***Kuwait J. Sci.* 44 (3)** pp. 44-55, 2017

Automatic generation of layered marker for long range augmented reality applications

Ihsan Rabbi[1,2,*] , Sehat Ullah[1], Dawar Khan[1]

[1]*Dept. of Computer Science and IT, University of Malakand, Pakistan*

[2]*Dept. of Computer Science, University of Science and Technology, Bannu, Pakistan*

**Corresponding author: ihsanrabbi@uom.edu.pk*

Abstract

Marker-based systems are one of the good tracking approaches used for augmented reality applications. Their registration is very accurate and no delay is noticeable, when overlying the virtual information on a real-world scene. However, markers often suffer from a limited tracking range entailing rare use in long range augmented reality applications. This paper presents a method for the automatic generation of a layered marker, a marker that can be tracked from short as well as long distances. An evaluation of a layered marker was performed employing the standard ARToolKit framework. The analysis shows that an automatic created layered marker extends the tracking range. The same marker can be successfully tracked from small and long distances; thus can be used in the development of such augmented reality applications that need long tracking range.

Keywords: Extending tracking range; long range augmented reality applications; marker-based tracking.

1. Introduction

Tracking an object is a key requirement for developing augmented reality (AR) applications. This allows the projection of virtual models at the proper location on real world images. Tracking, the process of estimating the camera pose (virtual or real) in the environment where augmentation takes place, is the most important part of AR systems. Accurate and robust camera tracking are prerequisites for a variety of applications including dynamic scene analysis and interpretation, 3D scene structure extraction and video data compression (Jiang *et al.*, 2000). AR environments in which synthetic objects are inserted into a real scene is a prime candidate since a potentially restricted workspace demands robust and fast tracking from few feature points. The alignment of virtual objects and real-world objects depends on accurate tracking of the viewing pose, relative to the real environment and the annotated objects (Neumann & Majoros, 1998). Therefore in AR applications, it is necessary to track the user movement in six Degrees of Freedom (6DOF) relative to the environment.

In AR applications, tracking can be performed using sensor-based, vision-based and/or hybrid techniques (Zhou *et al.*, 2008). Currently, vision-based tracking approach is the most widely used tracking method in AR applications. Vision-based tracking uses computer vision methods to calculate the camera pose relative to the real-world objects. Vision-based tracking approaches are categorized into marker-based and markerless techniques. In marker-based tracking, artificial markers are placed in the real environment for the development of augmented reality applications. This approach provides fast, accurate and real-time tracking solution for controlled indoor AR applications. Due to the short tracking range of marker-based approach, it is rarely used in the development of such AR applications that need long tracking range.

In this paper, we present a method facilitating the automatic generation of a layered marker and layered marker generation tool (LMGT) is developed on the basis of our previously proposed technique (Rabbi & Ullah, 2015). The layered marker provides a long tracking range as compared to a simple marker (non-layered). The tracking procedure of the layered marker is also presented. The developed layered marker is tested in a real long range augmented reality application. The experimental results show that the proposed marker is easily tracked from small as well as long distances. Using this model, the marker-based tracking can easily be extended to long range AR applications.

The literature review related to marker-based tracking is discussed in section 2. Section 3 contains the proposed marker and the technique to automatically generate it. The developed layered marker is implemented and evaluated in section 4. Additional applications of the proposed approach are discussed in section 5. The last section of the paper contains the conclusion along with future work.

2. Literature review

Fiducial markers are placed in the real environment for marker-based tracking to develop augmented reality applications. Each marker has a unique pattern, which makes it easy to identify its pose relative to the objects in real-world environment. Depending on different patterns inside a marker, it allows the design of many different markers to enable continuous tracking inside a large building (Naimark & Foxlin, 2002). Various tool kits are developed that uses marker-based tracking approach. Examples of such toolkits are ARToolKit (Kato *et al.*, 2000), ARToolKitPlus (Fiala, 2005a), ARTag (Fiala, 2005b), and ALVAR (2013). These toolkits provide a good framework for the development of an AR application. It is required to develop fast and accurate tracking system with less efforts, costs and changes in the environmental setup (Rabbi & Ullah, 2013). Edge detection techniques are used as a pre-processing step for image processing applications (Rangasamy & Subramaniam, 2017).

Ababsa & Mallem (2004) used corners information of markers to robustly estimate the camera pose. Uematsu & Saito (2007) correctly estimate rotation parameters of the camera by using particle filtering technique. This approach improves the accuracy of the 3D coordinate system. For correct pose parameters estimation, Maidi *et al.* (2010) designed a system that combined extended Kalman filter (Bishop & Welch, 2001) with analytical method (Dhome *et al.*, 1989). This system improves stability, convergence and accuracy of the pose parameters estimation in marker-based approach. Seo *et al.* (2011) used multiple key-points and feature tracking to handle the challenges of marker jitter and occlusion respectively. This system is computationally costly, as it takes more time to estimate the camera pose. The tracking capability of marker-based tracking is extended to semi-controlled environment by using multiple pattern files (Rabbi *et al.*, 2014).

Rabbi *et al.* (2014) analyzed various attributes of Fiducial markers for robust tracking. The attributes includes marker sizes, marker distance from camera, marker speed, brightness in environment, contrast level of lighting and the effect of marker size on tracking distance. Factors affecting the marker tracking are identified by Khan *et al.* (2015). The concept of a nested marker is presented by Tateno *et al.* (2007). The main limitations in their nested marker are that they used four markers inside each marker and the maximum hierarchy is set to three. The upper-layer marker is identified by its lower-layer markers, which may produce inter-marker confusion between upper layer and its lower-layer markers. As the maximum hierarchy is up to level 3, the distance covered is also limited (Teteno *et al.*, 2007).

The challenges of marker-based approach covers, as found in the literature, are its robustness, occlusion, jitter, blurredness, tracking stability and moving marker or camera in x or y directions. The tracking range of marker-based approach is still a challenge, due to which this approach is rarely used in the development of long range indoor AR applications. This paper presents the automatic generation of layered marker that extends tracking range up to a distance of our interest. Different AR applications combine multiple markers to achieve robust geometric registration against partial occlusion. But these systems still have the limitation of individual marker recognition in long range.

