Interaction Techniques in Mobile Augmented Reality: State-of-the-art

Loubna Ahmed[1]
lobna.ahmed@cis.asu.edu.eg

Salma Hamdy[1]
s.hamdy@cis.asu.edu.eg

Doaa Hegazy[2]
doaa.hegazy@cis.asu.edu.eg

Taha El-Arif[1]
taha_elarif@cis.asu.edu.eg

***Abstract* — Augmented Reality (AR) has recently evolved and one of its evolutions is mobile AR. A key point of mobile AR is being reactive, which imposes real-time constraints. Hence, developing and improving interaction methods for AR have gained a wide interest in the past few years with the massive growth of mobile technology. In this paper, we introduce state-of-the-art interaction techniques for mobile AR, categorized into tangible and intangible as well as a comparative study between them. According to the results of the studied techniques, tangible techniques proved to be easier for users whereas intangible techniques are more engaging. As a result, intangible techniques need more research and enhancements.**

Keywords—Mobile Augmented Reality; Tangible Interaction; Intagible Interaction.

I. INTRODUCTION

Human-Computer Interaction (HCI) [1] is the field of science that is concerned with the design, study and planning of interaction between people and computers. One of the evolutions of HCI is *Augmented Reality (AR)* [2]. AR is the addition (augmentation) of virtual digital objects to the real environment around us [2, 3, 4, 5, 6]. These augmented virtual objects can be plane 2D and 3D in the form of text, audio or images [6].

More technically, AR involves a wide spectrum of technologies that project computer generated materials onto users' perceptions of the real world

Fig. 1. Augmented reality example [7].

(Fig. 1). Consequently, it serves to aid and enhance individual's knowledge and understanding of what is going on around them.

Since the early 1950's, AR has passed through many phases. Fig. 2 outlines the major evolutions in AR [8, 9, 10]. It started when Heilig [11] introduced his idea "The cinema of the Future" which he called "Sensorama". Later in the 1960's, Ivan Sutherland made the first AR system on a head mounted display [12]. Afterwards, Milgram and Kishino introduced the continuum that spans through real and virtual environments, showing in between the AR and the AV [13]. The first definition of AR was introduced by Ronald Azuma in 1997 [2]. Later, the first AR mobile outdoor game "ARQuake" was made by Bruce Thomas [14]. Since then, AR continues to evolve producing different technologies and products, one of which is the mobile AR.

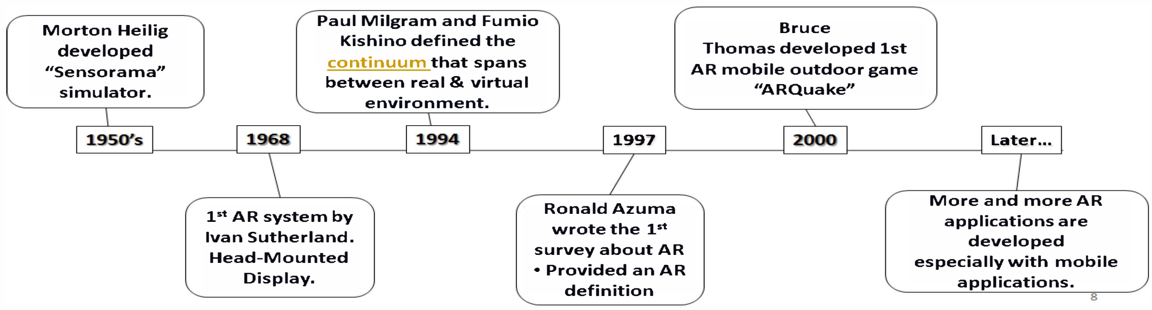


Fig. 2. History of AR

[1] Department of Computer Science, Ain-Shams University, Cairo, Egypt.
[2] Department of Scientific Computing, Ain-Shams University, Cairo, Egypt.

(a) Advertising and Marketing (b) Medical

(c) Education (d) Entertainment

Fig. 3. Examples of using AR in different applications.

