the experiments (see Figure 10). In a left to right motion, one complete cycle (i.e., left to right and back to left from right) was $1m$ in length with 45 cycles per min (the average marker speed) [13]. Meanwhile, one complete cycle in to and fro motion was $0.2\ m$. For each 5-minute experiment, the left to right motion lasted 4 min, while the to and fro motion lasted 1 min. The same time ratio (4:1) for both types of movement was used in all experiments.

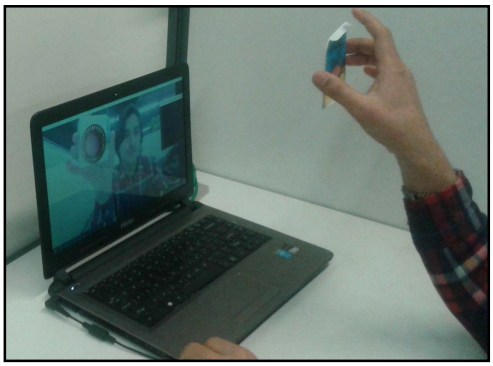


FIGURE 10. Experimental setup. The marker is in a continuous movement in front of the camera and the results (i.e., number of identifications and CF values) are continuously being stored in a text file.

(1) First, we manually designed three markers and passed them through the FMO for quality enhancement and obtained the output. We individually tested the original and optimized markers in the ARToolKit.

Figure 11 shows the three markers, where the first row represents manually designed markers while the second row represents the FMO optimized markers. Marker 2 has been edited further to optimize its complexity of information. During the experiments we noted the marker's true detection for 5 min. The library size was 20 markers. Hence, the true identification of marker A is validated if the camera recognizes marker A when it is flashed in front of the camera. Similarly, if marker A is recognized as marker B or any other marker in the library, then it is considered false identification.

The confidence factor (CF) values were also recorded for each instance of marker detection whose average was calculated for each experiment. CF is a quantitative value in the range [0, 1], which represents the level of similarity between the detected marker and corresponding stored

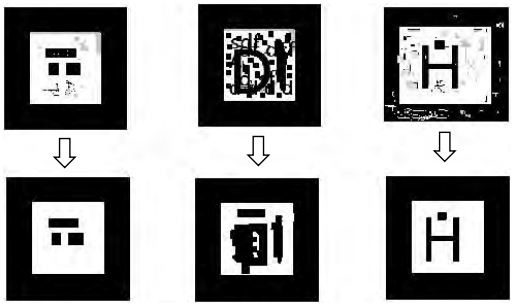


FIGURE 11. Three markers used for testing (i.e., experiment 1). From top to bottom: manually designed and FMO optimized markers. From left to right: markers 1, 2, and 3.

template. It highlights the endurance level of ARToolKit for the current detection i.e., the level of similarity between the marker and its template as determined by ARToolKit. A high value of CF denotes a high level of confidence, whereas a small value denotes less similarity. The results are compared, and a considerable enhancement was found in the average CF values as well as marker detection rate for markers that were optimized by the FMO (see Figure 12).

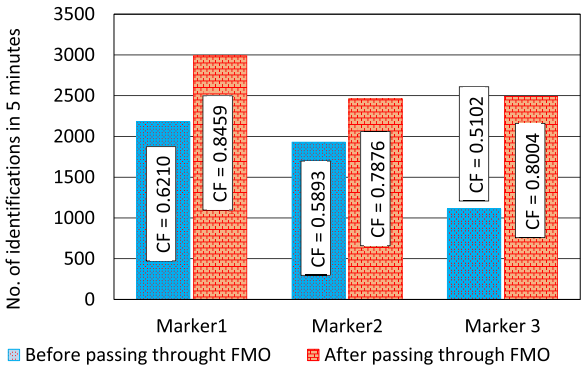


FIGURE 12. Results of experiments on the input and output markers of the FMO.

(2) The second experiment aimed to examine the meaningful relationship between the numerical value for computing marker similarity used in FMO and that of the ARToolKit. In short, we intend to show the semantical relationships of the FMO and ARToolKit marker matching methods. In this regard, we used 10 markers as a testing set (see Figure 13).