3. Layered marker

3.1. Need for layered maker

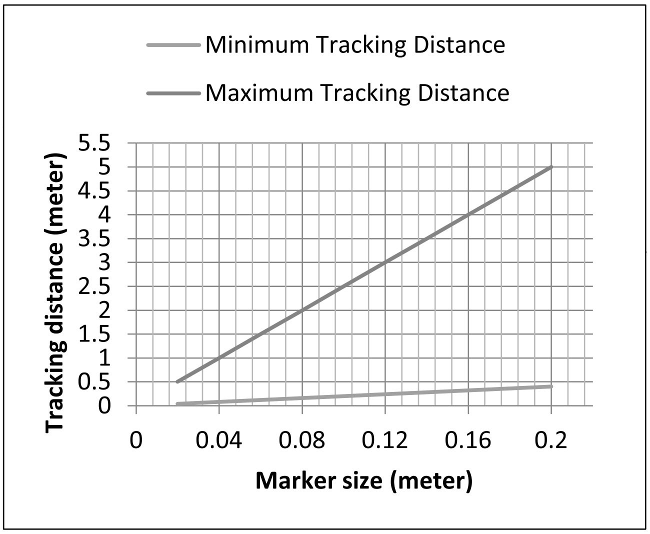


Fig. 1. Minimum and maximum tacking distance of markers

As the marker distance is decreased from the camera, it is zooming-in. The minimum tracking distance will be that distance at which the marker is occluded due to zoom-in. The zoom-in can cause the occlusion of the corner(s) due to which marker tracking failure occurs. Similarly, when a marker is moving away from the camera, it is zooming-out. This zoom-out can cause marker blurredness as the distance approaching to its maximum tracking distance. For example, the marker having size of 0.2 × 0.2 meter can be tracked successfully in the range from 0.4 meter to 5.0 meter.

Using the above analysis, the size and distance relationship is found out as given in Equation (1), (2) and (3)

$$TD_{Min} = MS \times 2 \text{ meters} \quad (1)$$

$$TD_{Max} = MS \times 25 \text{ meters} \quad (2)$$

$$TR = [TD_{Min}, TD_{Max}] \quad (3)$$

Where TD_{Min} and TD_{Max} represent the minimum and maximum tracking distance respectively, MS represents marker size in meter and TR is the tracking range of a marker. A marker will be efficiently tracked inside its tracking range and will give maximum tracking failure if the distance between the camera and marker is beyond this range. In order to verify the tracking range, we designed a module using ARToolKit that will report the marker distance from camera and the tracking errors found while tracking the 'Hiro' marker (See Figure 2).

Fig. 2. Pattern of ARToolKit (Kato *et al.*, 2000) marker

The marker having size of 0.10 × 0.10 meter is considered for this analysis. The marker is tracked from different distances and the tracking errors are recorded against each distance. The data is analyzed as shown in Figure 3.

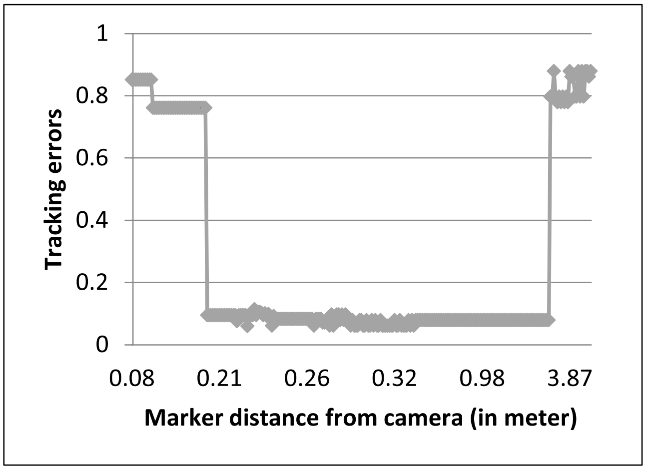


Fig. 3. Marker tracking errors of 0.10 × 0.10 meter marker at various distances

The graph in Figure 3 indicates that marker tracking is performed successfully within the minimum and maximum range (0.20 and 2.5 meter respectively). The tacking failure occurs when the marker distance is beyond the limits.

3.2. System information flow

The basic flow chart of the automatic generation of layered marker is shown in Figure 4. The first step is to take input from the user. The user inputs the minimum and maximum tracking distances as needed for a certain large AR application. The next module calculates the total number of layers required for the layered marker. The system generates the layered marker that contains the required number of layers calculated in the second step. This will be a simple layered marker containing multiple layers. It may create inter-marker confusion between its various layers because of their close similarity with each other. To solve this problem, various shapes, already stored in the database, are added in the white regions inside each layer. The system validates the similarities among various layers. If the similarity is close between any layers, the process goes back to previous module. Some more information is added to the similar layers. At the last step the designed layered marker is stored in the marker database.

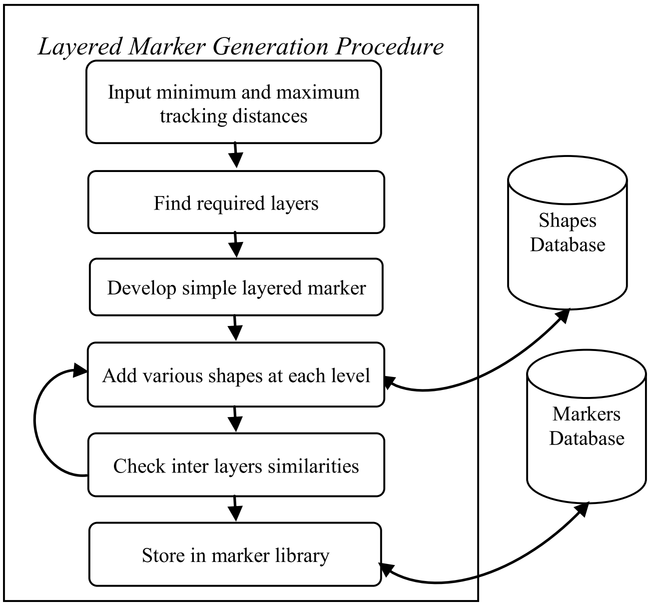


Fig. 4. Procedure for automatic generation of layered marker

3.3. Layered marker generation tool (LMGT)

The layered marker generation tool is developed by using MATLAB. The LMGT required two inputs i.e. minimum tracking distance and maximum tracking distance. Based on the input, number of layers required for layered marker is calculated. Figure 5 shows the basic interface of the implemented system. This tool will take minimum and maximum tracking distances from the user and will automatically generate the required layered marker.

