Jurnal
Teknologi

Full Paper

AN *S-PI* VISION-BASED TRACKING SYSTEM FOR OBJECT MANIPULATION IN AUGMENTED REALITY

Ajune Wanis Ismail[a], Mark Bilinghust[b], Mohd Shahrizal Sunar[c*]

[a]UTM ViCubeLab, Universiti Teknologi Malaysia, Johor, Malaysia
[b]Human Interface Technology Lab New Zealand, University of Canterbury, Christchurch, New Zealand
[c]UTM-IRDA Digital Media Centre, Universiti Teknologi Malaysia, Johor, Malaysia

Article history
Received
3 December 2013
Received in revised form
2 July 2014
Accepted
25 November 2014

*Corresponding author
shahrizal@utm.my

Graphical abstract

Abstract

In this paper, we describe a new tracking approach for object handling in Augmented Reality (AR). Our approach improves the standard vision-based tracking system during marker extraction and its detection stage. It transforms a unique tracking pattern into set of vertices which are able to perform interaction such as translate, rotate, and copy. This is based on arobust real-time computer vision algorithm that tracks a paddle that a person uses for input. A paddle pose pattern is constructed in a one-time calibration process and through vertex-based calculation of the camera pose relative to the paddle we can show 3D graphics on top of it. This allows the user to look at virtual objects from different viewing angles in the AR interface and perform 3D object manipulation. This approach was implemented using marker-based tracking to improve the tracking in term of the accuracy and robustness in manipulating 3D objects in real-time. We demonstrate our improved tracking system with a sample Tangible AR application, and describe how the system could be improved in the future.

Keywords: Component, augmented reality, vision-based tracking, gesture interaction, object manipulation

1.0 INTRODUCTION

Augmented Reality (AR) merges the virtual world with the psychical world in real time [1]. One of the most important research areas in AR is creating appropriate interaction techniques for AR applications to allow users to seamlessly interact with virtual content [3]. Many different interaction methods have been explored including using object motion [2], mouse input [4], magnetic tracking [5], real objects [6], penand tablet [7] and even natural gestures [8]. Researchers have also explored on interaction in an AR environment can imitate manipulation tasks in the real world [9].

In this research we are exploring how the Single-point Intersection (S-PI) algorithm [10] can be used to provide robust marker-based tracking in AR for object manipulation. Using the S-PI tracking system, the markers are represented by five vertices, which leads to improved tracking. In this paper we describe an accurate tracking system that we have develop during this S-PI approach. The robustness of the tracking system is explained and an intuitive paddle-based metaphor is explored. We show how 3D object manipulation methods works with our tracking system and demonstrate how it can be applied in an example AR application.

2.0 RELATED WORK

Our research is inspired by earlier work in tangible user interfaces (TUIs), which are interfaces where users can manipulate digital information with physical objects [11]. The TUI approach can also be used to interact with AR content as shown by Kato's VOMAR system [12] which uses a physical paddle to interact with AR content. Using real objects for input in an AR interface provides a very intuitive way to interact with the virtual content. This combination of TUI and AR has been called Tangible AR, with the goal of manipulating AR content as easily as interacting with real objects.

For input in Tangible AR interfaces, paddles, props, and cups have been especially designed by considering external properties such as shape, size, and weight [13, 14, 15]. For many of these objects marker-based camera pose estimation approaches [16] are used with the help of robust detection offiducial markers [17]. However, less research has been conducted on methods for natural user interaction, such as robust manipulator tracking with two-handed tool-based operation, and fast wireless input [15]. Hand gesturere cognition is one of the most natural ways to interact with an AR environment [18]. For example, in [19] camera pose was estimated by tracking the fingertips and virtual objects appear superimposed on the hand. However, one limitation of this approach was that the inspection of the object was hindered when the fingertips occluded each other [20]. A finger tracker was also presented in [21] that allowed gestural interaction, and a robust method was shown for detecting a hand on top of avision-based tracking pattern and rendering it over the attached virtual object [22].