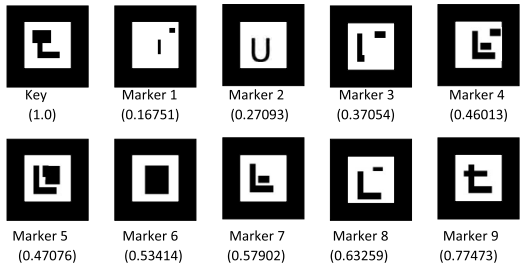


FIGURE 13. Markers used in experiment 2 with calculated degree of similarity with the key marker.

We designed one marker as key marker which is translated and rotated in front of the video camera for recognition via ARToolKit. The library contained five ARToolKit markers, namely, a key marker, another marker picked from the testing set, and three additional markers (to increase library size).

The key marker was rotated and translated in front of the camera for 10 min. We stored the number of instances that the marker is detected as the key (true detection) and the number of instances that it is detected as the testing marker (confusion with testing marker) markers and the total number of detection (see Figure 10). Similarly we replaced the testing marker with another marker from the testing set and repeated the experiment. In this manner the same experiment was re-executed for each of the testing marker.

In each experiment we computed the percent inter-marker confusion (i.e. confusion due to testing marker) with equation (2), where '$\sum I(test)$' shows the number of identifications as testing marker and '$\sum I(N)$' shows the total numbers of identifications.

$$\%\text{inter-marker confusion} = \frac{\sum I(test)}{\sum \mathrm{I}(N)} * 100, \qquad (2)$$

Similarly, the similarity of the key marker with each testing marker was also computed in the FMO. Marker similarity equal to 1 means the two markers (key and testing markers) have the same shape. It is to be noted that, the FMO crops the interior white region of a marker for calculation of its degree of similarity with the key marker. Figure 13 presents the markers used in this experiment along with the calculated degree of similarity with the key marker. Table 1 shows the results while Figure 14 shows the graph between the degree of similarity of the FMO and % inter-marker confusion in the ARToolKit; which highlight that marker similarity computed via FMO has a meaningful match with the inter-marker confusion in ARToolKit. Therefore, this module is feasible to control inter-marker confusion in ARToolKit.

TABLE 1. Results of experiments on the inter-marker confusion. Here *degree of similarity* means the similarity between key marker and testing marker calculated in FMO, whereas IMC means inter-marker confusion caused by testing marker (in ARToolKit).

Serial #	degree of similarity	% inter-marker confusion
1	0.16751	9.1%
2	0.27093	9.66%
3	0.37054	17.54%
4	0.46013	23.45%
5	0.47076	24.21%
6	0.53414	26.92%
7	0.57902	27.88%
8	0.63259	30.77%
9	0.77473	33.12%
10	1.00000	41.15%

Furthermore we observed a rapid increase in inter-marker confusion with the increase in the degree of similarity of up to 0.46. Hence, this value is recommended as a threshold for inter-marker confusion

(3) The objective of these experiments was also the same for experiment 2. The % inter-marker confusion in

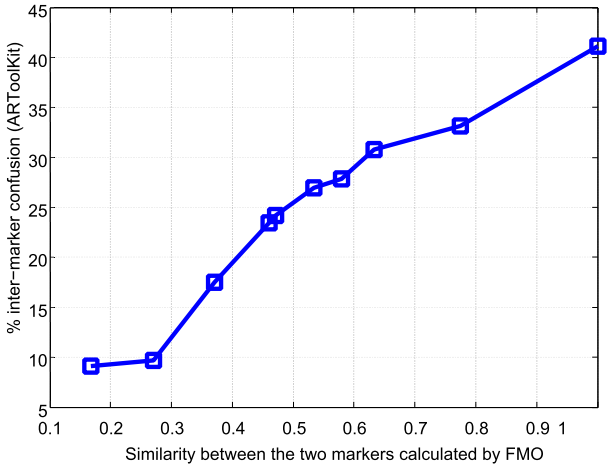


FIGURE 14. Graph between degree of similarity provided by FMO and % inter-marker confusion in ARToolKit.

ARToolKit was replaced by CF value. We used 40 markers in 20 pairs of markers. We measured the degree of similarity between the two markers in each pair in FMO. Similarly, to measure the CF value (representing the similarity between the two paired markers in this case, i.e., false positivity of one marker caused by the second one), we store the pattern of one marker in ARToolKit and brought the second one in front of the camera during testing. We repeated the same experiment for the 20 pairs. To get further valid results, we calculated the average of the positive CF values generated in a two-minute experiment for each pair (instead of a single CF value). Figure 15 plots the results, which shows the validity of the FMO prediction on inter-marker confusion for ARToolKit.