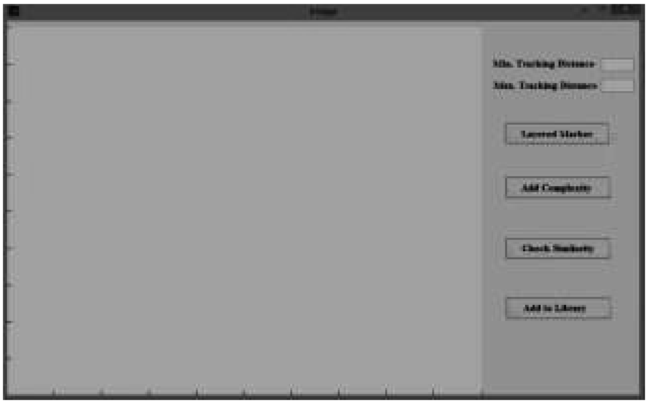


Fig. 5. Snapshot of layered marker generation tool

Let us consider that an augmented reality application needs the tracking range of [0.2, 50] meters. The inputs for the automatic layered marker generator are 0.2 meter and 50 meters. As the layered marker button is pressed, the tool calculates the number of required layers by using the following two steps:

1. Calculate the outermost layer and innermost layer markers size

   (a) $MS_{OL} = \frac{TD_{Max}}{25} \times \frac{TD_{Max}}{25}\ meter \quad (4)$

   (b) $MS_{IL} = \frac{TD_{Min}}{2} \times \frac{TD_{Min}}{2}\ meter \quad (5)$

2. $N = 1$

   $While\ (MS_{OL} > MS_{IL})$

   (a) $MS_{OL} = \frac{MS_{OL}}{2}$

   (b) $N++$

The size of each layer is calculated as:

$$Size(L_j) = \frac{Size(L_{j+1})}{2}$$

where $Size(L_j)$ represents the size of L_j layer.

N contain the number of layers required. The tool will draw the required number of layers as shown in the Figure 6.

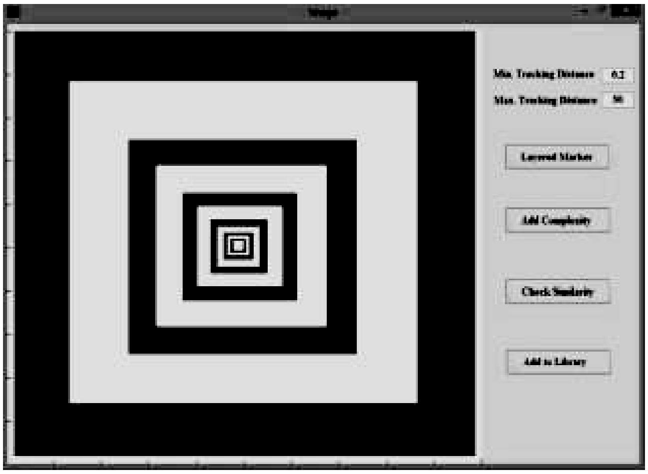


Fig. 6. Input given to layered marker generation tool

The numbers of layers are calculated based on the above two steps. In database, various shapes including alphabets, numerals, mathematical and irregular are stored. These shapes are used to add some complexity at each layer to minimize inter-marker confusion among layers. Therefore the next step is formulated to find total number of required markers such that they have less similarity. This is calculated by using the coefficient correlation between shapes insides two markers as.

```
$For\ i = 1\ to\ N$
    $For\ j = 1\ to\ i$
        $r(M_i, M_j) < 0.1$
```

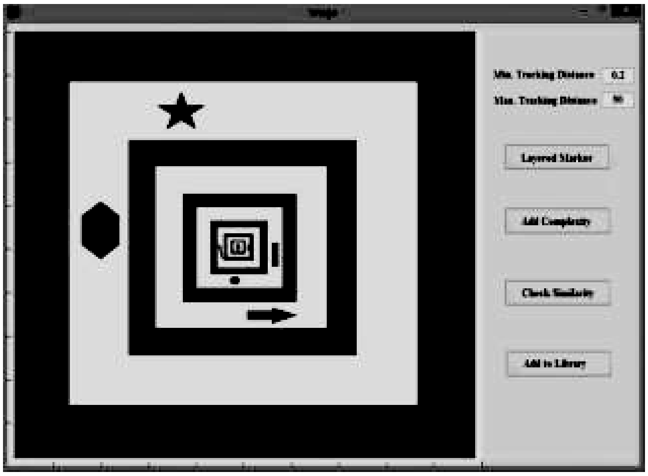
Where r represents the coefficient correlation between shapes inside markers M_i and M_j

The formula for calculating coefficient correlation is as below:

$$r = \frac{\sum_p (x_p - x_m)(y_p - y_m)}{\sqrt{\sum_p (x_p - x_m)^2}\sqrt{\sum_p (y_p - y_m)^2}}$$

Where x_p is the intensity of pth pixel of shapes in marker L_i and y_p is the intensity of pth pixel of shape in marker L_j. x_m and y_m are mean intensities of shapes L_i and L_j respectively.

The 'add complexity' button is used to include shapes from the database to layered marker. Figure 7 shows this procedure.



Min. Tracking Distance 0.2

Max. Tracking Distance 50

Fig. 7. Add complexity to individual layer

The system generates the required layered marker. After this the similarity between each layer is checked to find that each layer is considerably different. One can save this marker in marker library by clicking the button 'add to library'.