AR has numerous real-life applications in many fields, including entertainment, architecture, military, medicine, marketing and education [10] (Fig. 3). It encompasses several research tracks [4] including graphics rendering, displaying and visualization, tracking, and interaction as depicted in Fig. 4.

There are many forms of display in AR like head-mounted display, eye glasses and mobile AR [8] [15] [16]. Graphical rendering refers to both hardware and software needed for creating the virtual content. In tracking, changes in the users' positions are tracked to affect and reflect on the graphics rendered. There have been many researches in object tracking for AR [17] [18] [19] [20].

Interaction techniques focus on allowing the users to interact with the emerging virtual object, and are considered the basis for having a successful AR system.

In this paper, a comparative study of interaction methods in mobile AR is conducted. This entails elaborate presentation, analysis, and evaluation of selected techniques. We basically categorize interaction methods as tangible or intangible; adopting a classification introduced by Bai H. et al. [21]. We analyze each selected technique in terms of its interaction methodology, pointing out its strengths and weaknesses.

The rest of the paper is organized as follows: in sections 2 and 3 the two interaction categories are explored, respectively. A summary of the discussed techniques is tabulated in section 4. Finally, conclusions are presented in section 5.

II. TANGIBLE INTERACTION TECHNIQUES

Tangible interaction techniques refer to the type of interaction where the user physically touches something, whether a mobile screen (touch-based) or a keypad (device-based) [21]. Most interaction concepts are 2D device-based interaction; but this type is difficult and restrictive because of some problems like small form factor [22].

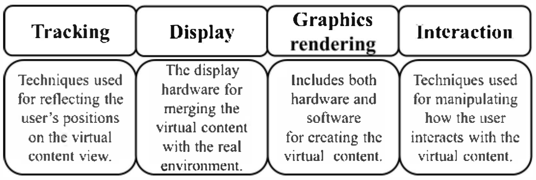


Fig. 4. AR research tracks.

One of the most popular publications in the literature was introduced in 2012, by Hürst and van Wezel [22]. They proposed two techniques; a touch-based (Fig. 5) and a device-based (Fig. 6).

In the touch-based method, selecting an object is achieved by clicking on it on the touch screen (Fig. 5.a). This selection evokes a context menu implemented in a pie menu style [23]. On the other hand, in the device-based, they get use of the data delivered for the accelerometer and compass to get the position and orientation of the device. A reticule is used to point to the desired object (Fig. 6.a). For an object selection, the reticule should remain for a period of time; this period is displayed in a progress bar (Fig. 6.b) for 1.25 seconds. For each technique three tasks were tested with respect to interaction time; object selection, menu selection and object translation.

A freeze view touch approach was introduced by Bai et al. [21]. The Freeze View interaction method was mainly introduced to overcome the problems arising with the previously proposed techniques due to shaking of the hands while holding the mobile device. In this method, the user holds the mobile device towards the intended object and after the object is correctly identified and tracked, a "Freeze" button appears on the screen.

(a) (b)

Fig. 5. Touch-based: (a) Object selection. (b) Entry in the context menu. [22]

(a) (b)

Fig. 6. Device-based: (a) Selection by pointing a reticule to the target. (b) The bar is filled while selection process. [22]

When the view is frozen, visual tracking stops and a menu pops listing the interaction types to be performed: scaling, translation, or rotation, as shown in Fig. 7.a. Moreover, world coordinates also appear (Fig. 7.b) facilitating the choice of the axis to operate on. After a successful selection, the user can unfreeze the view to restart the tracking.

This model is built on a tracking module and an interaction module and is implemented on top of a natural feature based tracking library that the authors have developed by porting BRISK algorithm [24] to Android using OpenCV4Android[3] and enhancing the performance using FastCV[4].

III. Intangible Interaction Techniques

Intangible interaction refers to the type of interaction that relies on the physical separation of the user from the device; like midair gesture, speech, etc. The interaction is mapped onto input parameters to control virtual content [21]. Intangible techniques can be classified into either marker based where markers are attached to the fingers or marker-less.