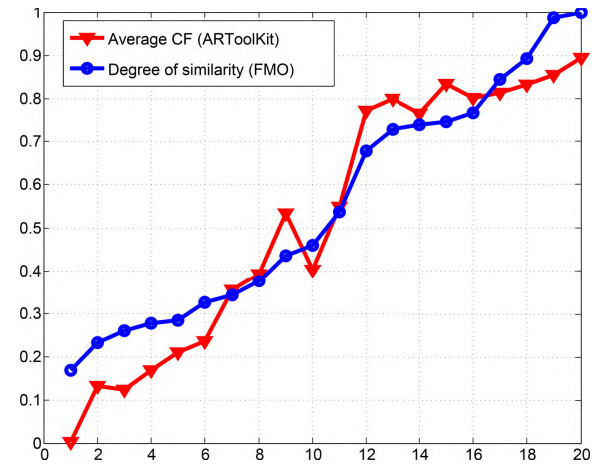


FIGURE 15. Graph between degree of similarity provided by FMO and the average CF values generated by ARToolKit.

(4) The fourth experiment intends to examine the common mistakes while manually designing fiducial markers. For this purpose, we asked five ARToolKit users to design a library of twenty distinct markers. The library made by each user was then passed through FMO and several shortcomings/mistakes were captured. Table 2 shows the results, which denotes that

TABLE 2. Results of experiments on capturing mistakes of the ARToolKit users

	user 1	user 2	user 3	user 4	user 5
markers with non optimal B:W	5	11	6	9	3
markers with non optimal complexity	0	1	0	0	1
edge sharpness or noise removal necessary	3	7	5	2	3
pairs with inter-marker confusion	1	5	3	6	2

ARToolKit users are making many errors in designing fiducial markers and that FMO can reduce these errors. The values in Table 2 show the number of markers. The optimal values are the recommended thresholds for the desired attributes.

(5) We asked the same users in experiment 4, to design these markers using FMO and perform experiments similar to experiment 1. In this manner, one marker was selected from the twenty markers as testing marker (in front of camera) and the library size was 20 markers. The experiments were repeated for each user in both cases (manually designed, as well as FMO optimized markers). Table 3 lists the results of each user. The average results have been graphically shown in Figure 17, which show a considerable improvement in all, three parameters were used (true identification, false identification, and CF values).

TABLE 3. Results of experiment 5: Comparison of manually designed and FMO optimized markers.

(a) Manually Designed markers			
	True identifications	False identifications	Average CF
user 1	2834	467	0.7863
user 2	2304	1489	0.7345
user 3	2548	1321	0.7913
user 4	2107	1730	0.7129
user 5	2944	863	0.8126
average	2547.4	1174	0.76752
(b) FMO optimized markers			
user 1	3090	411	0.7879
user 2	2611	743	0.7789
user 3	2819	438	0.8129
user 4	2571	535	0.8012
user 5	3112	344	0.8322
average	2840.6	494.2	0.80262

(6) These experiments aim to identify the recommended threshold values for complexity and intensity change (*T1* and *T2*) used in algorithm 2. In this regard, we used nine markers which have various complexity levels and tested each of them in front of the camera with a library size of 20 markers (similar to experiment 1). During the experiment we counted CF values, true identifications, and false identifications for 5 min. Figure 16 shows the results which imply that marker quality is slightly affected with the increase in complexity up to 0.76. No major effect on identification occurs below this value. Hence, a value of 0.76 is recommended as complexity threshold *T1*.

Similarly, we also repeated the same experiments for nine markers which have different intensity changes and stored the desired results. Figure 18plots the results, which show that quality is significantly negatively affected as intensity change is increased after 0.016; which means 16/1000 (i.e., 16 intensity changes per row of 1000 pixels).