The LMGT is tested on various inputs to design the layered marker for different applications. The inputs for the tool is taken as $[100, 1]$, $[83, 0.8]$, $[250, 1.5]$ and $[178, 0.8]$ meters. The output layered markers for these inputs are shown in Figure 8.

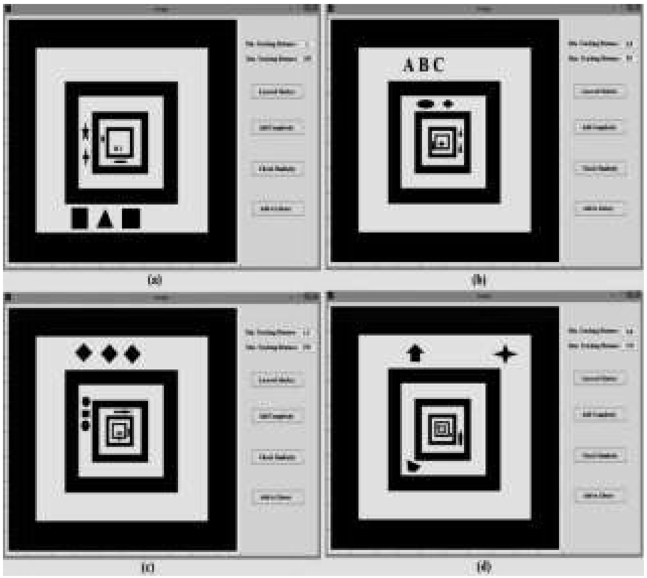


Fig. 8. LMGT on various inputs (a) $[\mathbf{100}, \mathbf{1}]$, (b) $[\mathbf{83}, \mathbf{0.8}]$, (c) $[\mathbf{250}, \mathbf{1.5}]$, (d) $[\mathbf{178}, \mathbf{0.8}]$

3.4. Large indoor industrial AR application (an example)

Consider a large indoor industrial environment that needs an AR application. The required maximum and minimum tracking distances are 50 and 0.2 meters respectively. So the two variables are initialized as:

$$TD_{Max} = 50\ meter$$

$$TD_{Min} = 0.2\ meter$$

The above algorithm is used to design the required layered marker for this environment is as: First calculate the marker size at inner-most and outer-most level using Equations (4) and (5).

$$MS_{IL} = \frac{TD_{Min}}{2} \times \frac{TD_{Min}}{2}\ meter$$

$$MS_{IL} = \frac{0.2}{2} \times \frac{0.2}{2} = 0.1 \times 0.1\ m$$

And

$$MS_{OL} = \frac{TD_{Max}}{2} \times \frac{TD_{Max}}{2}\ meter$$

$$MS_{OL} = \frac{50}{25} \times \frac{50}{25} = 2\ \times 2\ m$$

Now find the total number of layers required for the development of AR application in this environment. For this calculation the next step of the algorithm is applied.

The step produces that six layers are required. The sizes of each layer from outer-most layer to inner-most layer are $2 \times 2\ m, 1 \times 1\ m, 0.5 \times 0.5\ m, 0.25 \times 0.25\ m,$ $0.125 \times 0.125\ m$ and $0.0625 \times 0.0625\ m$ respectively. Take six markers having less correlation coefficient among them to avoid inter-marker confusion. The layered marker is designed by placing the largest marker at outer-most layer and the next marker at next level and so on as shown in Figure 9.

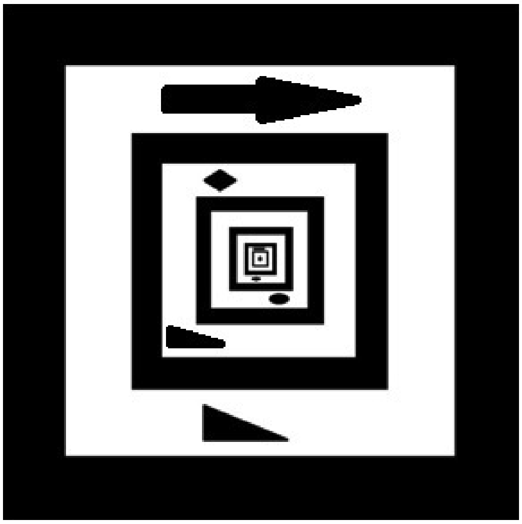


Fig. 9. Sample of a layered marker for long range augmented reality application

Equations (1) and (2) are used to find the tracking range of any size of marker. Using these equations, the minimum tracking distance (TD_{Min}) and maximum tracking distance (TD_{Max}) of each marker of the layered marker is calculated. The TD_{Max_O} of outer most layer will be the upper tracking distance of the proposed layered marker and the TD_{Min_i} of the inner most layer will be the lower tracking distance of layered marker. So the layered marker tracking distance will range from TD_{Max_O} to TD_{Min_i}. The other layers ranges are calculated as

$$L_j = \left[\frac{TD_{Max_{j-1}} + TD_{Min_j}}{2}, \frac{TD_{Max_j} + TD_{Min_{j+1}}}{2}\right] \quad (6)$$

This will be the minimum and maximum distance range for each layer. This improves the layered marker tracking in case of occlusion at any layer. The marker tracking is shifted to the adjacent lower layer. The virtual contents are placed according to the tracked layer.

For the above layered marker, the minimum and maximum tracking distances for this layered marker are 0.125 meter and 50 meters respectively. The tracking distance of each layer is calculated as:

$$L_6 = \left[\frac{TD_{Max_{6-1}} + TD_{Min_6}}{2}, 50\right]$$

$$L_6 = \left[\frac{TD_{Max_5} + TD_{Min_6}}{2}, 50\right]$$

$$L_6 = \left[\frac{25+4}{2}, 50\right]$$

$$L_6 = \left[\frac{29}{2}, 50\right]$$

$$L_6 = [14.5,\ 50]$$

So, the minimum and maximum tracking distance of layer six is set to 14.5 and 50 meters respectively.