In addition to their previously mentioned intangible approach, authors in [22] proposed a marker-based (Fig. 8) interaction with both virtual and real objects using 2D input and a single camera (the camera of the device). Two experiments were made: one for virtual objects floating in midair with no relation to physical objects, and another when the objects have a connection to physical ones. Only 2D interactions are handled in this paper as 3D tracking with one camera is difficult and noisy, especially on mobile phones where the processing power is low and the camera is moving. In the first experiment, a sticker is attached to fingernail or the inner side of the finger. In both object selection and menu selection, a progress bar is filled to ensure that the object is fully selected. Whereas in object translation, the object is moved by pushing.

Fig. 7. (a) The Marking menu. (b) Coordinate buttons. [21]

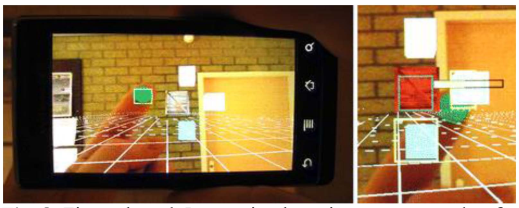


Fig. 8. Finger-based. Interaction by using a green marker for tracking. [22]

The second experiment explored the feasibility of having possible, natural and useful interactions suitable for an engaging game, where evaluations were based on a game board scenario. A green and a red marker are attached to the user's thumb and index finger, respectively. A bounding box is generated around both markers and the virtual object such that interaction is detected upon overlapping or touching between these bounding boxes. The authors implemented three interaction concepts in this experiment which are translation (Fig. 9), scaling (Fig. 10) and rotation (Fig. 11). The visualization strategy used in this implementation clearly indicates the interpretation of a particular interaction and the related status of the object (selected or not), hence alleviating problems such as lack of haptic feedback. Moreover, the tolerance distance between the bounding boxes of the markers and the object is increased allowing the object to be selective and responsive if the markers are within the area around it. Hence, the user does not have to touch the objects directly.

For the first experiment, the average time taken by all users to complete all tasks is computed as a measure of performance (Fig. 12). Moreover, a comparison between the interaction concepts based on performance, fun, and both is introduced in table 1, showing that the touch-based concept has the best performance while the finger-based is ranked to be the most fun.

Besides, for the second experiment, the average time it took the users to solve a task is computed.

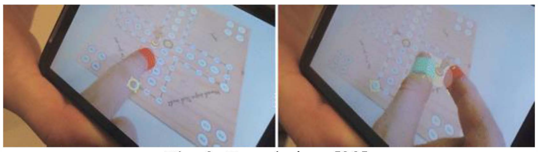
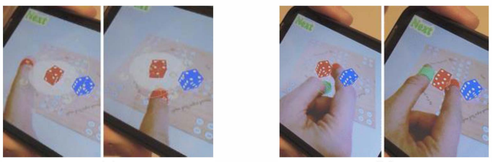


Fig. 9. Translation. [22]

Fig. 10. Scaling. [22]



(a) (b)

Fig. 11. Rotation. (a) Rotation using one finger. (b) Rotation using two fingers. [22]

[3] http://code.opencv.org/projects/opencv/wiki/OpenCV4Android/
[4] https://developer.qualcomm.com/mobile-development/mobile-technologies/computer-vision-fastcv/

TABLE 1. Times how often an interface was ranked first, second, and third with respect to performance versus fun and engagement versus both (T touch screen, D device, F finger) [22]

	T	D	F	T	D	F	T	D	F
Ranked 1st	11	6	1	1	4	13	8	3	7
Ranked 2nd	3	6	9	6	10	2	6	8	4
Ranked 3rd	4	6	8	11	4	3	4	7	7

The results shows that translation using one or two fingers worked well but the users preferred using two fingers as it is more natural. For scaling, within one interaction type, both midair and on board operations are almost the same but they differed with respect to accuracy. Finally, for rotation, using one finger and two fingers differed significantly in terms of average time while rotation in midair and on board differed slightly.