Our approach for Tangible AR interaction is based on the Single-point (S-PI) Algorithm [10], a point-based intersection method. In order to support this we developed our own tracking library that enhances the existing *ARTag* [17] library. *ARTag* is a popular marker based tracking library that uses a 2D code to uniquely identify each marker being tracked. The use of an encoded ID in this way improves tracking accuracy compared to older methods such as the *ARToolkit* [23]. For gesture interaction, we use a physical paddle with an attached fiducial marker on it. The pose of the paddle pose marker is calculated in a one-time calibration process and through vertex based calculation of the 6DOF camera pose relative to it. Once the position of the camera is known relative to the paddle 3D graphics can be drawn on top of it. This allows the user to inspect virtual objects from different viewing angles and perform 3D object manipulation simply by moving the paddle around.

3.0 S-PI VISION-BASED TRACKING SYSTEM

In AR, issues in setting-up AR tracking systems mostly focus on robustness [24]. In the real-world, visual information can be very rich, noisy, and incomplete, due to changing lighting and illumination, clutter, dynamic backgrounds, and occlusion problems [25, 26]. Ideally, AR tracking systems should be user independent and robust against all these factors. The vision-based methods used for gesture interaction should also be effective with regards to speed and accuracy as well as computational cost. To achieve this, we combined the *ARTag* tracking system with the *S-PI* method. *S-PI* improves tracking robustness by representing markers as a set of vertices; the four corners of the marker and a fifth vertex defined at the center of the marker. This has the advantage that partial occlusion of the tracking marker can often be resolved correctly as shown in Figure 1. In the S-PI tracking system provides both 6DOF orientation and position tracking and handles some types of marker occlusion.

S-PI tracking system represents an AR tracking marker as a set of five vertices linked-up together. The vertices are found using the following algorithm;

STEP 1: Image processing identifies the ideal threshold value that can be used to binarize the input image as in Figure 1(a),

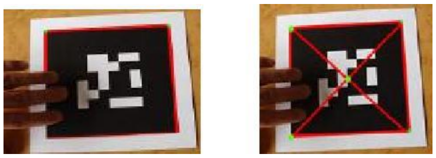
STEP 2: A contouring process is used to find square regions that can be fitted by extracting four boundary lines as illustrated in Figure 1(b),

STEP 3: This set of lines is used to produces four vertices V_1, V_2, V_3, and V_4, at each corner of the square as presented in Figure 1(c),

STEP 4: Once the four corner vertices are found, we connect the vertices crosswise to produce a new vertex, V_5 as shown in Figure 1(d). The new value of V_5 is stored in the vertex list.

STEP 5: Then, repeat *STEP 1* to *STEP 4* for all *n* markers, where *n* represents the number of visible markers.

(a) A seed marker (b) Edge detection process



(c) Edges with four vertices (d) Five vertices

Figure 1 *S-PI* marker-based tracking system

As shown in Figure 1, after these five vertices are successfully generated by the S-PI algorithm, partial occlusion of the tracking marker can often be resolved correctly. In this way, when the edge of the marker is covered by hand, the virtual content still appears. Using other marker-based tracking approaches, the AR system commonly fails when the marker is occluded by the user's hand. The occlusion problem also occurs when a user moves their hand above the ground and the 3D objects on the ground planar appear incorrect. With the S-PI tracking system, this can be fixed by using sets of vertices list stored recursively, since the number of tracking markers represented by vertices cannot be all covered up at the same time. Besides, this speeds up the marker detection process and produces a list of coordinates in external data. Once the marker is captured by the camera, the system will retrace the numbers of coordinates list that had been already stored into an array of integers when the text file is successful loaded. Therefore, the marker identification will be much faster since image processing is not involved.