Using Equation (6), the tracking distances of others layers are calculated and given as:

$$L_5 = [7.25,\ 14.5]$$

$$L_4 = [3.625,\ 7.25]$$

$$L_3 = [1.8125,\ 3.625]$$

$$L_2 = [0.90625,\ 1.8125]$$

$$L_1 = [0.125,\ 0.90625]$$

4. Tracking efficiency of generated layered marker

4.1. Experimental protocol

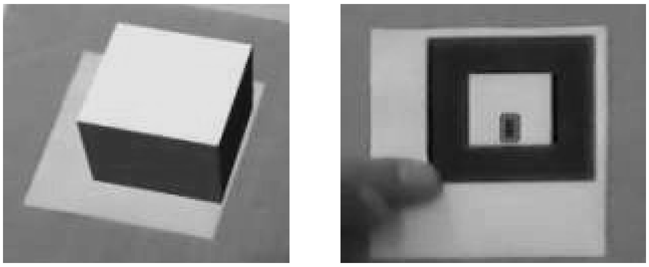
ARToolKit (Kato *et al.*, 2000) library is used to evaluate the proposed automatic layered marker. Three modules are designed for this purpose; the first for checking the marker tracking against partial occlusion, the second one for comparing our proposed layered marker with simple markers of different size and third one for validating the tracking results of automatic design layered marker against manually designed layered marker.

The last analysis is performed to validate the efficiency of LMGT. For this validation, eight volunteers are considered to evaluate the LMGT. Each participant is familiar with marker-based tracking, its use and marker design. The participants belong to the virtual reality and intelligence system group of University of Malakand. The steps for developing layered marker are explained to every participant and are asked to design layered marker for four environments. Initially one example is solved that explain the complete procedure for developing a layered marker. They are asked to develop layered markers using manual and through LMGT. In both cases, the time taken to calculate the number of layers required for each scenario, the designing of layers in marker, adding the complexity to each layer are recorded for individual participant.

The experiments are performed by using Sony VAIO laptop with specification of 2.4GHZ processor and 4GB RAM. Built-in NVIDIA graphics card and a webcam having resolution of 640×480 pixels are attached with laptop.

4.2. Evaluation

A simple marker and layered marker having six layers are considered for the first experiment. Figure 10 shows the simple marker tracking under partial occlusion. The marker tracking process fails when a small portion of the marker is occluded.



(a) **(b)**

Fig. 10. Simple marker tracking (a) without occlusion (b) with occlusion

On the other hand the layered marker is tracked under partial occlusion and is shown in Figure 11.

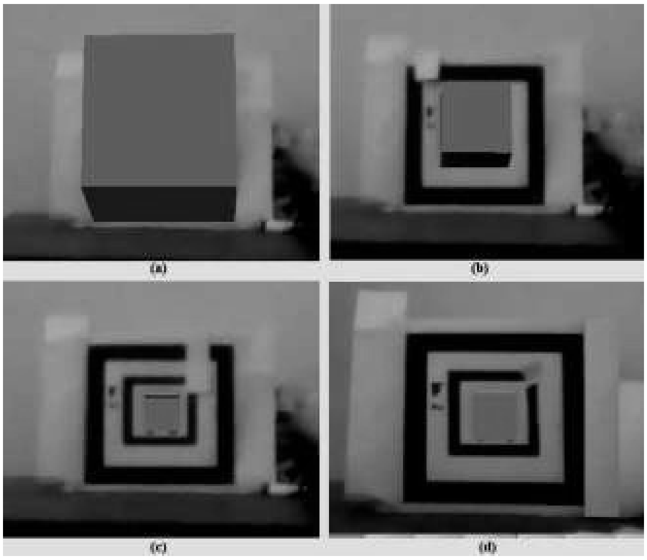
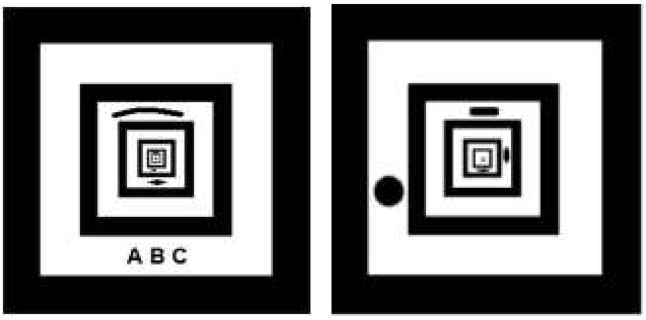


Fig. 11. Layered marker tracking (a) without occlusion (b) occluding upper layer (c) occluding upper two layers (d) occluding the second layer

In Figure 11, the layered marker is occluded partially and the result reveals that the tracking is still performed well under occlusion. In Figure 11 (a) there is no occlusion, so the outermost layer is tracked and the virtual information is displayed on it. When the outmost layer is occluded in Figure 11 (b) (use white paper to occlude the upper layer), the system automatically shifts the tracking to next lower level. Again the top two layers are occluded in Figure 11 (c) the system tracks the next lower layer of the layered marker. Similarly in Figure 11 (d) the second layer is occluded, which produces the tracking of its next lower level.

For the second experiment, a layered marker having six layers and six simple markers are considered. The size of layered marker is $20 \times 20\ cm$ having innermost layer of size $0.625 \times 0.625\ cm$ (Figure 12). The sizes of simple markers are equal to the size of each layer in layered marker. For this purpose 'sample1' (pattern in ARToolKit) marker is considered. The layered marker extends the tracking range of marker to a certain application where a single simple marker cannot be applicable.