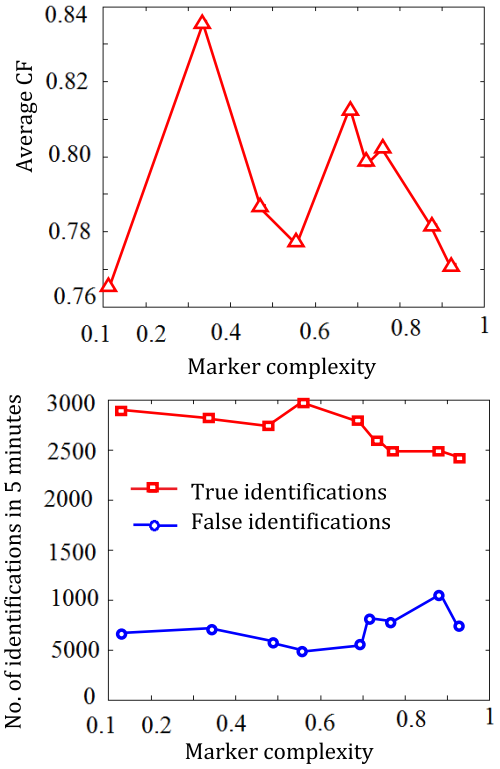


FIGURE 16. Experiments on marker complexity. Top: Graph between marker complexity and average CF values. Bottom: Graph between marker complexity and number of identifications in 5 minutes.

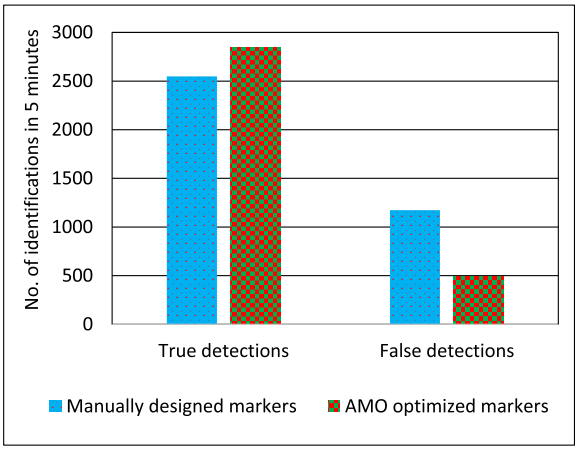


FIGURE 17. Comparison of the manually designed vs FMO optimized markers (by end users). The average CF value for the manually designed markers was 0.80262 whereas for the FMO optimized markers was 0.76752 (calculated from the experiments by 5 users for 5 minutes each).

Hence, the recommended value for intensity change threshold $T2$ is 0.016 (used in algorithm 2).

(7) These experiments are concerned with testing the algorithm used for marker classification (i.e., Algorithm 6). Seven markers (see figure 19) were divided in the four different groups using Algorithm 6.

The four classes according to algorithm were:
Class 1: A
Class 2: B and D

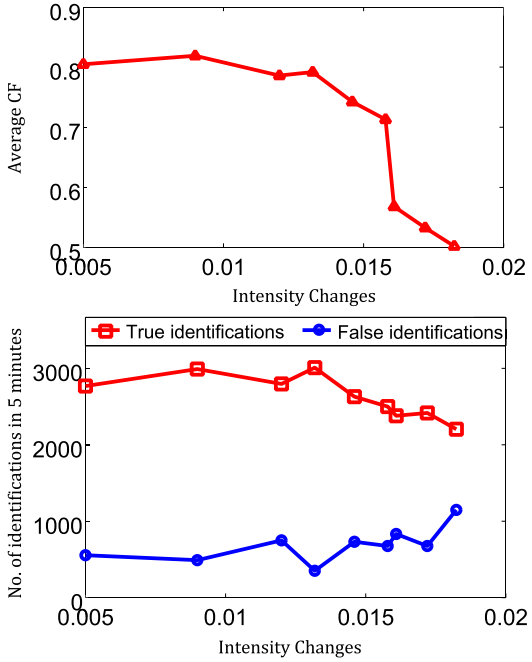


FIGURE 18. Experiments on intensity changes. Top: Graph between intensity changes and average CF values Bottom: Graph between intensity changes and number of identifications in 5 minutes.

FIGURE 19. Seven markers used for testing the classification algorithms.

Class 3: C and G
Class 4: E and F
Similarly for $k = 3$ the results were:
Class 1: A
Class 2: B, E and F
Class 3: C, D and G
For $k = 2$, the grouping was:
Class 1: A, B, E and F
Class 2: C, D and G
The marker from the smaller-sized Class when the total number of Classes (k value) is greater will be additionally distinct in the library. For example, in the above experiments marker A is more distinct than the 7 markers. For $k = 4$, the Class size is 1; although its class size is 4 when $k = 2$. The described distinction of marker A can also be understood, as shown

TABLE 4. A 7 × 7 *Results* matrix, classifying seven markers *A, B, C, D, E, F,* and *G*, into four distinct classes.