In 2008 [25], Seo et al. proposed a one-handed interaction technique where virtual objects are augmented on the palm of the user's free hand, while the other hand holds the mobile device. The author introduced both visual and tactile interactions. The visual interaction technique does not need any external visual markers or tags, as it interaction is done by motions of the hand such as opening and closing of the palm causing the virtual object to respond to the pose changes. On the other hand, the tactile interaction is obtained through receiving feedback from the virtual object. Fig. 13 shows the suggested visual interaction methodology.

Hand region segmentation takes place using a generalized statistical color model to classify skin color pixels on the captured scene. Then, an estimate of the palm poses is determined by finding the mean direction of the hand, then the palm direction, using the least square line fitting of its dominant features. Virtual objects are successfully rendered on the user's palm using a ratio of the palm's reference lengths. Visual interaction can be achieved by detecting the hand's motions based on the fingertip tracking method. Using the curvature based algorithm, candidate points of fingertips are detected and the candidate points which are relatively far away from the forearm can be selected as the fingertips.

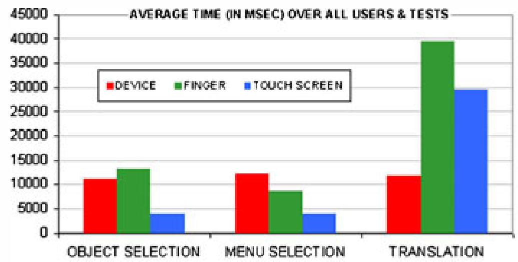


Fig. 12. Time it took to solve the tasks (in milliseconds, averaged over all users) [22]

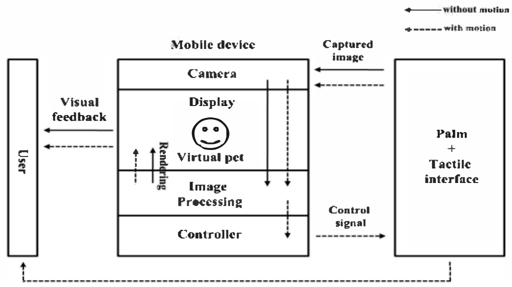


Fig. 13. Interaction flow. [25]

In addition, the authors estimate the exact location of fingertips by ellipse fitting of the selected points. In the demonstration, the authors used a UMPC (Sony, VGN-UX27LN). Fig. 14 shows an application for this model, where the user interacts with virtual pets.

Choi et al. proposed in 2011 a bare-hand-based AR interface for mobile phones [26]. This interface is similar to what Seo. et al. proposed in [25] but with more accurate estimate of all possible palm poses. They covered pitch of -60° to 60°, yaw of -45° to 60°, and roll of -180° to 180°, while the old technique had a limited scope of palm pose with only one direction of pitch and yaw rotation. Moreover, their proposed methodology is less time consuming.

A finger gesture-based intangible technique is introduced in [21]. The authors' Freeze View tangible technique was discussed in the previous section. Finger gesture interaction is based on midair gestures that are detected by the mobile device camera enabling the user to indirectly interact with the AR scene. It has been built using the same mobile platform, tracking system and UI framework used in the Freeze-view touch. Moreover, authors integrated their finger tracking system (C++ in Android NDK) into the previous system. After the object is successfully tracked, a hand button appears for segmenting the hand from the background if the hand area detected exceeded a certain threshold value. Then the prominent fingertips are marked. A big white circle on the fingertip is used to indicate it is not touching the object. Otherwise, the circle disappear and all interaction types become activated (Fig. 15).

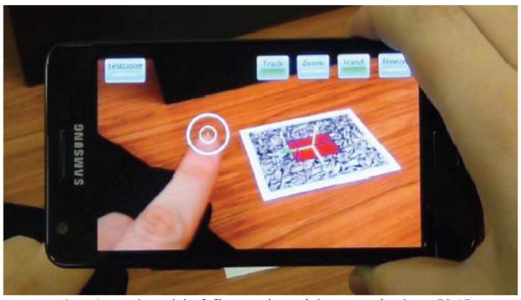


Fig. 15. The chief fingertip with two circles. [21]

TABLE 2: Completion Time and Operation Error of Each Task [21]
* Mean(*SD*/*SE*). Time values in second; Error values in pixel for translate and scale, degree for rotate.