(a) **(b)**

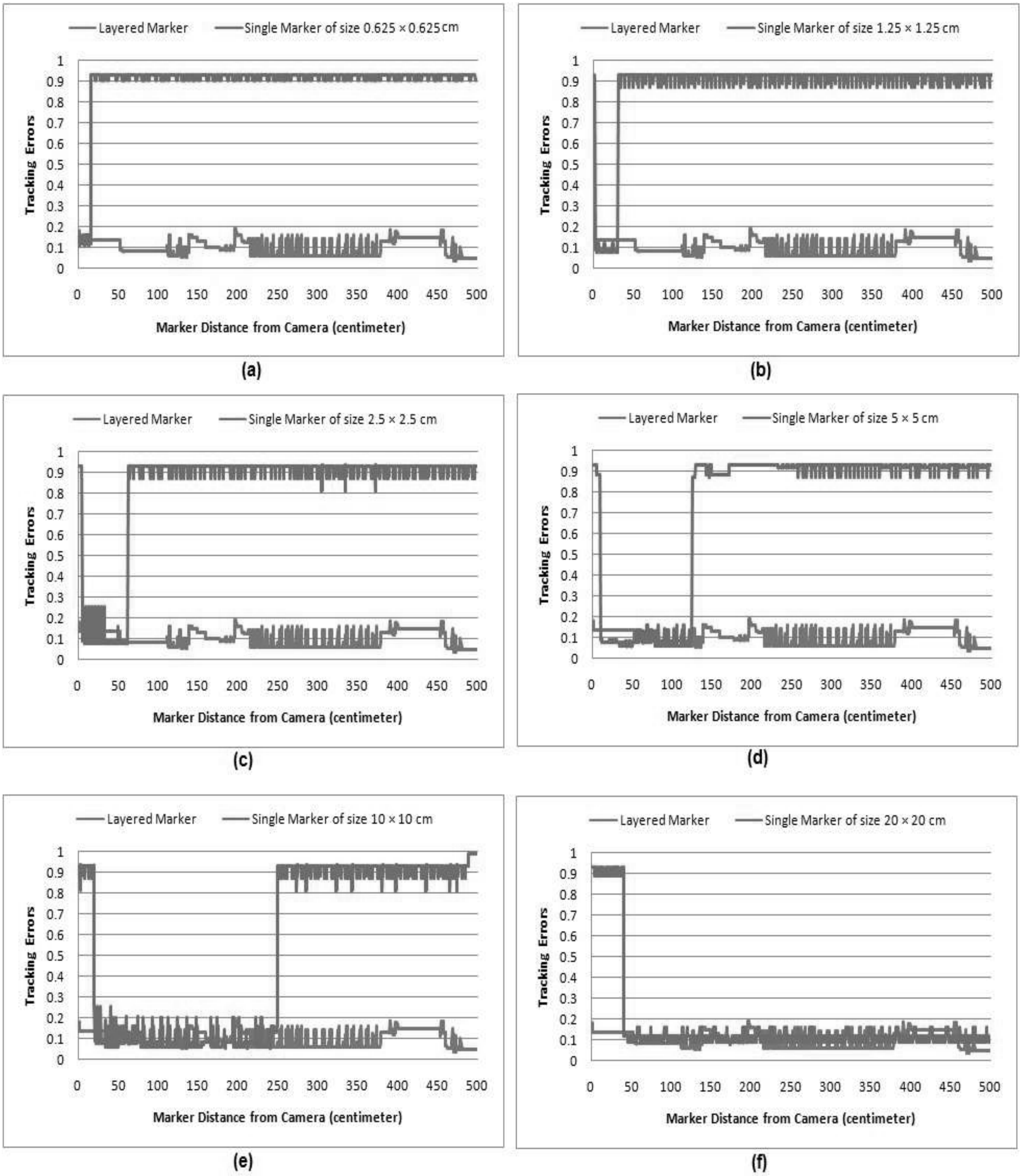
Fig. 12. Layered marker for experiments (a) automatic generated through LMGT (b) manually developed

The experiments produce the marker tracking errors at various distances. The marker tracking errors for these experiments ranges from zero to one. The zero indicates tracking is fully successful, while one indicates complete failure. The threshold of these errors is set to 0.5. The marker tracking will be successful if the tracking errors are below the threshold value. Above the threshold value the system will report no marker detection.

Figure 13 shows the marker tracking errors of layered marker and six single marker having sizes equal to each layer size. The graphs indicate that the successful tracking of single markers are achieved only in a small tracking range of distance while the automatic layered marker produces less tracking errors in the entire distance range.

In Figure 13 (a), the automatic generated layered marker is compared with single marker having size $0.625 \times 0.625\ cm$. The results show that single marker tracking failure occurs at distance of 16 and above while the layered marker performs the tracking throughout the specified range. The Figure 13 (b) compare single marker having size $1.25 \times 1.25\ cm$ with layered marker. The graph indicates that the tracking range of single marker is 2.5 cm to 32 cm. Beyond this range the marker will not be tracked and hence generating maximum tracking errors.

Similarly, the single markers having sizes equal to $2.5 \times 2.5\ cm$, $5 \times 5\ cm$, $10 \times 10\ cm$, and $20 \times 20\ cm$ are compared with automatic generated layered marker and the results are visualized in Figure 13 (c), (d), (e) and (f) respectively. Each single marker is successfully tracked within its tracking range. The layered marker can be tracked from any place throughout the tracking distance of the developed application.



(a) (b) (c) (d) (e) (f)

Fig. 13. Comparison between layered marker tracking with single maker of various sizes of (a) $\mathbf{0.625 \times 0.625\ cm}$ (b) $\mathbf{1.25 \times 1.25\ cm}$ (c) $\mathbf{2.5 \times 2.5\ cm}$ (d) $\mathbf{5 \times 5\ cm}$ (e) $\mathbf{10 \times 10\ cm}$ (f) $\mathbf{20 \times 20\ cm}$

For the efficiency of LMGT, the eight participants were asked to develop layered marker for the inputs [1, 100], [0.5, 85], [0.6, 48] and [1.2, 133] meters. They were asked to use manual method for calculating the layers required and for the development of layered markers. After that they were asked to use LMGT for the same inputs and the corresponding developing time (in seconds) was recorded. The mean and standard deviation of both methods are shown in Table 1. The mean completion time of manual method is 125.41 seconds and LMGT is 35.62 seconds.

Table 1. Mean and standard deviations of developing time in second

	N	Mean	Std. Deviation
Manual	8	125.41	9.319
LMGT	8	35.62	2.375

To test that there is significant difference of designing time of layered markers using manual and LMGT, we use ANNOVA test. The test question is as: is there any significant difference between the mean time of designing layered marker using manual and LMGT?