Marker	A	B	C	D	E	F	G
A	1.0000	0.3735	0.2669	0.2308	0.3642	0.4523	0.2919
B	0.3735	1.0000	0.4069	0.7069	0.6736	0.6074	0.4720
C	0.2669	0.4069	1.0000	0.5436	0.5485	0.4273	0.5965
D	0.2308	0.7069	0.5436	1.0000	0.4990	0.3785	0.6132
E	0.3642	0.6736	0.5485	0.4990	1.0000	0.7229	0.3941
F	0.4523	0.6074	0.4273	0.3785	0.7229	1.0000	0.3312
G	0.2919	0.4720	0.5965	0.6132	0.3941	0.3312	1.0000
Class	**1**	**2**	**3**	**2**	**4**	**4**	**3**

in Table 4, where the maximum similarity of marker *A* with each value for the rest in the table is less than the remaining values in the Table.

V. CONCLUSION

We designed a tool called FMO to design high quality ARToolKit markers. The FMO enables the marker designers to individually upload all markers of the library and optimize their design attributes, such as B:W ratio, edge sharpness, and information complexity, and checks for possible confusion with other markers of the library. The reduction of inter-marker confusion is the most important factor in this study which has been achieved by ensuring marker distinction in the library. Similarly, to make markers increasingly visible is also a key factor. The visibility of the marker has been enhanced through removal of noise and edge sharpness. FMO provides various cognitive aids for marker designers at various stages of marker enhancement. In addition, the marker classifier module classifies the markers into a given number of classes based on their degree of similarities for reduction of inter-marker confusion. The key achievements of the FMO are: (1) reducing inter-marker confusion, (2) increasing true detection, and (3) minimizing false detection. The FMO will be used for designing reliable markers in an easy to use fashion and, as an outcome, will considerably improve the efficiency and performance of desired applications.

Limitations: The limitations in the current study are listed below. (1) A number of toolkits exist for marker tracking (as discussed in Section II)). However, our study specifically focuses on ARToolKit. Although the importance of ARToolKit is evident based on recent research, other toolkits are also meritable. Conducting a general study of all marker based systems is, therefore, desirable. (2) This study is limited to black and white markers. Other ARToolKit markers also use colors. (3) For certain markers, such as those that contain complex information, FMO only detects the optimality of attributes (such as information complexity) without automatically optimizing the marker. In this case, the user will manually edit the marker (using an external tool). Hence, the input and output markers may be completely different in structure (see marker 2 in Figure 11).

Future work: We aim to generalize and extend the scope of our study to all marker tracking systems. In addition, we are also interested to introduce a new marker-based interactive toolkit with voice and motion detectors to increase reliability. The additional detectors will extend the functionality of the toolkit and overcome the problems of occlusion and light intensity.

REFERENCES

[1] R. S. Patkar, S. P. Singh, and S. V. Birje, "Marker based augmented reality using Android OS," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 3, no. 5, pp. 64–69, 2007.

[2] R. Sun, Y. Sui, R. Li, and F. Shao, "The design of a new marker in augmented reality," in *Proc. Int. Conf. Econ. Finance Res.*, Singapore, 2011, pp. 129–132.

[3] I. Rabbi and S. Ullah, "A survey on augmented reality challenges and tracking," *Acta Graph.*, vol. 24, nos. 1–2, pp. 29–46, 2013.

[4] N. Thiengtham and Y. Sriboonruang, "Improve template matching method in mobile augmented reality for thai alphabet learning," *Int. J. Smart Home*, vol. 6, no. 3, pp. 25–32, 2012.

[5] C. G. Shetty, U. J. Ujawal, J. Joseph, and K. Chidananda, "Interactive digital learning system (IDLS)," *Int. J. Adv. Res. Comput. Sci. Softw. Eng.*, vol. 2, no. 10, pp. 479–494, 2012.

[6] R. Luke and K. Bradshow, "Fiducial marker navigation for mobile robots," Dept. Coumput. Sci., Rhodes Univ., Grahamstown, South Africa, Tech. Rep., 2012, pp. 129–132. [Online]. Available: http://www.cs.ru.ac.za/research/g.09r5654/index.html

[7] H. Kato and M. Billinghurst, "Marker tracking and HMD calibration for a video-based augmented reality conferencing system," in *Proc. 2nd IEEE ACM Int. Workshop Augmented Reality (IWAR)*, San Francisco, CA, USA, Oct. 1999, pp. 85–94.