4.0 OBJECT MANIPULATION METHOD

The main form of user interaction was using paddle-based gestures, in which a real object (the paddle) with an attached fiducial marker, allows the user to make gestures to interact with the virtual objects. One of the primary interaction tasks is virtual object manipulation which is initialized by an action executed on a previously selected point or virtual object in the AR environment. As well as the conventional moving operations, there are also other possible operations based on presence or absence of an object, and changing characteristics, motion, behavior, cues and feedback. However in our approach, we implemented conventional manipulation methods such as translation, rotational, copying, and picking and dropping gestures.

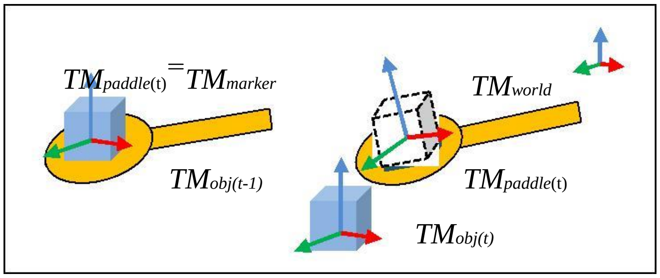


Figure 3 Paddle Interaction for object manipulation

The paddle (*paddle*) manipulation technique as shows in Figure 3 involved a picking and dropping gesture and was used for the 3D object selection and translation or rotation manipulations. The paddle V_5 had a coordinate set to a constant value and a radius RADIUS. A fiducial marker was attached to the top of the paddle. Selection occurs when the paddle and the position of the *paddle* are close together. In this case they move to the nearest point of the boundary, bound_box, the boundary of the desired object, as shown in Equation 1.

Selection: if **distance** (TM_{paddle},bound_box(M_{obj}) < RADIUS and no selected object *Equation 1*

This point is derived from the edge vertices of the boundary of the desired object. For the dropping (*Release*) condition, the 3D object was attached to the physical paddle and the verticesseeking the vertices drawn on the plane, as shown in Equation 2, where *TM* is the transformation matrix. During thetranslation and rotation tasks, a position property from TM_{paddle} is obtained from the frame most recently inserted into a localcoordinate transformation matrix of the 3D object (TM_{obj}).

Release : if **degree**(TM_{obj}) >εuntil object detect the markerand object selected *Equation 2*

Furthermore, the rotational matrix of the 3D object is established using a relative angle from a rotation matrix of *TMpaddle*in the previous frame to the rotation matrix in thecurrent frame. This relationship is shown in Equation 3, where T is the translation and R is the rotation matrices of *TM* respectively.

Translation/Rotation:$TM_{obj\,(t)}$ =
(R)$TMobj$(t-1)((R)$TMpaddle$(t-1)$^{-1}$ (R)$TMpaddle$(t))
(T)$TMpaddle$(t) *Equation 3*

In AR, interaction occurs when the user requests to perform a 3D object manipulation. The viewpoint calculation is needed to allow the virtual object to be overlaid on the physical tool marker. In this case, the ground marker array's transformations have to be offset. Thus, the paddle marker array transformation was multiplied with the inverse of the ground marker, where TM_{paddle}represents the transformation of the paddle marker, and TM_{world} the transformation of the ground marker. Finally, the world coordinates (X_w,Y_w,Z_w) as shown in Figure 4 have been defined when the identity matrix was identified.

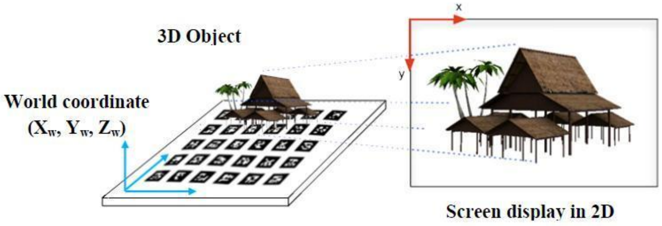


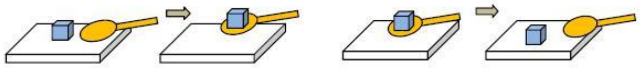
Figure 4 Coordinates system for object manipulation