So, our null and alternative hypotheses are as:

$$H_0: \mu_1 = \mu_2$$

$$H_a: \mu_1 \neq \mu_2$$

Where μ_1 and μ_2 are the mean time taken to design layered markers using manual and LMGT respectively. The values of F and ρ is calculated by using SPSS and is given as:

$$F(1, 14) = 697.28$$

$$\rho < 0.001$$

The value of $\rho < 0.05$, there is significant difference between the two mean, so we reject our null hypothesis and accept alternative hypothesis that the two mean are significantly different from each other.

The total mean time with error bar is shown in Figure 14. The graph indicates that using LMGT for developing layered marker in any environment takes less time than running the steps of layered marker development manually.

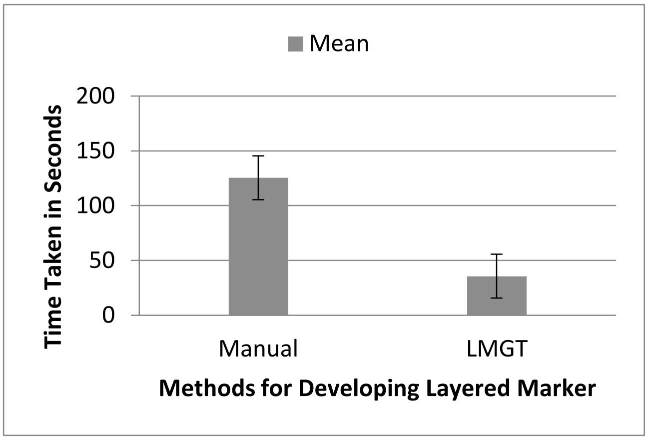


Fig. 14. Difference between standard deviation of manual and LMGT

The detail set of average time taken in different steps are displayed in Table 2. In the first two steps (i.e. calculating the number of layers and designing the layers), the time spent in manual method is much more than LMGT. The last step of adding complexity to various layers is somehow equal in both cases. Table 2 indicates that the average manual development is taking much more time than the LMGT. In each case there is significant difference in designing time.

The next experiment is carried out to validate the results of automatic layered marker with the manually designed layered marker. Figure 15 shows the comparison of both layered markers.

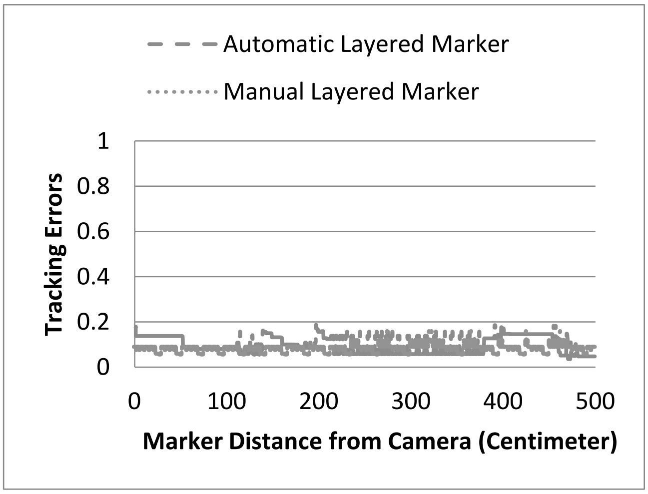


Fig. 15. Comparison between automatic and manual layered marker

The graph shows that the automatically generated layered marker through LMGT tracks successfully in the required range. The above discussion indicates that the layered marker increases the tracking range of marker.

Table 2. Average time (in second) taken for various steps in designing layered marker by using manual and LMGT methods against each participant

Participant	Calculating Layers		Designing Layers		Adding Objects in Layers		Total Time	
	Manual	LMGT	Manual	LMGT	Manual	LMGT	Manual	LMGT
P1	61	2.25	50.5	8	26	23.75	137.5	34
P2	45.25	2.25	43.75	8	25.5	24.75	114.5	35
P3	37	2.25	47	8	25.25	22.5	109.25	32.75
P4	45.25	2.25	42.75	8	38.25	23.25	126.25	33.5
P5	41.75	2.25	51	8	40.25	24.75	133	35
P6	48	2.25	44.75	8	37	27	129.75	37.25
P7	40.25	2.25	47.75	8	37.5	27.75	125.5	38
P8	50.75	2.25	54	8	22.75	29.25	127.5	39.5

5. Applications of layered marker

The layered marker provides the following advantages while developing AR applications:

- The layered marker increases the tracking distance range. It can be used in the development of AR applications where long distance is required.
- A new navigation technique i.e. layer by layer navigation inside virtual environments. One can define different levels of speed at different layers.
- New interaction techniques in virtual environments can be developed using the layered marker.
- Another important application of layered marker is its tracking capability under partial occlusion. When the marker at one layer is occluded by any object in the environment, the camera parameter can be easily estimated for the marker at another visible layer. The tracking module will automatically choose the non-occluded marker at the next higher or lower level.
- This marker can be used for zooming purposes.

6. Conclusion and future work

Vision-based tracking are the most active tracking techniques used in the development of AR applications. Marker-based tracking is a vision based approach in which Fiducial markers are used in the environment for tracking the camera pose. This tracking methodology provides fast, accurate and real-time tracking solution for indoor AR applications, but due to the short tracking range of this approach, it is rarely used for long range applications. This paper introduced the design and implementation for the automatic creation of layered marker that is based on the existing marker-based approach that extends the marker tracking range to be applicable in long range AR environments. The layered marker is designed using the proposed algorithm and is tracked using the proposed tracking methodology. This layered marker is tested in large indoor environment and the evaluation showed that the layered marker increases the marker tracking range reasonably. The automatic layered marker is also tested against manual designed marker.

Layered marker can be used for interaction with virtual world object. At each layer one can proposed interaction of his/her own interest. Our future work will focus to design a predictive model for layered by layered interaction with the objects of virtual environment.

References

Ababsa, F. & Mallem, M. (2004). Robust camera pose estimation using 2D fiducials tracking for real-time augmented reality systems. Proceedings of ACM SIGGRAPH International Conference on Virtual-Reality Continuum and its Applications in Industry (VRCAI'04)**:** 431-435.