[8] J. R. Vallino and C. M. Brown, "Interactive augmented reality," Ph.D. dissertation, Dept. Comput. Sci., College Arts Sci., Univ. Rochester, New York, NY, USA, 1998.

[9] M. Fiala, "Designing highly reliable fiducial markers," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 32, no. 7, pp. 1317–1324, Jul. 2010.

[10] R. M. Freeman, S. J. Julier, and A. J. Steed, "A method for predicting marker tracking error," in *Proc. 6th IEEE ACM Int. Symp. Mixed Augmented Reality (ISMAR)*, Nov. 2007, pp. 157–160.

[11] T. Vriends and H. Corporaal, "Evaluation of high level synthesis for the implementation of marker detection on FPGA," M.S. thesis, Eindhoven Univ. Technol., Eindhoven, The Netherlands, 2011. [Online]. Available: http://alexandria.tue.nl/extra1/afstversl/wsk-i/vriends2011.pdf

[12] M. Fiala, "ARTag, a fiducial marker system using digital techniques," in *Proc. Comput. Vis. Pattern Recognit.*, Ottawa, ON, Canada, Jun. 2005, pp. 590–596.

[13] D. Khan, S. Ullah, and I. Rabbi, "Factors affecting the design and tracking of ARToolKit markers," *Comput. Standards Interfaces*, vol. 41, pp. 56–66, Sep. 2015.

[14] D. Khan, S. Ullah, and I. Rabbi, *Sharp-Edged, De-Noised, and Distinct (SDD) Marker Creation for ARToolKit*. Cham, Switzerland: Springer, 2014, pp. 396–407.

[15] D. Khan, S. Ullah, and I. Rabbi, "Classification of markers in the ARTool kit library to reduce inter-marker confusion," in *Proc. 12th Int. Conf. Frontiers Inf. Technol. (FIT)*, Islamabad, Pakistan, Dec. 2014, pp. 269–273.

[16] D. Khan, S. Ullah, and I. Rabbi, "Optimality of black to white ratio and information complexity for robust marker recognition," in *Proc. 4th Int. Conf. Image Process. Theory, Tools Appl. (IPTA)*, Paris, France, Oct. 2014, pp. 283–288.

[17] S. Siltanen and V. teknillinen Tutkimuskeskus, *Theory and Applications of Marker-Based Augmented Reality* (VTT Science). Espoo, Finland: VTT, 2012.

[18] H. Wu, F. Shao, and R. Sun, "Research of quickly identifying markers on augmented reality," in *Proc. IEEE Int. Conf. Adv. Manage. Sci. (ICAMS)*, Chengdu, China, Jul. 2010, pp. 671–675.

[19] V. F. da Camara Neto, D. B. de Mesquita, R. F. Garcia, and M. F. M. Campos, "On the design and evaluation of a precise scalable fiducial marker framework," in *Proc. 23rd SIBGRAPI Conf. Graph., Patterns Images*, Aug./Sep. 2010, pp. 216–223.

[20] J. Rekimoto and Y. Ayatsuka, "CyberCode: Designing augmented reality environments with visual tags," in *Proc. DARE Designing Augmented Reality Environ. (DARE)*, New York, NY, USA, 2000, pp. 1–10.

[21] A. C. Rice, A. R. Beresford, and R. K. Harle, "Cantag: An open source software toolkit for designing and deploying marker-based vision systems," in *Proc. 4th Annu. IEEE Int. Conf. Pervasive Comput. Commun. (PerCom)*, Mar. 2006, pp. 12–21.

[22] *Vuforia Developer Portal*. Accessed: Jan./Apr. 21, 2017. [Online]. Available: https://developer.vuforia.com/

[23] E. Olson, "AprilTag: A robust and flexible visual fiducial system," in *Proc. IEEE Int. Conf. Robot. Autom. (ICRA)*, May 2011, pp. 3400–3407.

[24] A. Richardson, J. Strom, and E. Olson, "AprilCal: Assisted and repeatable camera calibration," in *Proc. IEEE/RSJ Int. Conf. Intell. Robots Syst. (IROS)*, Nov. 2013, pp. 1814–1821.