In the manipulation method, the 3D information of the paddle and the virtual object was used to detect the collision between two objects when the user"s paddle was approaching to the virtual object. *Paddle* can control the virtual object efficiently when the collision is detected between their intersection vertices. *Paddle* can directly contact with a virtual object by simply detecting the collision between the paddle and the virtual object while the interaction is being processed. The possible interaction tasks for the object manipulation conducted by user are defined as follow:

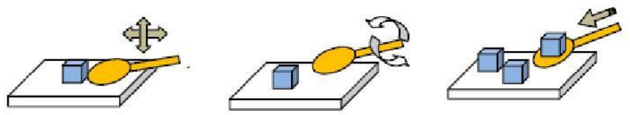
(a) *Selection:* Requiring user to pick the object
(b) *Release:* Requiring user to drop the object

(c) *Translation:* Requiring user to translate the object
(d) *Rotation:* Requiring user to rotate the object
(e) *Copying:* Requiring user to duplicate the object

The user can then picks objects from the menu pages and place them in the workspace using the paddle. The user performs a selection task in Figure 5(a) when they select a virtual object from the menu or workspace, and place it on the paddle.



(a) Selection (b) Release



(c) Translation (d) Rotation (e) Cloning

Figure 5 Paddle interaction task

Then they can perform a release operation, shown in Figure 5(b), when the attached object at the paddle location is placed in the workspace while detecting the nearest point intersection. A user can perform a translation task, shown in Figure 5(c) when they attach a virtual object on the workspace to the paddle so that it follows the paddle movement accordingly. Giving users 6DOF control of a virtual object may be useful in certain cases, but for our manipulation method, during object placement users are not allow to translate along the vertical up-axis. It is helpful to constrain the motion of the virtual object, however users are allowed to perform rotation about the up-axis (yaw). For rotational task shown in Figure 5(d), the user is allowed to perform 6DOF control using the paddle orientation (yaw, pitch, roll). The user can perform a copying task to duplicate the object that is attached to the paddle and place the copy on the workspace continuously at the desired position they want as shown in Figure 5(e).

When user the looks at different menu pages through the USB camera, they see the virtual objects on the pages, such as a tree, river or hut. The user will use the paddle to pick up the desired object and place it on the workspace. Using this interaction metaphor, instead of picking and dropping the objects, users are able to perform manipulation tasks such as translate, rotate and copying. The menu pages and the ground page are the real elements, each virtual object is registered to the real elements or physical objects and the user interacts with the virtual objects by manipulating the corresponding physical objects. The physical objects and interactions are as significant as the virtual objects and provide a very intuitive way to interact with the AR interface.

5.0 TEST APPLICATION

We developed a sample application to test our paddle-based interaction technique. This involved the manipulation of virtual objects in the *Ancient Malacca* project [27], which is a virtual heritage system that visualizes the ancient village of *Malacca* in 15th century [28]. In our AR interface, we invoke the paddle interaction metaphor mentioned in the previous section. In this test application, the user used a USB camera attached to their monitor and held a paddle to perform interaction tasks. The application was running at 30 frames per second with *640x480* pixel video images of the real world shown on the monitor with virtual graphics overlaid onto it.

The AR *Ancient Malacca* application, shown in Figure 6, allows users to manipulate 3D objects using paddle gestures and helps them to complete activities in the real world. When the paddle collides with any virtual object, object selection was executed (see Figure 6(a)). When the paddle is moved near to a virtual object, it can successfully pick up the virtual object once the intersection between the geometries of the object and marker is valid. The virtual object is then registered on the marker. The vertices drawn on the paddle marker will intersect with the points of objects that are overlaid on the ground marker. When the intersection between both vertices is successful, the transformation matrix will update the values by calling the function defined for the user‟s current selection (see Figure 6(b)). When collision detection is true, then there is a need to offset the workspace marker transformation as presented in Figure 6(c). When both markers are successfully detected, then the system checks for collision between the paddle marker and the desired 3D object and it will decide whether the 3D object is picked or released. Figure 6(d) shows the ground change from transparent rendering mode to solid rendering, indicating that the virtual object was successful placed on the ground. S-PI interaction will allow the user to freely move around the ground and view the virtual objects from almost any angle. The system needs to detect both of the paddle markers and the intersection between the near point and the desired virtual object allows the user to manipulate the picked object in real-time.