ALVAR. (2013). ALVAR – A library for virtual and augmented reality. Retrieved January 16, 2013, from www.vtt.fi/multimedia/alvar.html.

Bishop, G. & Welch, G. (2001). An Introduction to the Kalman Filter. Computer Graphics, Annual Conference on Computer Graphics and Interactive Techniques, (SIGGRAPH '01): 12-17.

Dhome, M., Richetin, M., Lapreste, J.T. & Rives, G. (1989). Determination of the attitude of 3D Objects from a Single Perspective View. IEEE Transaction. Pattern Analysis Machine Intelligence, **11**:265-1278.

Fiala, M. (2005a). ARTag, a fiducial marker system using digital techniques. IEEE Computer Society Conference on Computer Vision and Pattern Recognition, (CVPR'05), **2:**590-596

Fiala, M. (2005b). Comparing ARTag and ARToolkit Plus fiducial marker systems. IEEE International Workshop on Haptic Audio Visual Environments and their Applications (HAVE'05), IEEE Computer

Society: 1-6.

Jiang, B., You, S. & Neumann, U. (2000). Camera tracking for augmented reality media. IEEE International Conference on Multimedia and Expo (III). New York, NY, USA**:** 1637–1640.

Kato, H., Billinghurst, M., Poupyrev, I., Imamoto, K. & Tachibana, K. (2000). Virtual object manipulation on a table-top AR environment. IEEE and ACM International Symposium on Augmented Reality**:** 111-119.

Khan, D., Ullah, S. & Rabbi, I. (2015). Factors affecting the design and tracking of ARToolKit markers. Computer Standards & Interfaces, **41**(0):56-66.

Maidi, M., Didier, J.Y., Ababsa, F. & Mallem, M. (2010). A performance study for camera pose estimation using visual marker based tracking. Machine Vision and Application, **21**(3):365-376.

Naimark, L. & Foxlin, E. (2002). Circular data matrix fiducial system and robust image processing for a wearable vision-inertial self-tracker. International Symposium on Mixed and Augmented Reality, (ISMAR '02): 27-36.

Neumann, U. & Majoros, A. (1998). Cognitive, performance, and systems issues for augmented reality applications in manufacturing and maintenance. IEEE Virtual Reality Annual International Symposium: 4-11.

Rabbi, I. & Ullah, S. (2013). A survey on augmented reality challenges and tracking. Acta Graphica, **24**(1-2):29-46.

Rabbi, I. & Ullah, S. (2015). Extending the tracking distance of fiducial markers for large indoor augmented reality applications. Advances in Electrical and Computer Engineering, **15**(2):59-64.

Rabbi, I., Ullah, S., Rahman, S.U. & Alam, A. (2014). Extending the functionality of ARToolKit to semi-controlled / uncontrolled environment. INFORMATION, **17**(6(B)):2823-2832.

Rangasamy, V. & Subramaniam, S. (2017). Framelet transform based edge detection for straight line detection from remote sensing images. Kuwait Journal of Science, **44**(1):78-85.

Seo, J., Shim, J., Choi, J.H., Park, J. & Han, T.D. (2011). Enhancing Marker-Based AR Technology Virtual and Mixed Reality, Part I, HCII 2011, LNCS 6773: 97–104.

Teteno, K., Kitahara, I. & Ohta, Y. (2007). A nested marker for augmented reality. IEEE Virtual Reality Conference, Charlotte, (VR'07): 259-262.

Uematsu, Y. & Saito, H. (2007). Improvement of accuracy for 2D marker-based tracking using particle filter. 17th International Conference on Artificial Reality and Telexistence, Esbjerg, Jylland, *(ICAT 2007)*: 183-189.

Zhou, F., Duh, H. B.L. & Billinghurst, M. (2008). Trends in augmented reality tracking, interaction and display: A review of ten years of ISMAR. IEEE International Symposium on Mixed and Augmented Reality, (ISMAR '08): 193-202.

Submitted **:** 13/07/2015
Revised **:** 07/01/2017
Accepted **:** 17/01/2017

توليد تلقائي للإشارة الدليلية ذات الطبقات في تطبيقات الواقع المُعزز[1] طويلة المدى

إحسان ربيع[1،2،*]، [1]سيهات أولاه، [1]دوار خان
[1]قسم علوم الحاسوب وتكنولوجيا المعلومات، جامعة مالاكاند، باكستان
[2]قسم علوم الحاسوب، جامعة العلوم والتكنولوجيا، بانو، باكستان
[*]المؤلف المراسل: ihsanrabbi@uom.edu.pk

ملخص

تُعَدُ الأنظمة القائمة على الإشارة الدليلية إحدى طرق التتبع الجيدة التي تُستخدم في تطبيقات الواقع المُعزز. فإن تسجيل هذه الأنظمة دقيق جداً ولم يُلاحظ عليها أي تأخير عند تركيب المعلومات الافتراضية فوق المشهد الحقيقي. ومع ذلك، فغالباً ما تعاني الإشارة الدليلية من نطاق تتبع محدود يستلزم قلة الاستخدام في تطبيقات الواقع المُعزز طويلة المدى. يقدم هذا البحث طريقة للتوليد التلقائي لإشارة دليلية ذات طبقات، وهي إشارة يمكن تتبعها من مسافات قصيرة وطويلة على السواء. تم إجراء تقييم للإشارة الدليلية ذات الطبقات باستخدام إطار أر تول كيت (ArToolKit) القياسي. ويُظهر التحليل أن إشارة دليلية ذات طبقات، والتي تم إنشاؤها تلقائياً، تمتد عبر نطاق التتبع. ويمكن تتبع نفس الإشارة الدليلية بنجاح من المسافات القصيرة والطويلة؛ وبالتالي يمكن استخدامها في تطوير تطبيقات الواقع المُعَزز التي تحتاج إلى نطاق تتبع طويل المدى.

---

[1] تقنيات الحقيقة الافتراضية والحقيقة المُدمجة بين الواقع والخيال (Augmented reality)