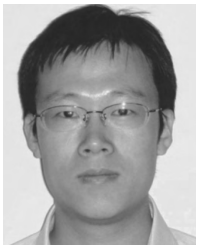
[25] S. Garrido-Jurado, R. Muñoz-Salinas, F. Madrid-Cuevas, and M. Marín-Jiménez, "Automatic generation and detection of highly reliable fiducial markers under occlusion," *Pattern Recognit.*, vol. 47, no. 6, pp. 2280–2292, Jun. 2014.

[26] H. Kato, M. Billinghurst, and I. Poupyrev. (2000). *Artoolkit Version 2.33*. Accessed: Aug. 17, 2013. [Online]. Available: http://www.tinmith.net/lca2004/ARToolkit/ARToolkit2.33doc.pdf

[27] (2016). *ARToolKit: The World's Most Widely Used Tracking Library for Augmented Reality*. Accessed: Jun. 8, 2015. [Online]. Available: http://artoolkit.org/
[28] D. Wagner and D. Schmalstieg, "ARToolKitPlus for pose tracking on mobile devices," in *Proc. 12th Comput. Vis. Winter Workshop (CVWW)*, 2007, pp. 139–146.
[29] M. Fiala, "Comparing ARTag and ARToolkit Plus fiducial marker systems," in *Proc. IEEE Int. Workshop Haptic Audio Vis. Environ. Appl. (HAVE)*, Ottawa, ON, Canada, Oct. 2005, pp. 148–153.
[30] M. A. Kumar and K. SahityaPriyadharshini, "A survey of ARToolkitBased augmented reality applications," *J. Comput. Appl.*, vol. 5, no. EICA2012-3, pp. 261–264, 2012.
[31] M. Badeche and M. Benmohammed, "Classification of the Latin alphabet as pattern on AR toolkit markers for augmented reality applications," *World Acad. Sci., Eng. Technol.*, vol. 6, no. 9, pp. 386–390, 2012.
[32] I. Rabbi, S. Ullah, S. U. Rahman, and A. Alam, "Extending the functionality of ARToolKit to semi-controlled/uncontrolled environment," *Int. J. Inf. Inst.*, vol. 17, no. 6B, pp. 2823–2832, 2013.
[33] D. F. Abawi, J. Bienwald, and R. Dorner, "Accuracy in optical tracking with fiducial markers: An accuracy function for ARToolKit," in *Proc. 3rd IEEE/ACM Int. Symp. Mixed Augmented Reality (ISMAR)*, Washington, DC, USA, Nov. 2004, pp. 260–261.
[34] P. Malbezin, W. Piekarski, and B. H. Thomas, "Measuring ARTootKit accuracy in long distance tracking experiments," in *Proc. 1st IEEE Int. Workshop Augmented Reality Toolkit*, Sep. 2002, pp. 1–2.
[35] M. Billinghurst and B. Thomas, "Mobile collaborative augmented reality," in *Recent Trends of Mobile Collaborative Augmented Reality Systems*, L. Alem and W. Huang, Eds. New York, NY, USA: Springer, 2011, pp. 1–19.
[36] A. Zeb, S. Ullah, and I. Rabbi, "Indoor vision-based auditory assistance for blind people in semi controlled environments," in *Proc. 4th Int. Conf. Image Process. Theory, Tools Appl. (IPTA)*, Paris, France, Oct. 2014, pp. 398–403.
[37] I. U. Rehman, S. Ullah, and I. Rabbi, "Measuring the student's success rate using a constraint based multi-modal virtual assembly environment," in *Proc. 1st Int. Conf. Augmented Virtual Reality (AVR)*, Lecce, Italy, 2014, pp. 53–64.
[38] E. Hornecker and T. Psik, "Using ARToolKit markers to build tangible prototypes and simulate other technologies," in *Human-Computer Interaction—INTERACT* (Lecture Notes in Computer Science), vol. 3585. Berlin, Germany: Springer, 2005, pp. 30–42. [Online]. Available: http://link.springer.com/chapter/10.1007/11555261_6#citeas
[39] I. U. Rehman, S. Ullah, and I. Rabbi, "The effect of semantic multi-modal aids using guided virtual assembly environment," in *Proc. Int. Conf. Open Source Syst. technol. (ICOSST)*, vol. , Dec. 2014, pp. 87–92.
[40] (2008). *FLARToolKit Marker Generator Online*. Accessed: Jan. 17, 2016. [Online]. Available: http://flash.tarotaro.org/blog/2008/12/14/artoolkit-marker-generator-online-released/
[41] (2013). *Augmented Reality Marker Generator*. Accessed: Jan. 27, 2016. [Online]. Available: http://www.brosvision.com/ar-marker-generator/
[42] (2013). *ARToolKit Marker Maker*. Accessed: Jan. 27, 2016. [Online]. Available: http://www.roarmot.co.nz/ar/
[43] I. Rabbi, S. Ullah, and D. Khan, "Automatic generation of layered marker for long range augmented reality applications," *Kuwait J. Sci.*, vol. 44, no. 3, pp. 44–55, 2017.
[44] M. Ashok, J. SreeDevi, and M. R. Bai, "An approach to noisy image skeletonization using morphological methods," *Int. J. Sci. Eng. Res.*, vol. 3, no. 9, pp. 620–627, 2012.