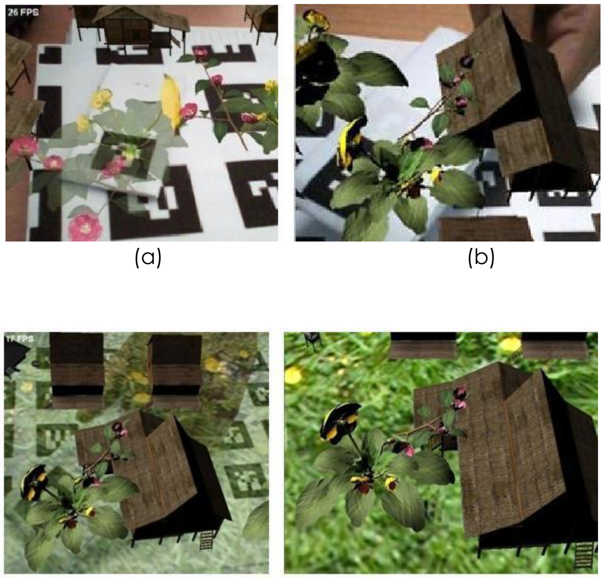

(a) (b)
(c) (d)

Figure 6 *AR ancient malacca* test application

6.0 CONCLUSION AND FUTURE WORKS

This paper discusses the S-PI vision-based tracking system for object handling in AR. In order to develop interaction techniques in AR, the robustness of the tracking system is explained and an intuitive paddle-based metaphor is explored. S-PI vision-based is for marker-based tracking proposes to enhance the existing *ARTag* tracking system. *ARTag* has improved the standard tracking with digital processing by storing the values in the array that represents the pattern IDs. Unlike *ARToolkit*, it uses digital encoding methods. All stored markers that are defined by IDs, known as user-defined markers, have improved the tracking accuracy.

S-PI improves tracking robustness even further by representing markers as a set of vertices. It converts the unique feature into five vertices, with the fifth vertex at the center of the marker. In the S-PI tracking system, we had assigned an image input with five vertices and performed a one-time calibration of a paddle marker to perform interaction. This can speed up the camera tracking and helps with the accuracy and robustness. The robustness in tracking is required to ensure both elements of real and virtual can be combined relative to the user‟s viewpoints without any delays and reduce the system lag. 3D object manipulations such as translation, rotation and cloning have been shown in S-PI using 6DOF position and orientation tracking. During the testing, we found that when they used the paddle-based system novices were typically more impressed and enthusiastic about the virtual object overlay in their physical world. The time required by users to learn this interface was shorter than expected, showing the application is easy to walkthrough.

In the future, we intend to explore multimodal interaction where a user can pick objects up and place them in the workspace using paddle and speech commands. Multimodal interfaces can be very intuitive because the strengths of voice input compliment the limitations of gesture interaction and vice versa. We will also explore gesture recognition techniques using other external controllers to replace the paddle such as the *kinect* or *leap motion* devices. *Leap motion* is a gesture tracking device that can detect our real palm and fingers [29]. Meanwhile *kinect* is a motion sensing input device enabling users to control and interact through a natural user interface using gestures and spoken commands [30]. These technologies would make the interface more natural. The goal of this on-going research will be to explore intuitive interaction metaphors with robust tracking in AR.

Acknowledgement

We would to express our appreciation to the staff and students at the Human Interface Technology Laboratory New Zealand (*HITLabNZ*) at University of Canterbury. This work was funded by Ministry of Science, Technology and Innovation through *eScieneFund Grant* 01-01-06-SF1188.