DAWAR KHAN received the bachelor's and master's degrees from the Department of Computer Science and IT, University of Malakand Pakistan, in 2011 and 2014, respectively. He is currently pursuing the Ph.D. degree with the National Laboratory of Pattern Recognition (NLPR), Institute of Automation, Chinese Academy of Sciences, Beijing, China. He has been with NLPR since 2015. He is a recipient of the Chinese Government Scholarship for his Ph.D. degree. His research interests include computer graphics, computational geometry, mesh processing, virtual and augmented reality, and pattern recognition.

SEHAT ULLAH received the M.Sc. degree in computer science from the University of Peshawar, Pakistan, and the master's degree in complex systems and virtual reality and the Ph.D. degree in robotics from the University of Évry Val d'Essonne, France. He is currently an Assistant Professor with the Department of Computer Science and IT, University of Malakand, Pakistan. He is also supervising a number of master's and Ph.D. students. His research interests include computer graphics, virtual and augmented reality, computer vision, HCI, pattern recognition, and technology enhanced learning.



DONG-MING YAN received the master's and bachelor's degrees in computer science and technology from Tsinghua University in 2005 and 2002, respectively, and the Ph.D. degree in computer science from Hong Kong University in 2010. He is currently an Associate Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academy of Sciences. His research interests include computer graphics, geometric processing, and visualization.

IHSAN RABBI received the master's degree in computer science from the University of Peshawar, Pakistan, and the Ph.D. degree in computer science from the University of Malakand, Pakistan. He is currently an Assistant Professor with the Department of Computer Science, University of Science and Technology, Bannu, Pakistan. His research interests include marker-based augmented reality, computer vision, and pattern recognition.

PAUL RICHARD was born in 1964 in Négrepelisse, France. He received the M.S. and Ph.D. degrees in robotics from University Paris VI in 1992 and 1996, respectively. He is currently an Associate Professor with the Université d'Angers, where he leads the virtual reality research activities with the Laboratoire Angevin de Recherche en Ingénierie des Systèmes. For two decades, his research has focused on virtual reality, 3-D interaction, and human performance in virtual environments. He has contributed as either program or conference chair in many international conferences.

THUONG HOANG held a post-doctoral position at the Laboratoire Angevin de Recherche en Ingénierie des Systèmes, Universitéd'Angers, France. He is currently a Lecturer of virtual and augmented reality with Deakin University, Australia. His research experience is in outdoor wearable augmented reality system, interactions for AR systems, and menu systems for AR systems.

MARK BILLINGHURST received the Ph.D. degree from the University of Washington in 2002 with a focus on innovative computer interfaces that explore how virtual and real worlds can be merged. He was with British Telecom, Nokia, Google, and the MIT Media Laboratory. He was the Director of the HIT Lab NZ, University of Canterbury. He is currently a Professor of human computer interaction with the University of South Australia, Adelaide, Australia. He has published over 300 papers in topics, such as wearable computing, augmented reality, and mobile interfaces. In 2013, he was selected as a fellow of the Royal Society of New Zealand. For his MagicBook project, he received the 2001 Discover Award for best entertainment application, and he received the 2013 IEEE VR Technical Achievement Award for contributions to research and commercialization in augmented reality.

XIAOPENG ZHANG received the Ph.D. degree in computer science from the Institute of Software, Chinese Academy of Sciences, in 1999. He is currently a Professor with the National Laboratory of Pattern Recognition, Institute of Automation, Chinese Academic of Sciences. His main research interests include computer graphics and image processing. He received the National Scientific and Technological Progress Prize (second class) in 2004.