References

[1] Azuma, Y. Baillot, R. Behringer, S. Feiner, S. Julier and B. MacIntyre. 2001. Recent Advances in Augmented Reality. *IEEE Computer Graphics & Applications*. 21(6): 34-47.

[2] White, S., D. Feng and S. Feiner. 2009. Poster: Shake Menus: Activation and Placement Techniques for Prop-based 3D Graphical Menus. In *Proc. IEEE Symposium on 3D User Interfaces*. Lafayette, LA.14-15 March 2009. 129-130.

[3] Billinghurst, M., H. Kato and I. Poupyrev. 2001. The Magic-Book – Moving Seamlessly between Reality and Virtuality. In Healy, G., Shafer, S. and Wolff, L. *Physics Based Vision: Principles and Practice, COLOR*. Boston: Jones and Bartlett.

[4] Geiger, C., L. Oppermann and C. Reimann. 2003. 3D-Registered Interaction-Surfaces in Augmented Reality Space. In *Proceedings of 2nd IEEE International Augmented Reality Toolkit Workshop*. Tokyo, Japan. 7 October 2003. 5-13.

[5] Kiyokawa, K., H. Takemura and N.Yokoya. 1999. A Collaboration Support Technique by Integrating a Shared Virtual Reality and a Shared Augmented Reality. In *Proceedings of IEEE International Conference on Systems Man and Cybernetics*. Tokyo, Japan. 12-15 October 1999. 6: 48-53.

[6] Kato, H., M. Billinghurst, I. Poupyrev, N. Tetsutani and K. Tachibana. 2001. Tangible Augmented Reality for Human Computer Interaction. In *Proceedings of 17thNicograph (In Japanese)*. Nagoya, Japan. 39-44.

[7] Szalavari, Z. and M.Gervautz. 1997. The Personal Interaction Panel – A Two-Handed Interface for Augmented Reality. In *Proceedings of EUROGRAPHICS, Computer Graphics Forum*. 16(3): 335-346.

[8] Buchmann, S. Violich, M. Billinghurst and A. Cockburn. 2004. FingARtips: Gesture based Direct Manipulation in Augmented Reality. In *Proc. 2nd International Conference on Computer Graphics and Interactive Techniques in Australasia and South East Asia*. Singapore. 15-18June 2004.

212-221.

[9] Zhou, F., H. B. L. Duh and M. Billinghurst. 2008. Trends in Augmented Reality Tracking, Interaction and Display: A Review of Ten Years of ISMAR. In *Proc. IEEE ISMAR*. Cambridge, UK. 15-18 September 2008. 193-202.

[10] Ismail, A. W., M. S. Sunar and C. S. Yusof. 2011. Single-Point Interaction for 3D Object Handling in Augmented Reality. *Journal of Computing*. 3(1):61-65.

[11] Ishii, H. and B. Ullmer. 1997. Tangible Bits: Towards Seamless Interfaces between People, Bits and Atoms. In *ACM CHI 97*. Los Angeles, USA. 18-23 April 1997. 234-241.

[12] Kato, H., M. Billinghurst, I. Poupyrev, K. Imamoto and K. Tachibana. 2000. Virtual Object Manipulation on a Table-Top AR Environment. In *Proceedings of the International Symposium on Augmented Reality (ISAR 2000)*. Munich. 5-6 October 2000. 111–119.

[13] Irawati, S., S. Green, M. Billinghurst, A. Duenser and H. Ko. 2006. An Evaluation of an Augmented Reality Multimodal Interface using Speech and Paddle Gestures. 2006. In Pan, Z., Cheok, D. A. D., Haller, M., Lau, R., Saito, H., Liang, R. (eds.). *ICAT 2006*. Springer, Heidelberg.

[14] Henrysson, A., M. Billinghurst and M. Ollila. 2005. Virtual Object Manipulation using a Mobile Phone. In *Proceedings of the 2005 International Conference on Augmented Tele-Existence*. Christchurch, New Zealand. 5-8 Dec. 2005. 164-171.

[15] Lee, B. S. and J. C. Chun. 2010. Interactive Manipulation of Augmented Objects in Marker-Less AR using Vision-Based Hand Mouse. In *Inn Conference on Information Technology (ITNG)*. Las Vegas, NV. 12-14 April 2010. 398-403.

[16] Billinghurst, M., I. Poupyrev, H. Kato and R. May. 2000. Mixing Realities in Shared Space: An Augmented Reality Interface for Collaborative Computing. In *Proceedings of the IEEE International Conference on Multimedia and Expo (ICME2000)*. New York. 30July–2 August 2000. 3: 1641-1644.

[17] Fiala, M. 2005. ARTag, A Fiducial Marker System using Digital Techniques. In *Proc. CVPR 2005*. San Diego, CA. 20-25 June 2005. 2: 590-596.

[18] Lee, T. and T. Hollerer. 2008. Hybrid Feature Tracking and User Interaction for Markerless Augment Reality. In *IEEE International Conference on Virtual Reality'08*. Reno, NE. 8-12 March 2008. 145-152.

[19] Lee, T. and T. Hollerer. 2007. Hand AR: Markerless Inspection of Augmented Reality Objects using Fingertip Tracking. In *Proc. IEEE ISWC '07*. Boston, MA. 11-13 October 2007. 83-90.

[20] Lee, T. and T. Hollerer. 2007. Initializing Markerless Tracking using a Simple Hand Gesture. In *Proc. ACMIIEEE ISMAR '07*. Nara. 13-16 November 2007. 1-2.

[21] Dorfmüller-Ulhaas, K. and D. Schmalstieg. 2001. Finger Tracking for Interaction in Augmented Environments. In IEEE and ACM International Symposium on Augmented Reality *(ISAR 01)*. New York, NY. 29-30 October 2001. 55-64.

[22] Malik, S., C. McDonald and G. Roth. 2002. Hand Tracking for Interactive Pattern-Based Augmented Reality. In *Proceedings of the 1st International Symposium on Mixed and Augmented Reality* (*ISMAR 02)*. Darmstadt, Germany. 30 September–1 October 2002. 117-126.

[23] Philip L. ARToolKit. [Online] From: http://www.hitl.washington.edu/artoolkit.

[24] Park, J., B. Jiang and U. Neumann. 1999. Vision-based Pose Computation: Robust and Accurate Augmented Reality Tracking. In *2nd IEEE and ACMInternational Workshop on Augmented Reality* (*IWAR 99)*. San Francisco, California. 20-21 October 1999. 3-12.

[25] Klein, G. and T. Drummond. 2003. Robust Visual Tracking for Non-Instrumented Augmented Reality. In *Proceedings of the 2nd International Symposium on Mixed and Augmented Reality* (*ISMAR 03)*. Tokyo, Japan. 7-10 October 2003. 113-122.

[26] Bajura, M. and N. Ulrich. 1995. Dynamic Registration Correction in Video-Based Augmented Reality Systems. *IEEE Computer Graphics and Applications*. 15(5): 52-60.

[27] Sunar, M. S., A. M. Zin and T. M. Tengku Sembok. 2008. Improved View Frustum Culling Technique for Real-Time Virtual Heritage Application. *The International Journal of Virtual Reality*. 7(3): 43-48.

[28] Noh, Z., M. S. Sunar and Z. Pan. 2009. *Review on Augmented Reality for Virtual Heritage System*. Edutainment 2009 Springer-Verlag Berlin Heidelberg. Lecture Note in Computer Science LNCS 5670.

[29] Leap Motion, Inc. Leapmotion.[Online] From https://www.leapmotion.com/.

[30] Ren, Z., J. Yuan, J. Meng and Z. Zhang. 2013. Robust Part-Based Hand Gesture Recognition using Kinect Sensor. *IEEE Transactions on Multimedia*. 15(5): 1110-1120.