	Time			Error		
Task	Free touch	Freeze touch	Finger gesture	Free touch	Freeze touch	Finger gesture
Translate	8.0(1.9/0.3)	6.3(0.8/0.1)	24.9(9.7/1.5)	8.2(4.5/0.7)	5.3(2.9/0.5)	14.8(11.0/1.7)
Rotate	11.0(4.1/0.6)	7.5(3.2/0.5)	22.0(12.1/1.9)	1.7(1.2/0.2)	0.8(0.8/0.1)	6.9(7.2/1.1)
Scale	7.1(2.9/0.5)	5.3(1.3/0.2)	21.4(11.3/1.8)	10.7(6.0/1.0)	12.9(7.7/1.2)	20.1(7.2/1.1)

In this technique, skin color detector working in HSV color space are applied followed by the distance transform. The fingertips are then identified based on the curvature-based contour point sampling and elliptical fitting method used in Handy AR [27]. This implementation works under stable light condition and with an assumption that the hand is placed in front of the user's face, and during the gesture interaction, the finger is always visible. The authors calculated the completion time taken for each task and the error as shown in table 2. Analyzing these data, they found that the gesture-based concepts took more than twice the time of the freeze-view and the free-view touch.

Gao [28] presented two 3D gesture-based interaction methods for mobile AR. Both methods support 3D interaction by using depth camera to obtain 3D coordinates of the users' fingertips along with the virtual objects. Moreover, a touch-based method was introduced in this thesis and compared to the natural gesture-based methods.

The first technique, Gesture-Based Interaction using Client-Server Framework, consists of a PC desktop as a server, a Kinect depth camera, a tablet as a client, and a marker. The Kinect camera sends the RGB and depth images to the PC server which combines them to get the depth information of the images. The fingertip 3D coordinates are detected by this calibrated RGB and depth output and sent to the mobile client for mobile AR 3D interaction via a wireless connection (Fig. 16). This framework implements three atomic gestures: translation, rotation and scaling of objects.

The second technique, Gesture-based interaction using a tablet, consists of a tablet and a Primesense depth sensor connected using a USB cable (Fig. 17). It just combines the PC server system and the mobile client system together into a tablet. RGB and depth images are acquired from the depth sensor and combined together to support a pixel to pixel mapping for the system. The fingertip coordinates are calculated based on the combined images and a full 3D manipulation is supported for the users by using the received fingertip coordinates.

This method implements pinch-like gestures for selecting objects which simulates the real life for grabbing real objects. This is achieved by comparing the midpoint between the two finger tips; thumb and index with the center of the object.

The authors presented their results comparing the gesture-based interaction with the touch-based; stating that on average, 2D-touch based interaction performed better and faster than 3D gesture-based interaction, as Fig. 18 indicates. In addition, according to the subjective questionnaire; it was found that, in general, gesture-based is more fun and engaging than touch-based. However, touch-based has proven to be much easier and less stressful.

To sum up, the results were not as the authors expected as most users preferred touch-based to gesture-based. They analyzed the reasons as follows; there is no physical feedback to the user from the objects on selection, the user's fingers are always covered by the virtual objects overlaid on the video frames. Another reason is the problem of losing the fingers position when the user's free hand gets out of the video frame accidentally. Finally, a problem arises when using two fingers for translation or scaling; as the distance between the two fingers can be greater or smaller than the virtual object size, also while rotating the object one finger can cover the other in certain angles. All these problems need to be considered and handled in the future.

Baldauf et al. [29] developed a visual marker less fingertip detection engine to detect 3D objects along with several use cases for using this engine in interaction. One of these use cases is the Virtual Object Interaction, where the virtual objects can be manipulated through selection by pointing, grabbing or dragging and dropping by pinch like gesture using the thumb and index fingers.

The fingertip detection methodology works as follows. First, skin-specific values identified by Peer et al. [30], were used as thresholding values for segmenting the hand (Fig. 19.a). Next, contour of the hand is marked, then the palm is identified through the largest possible inner circle found by a distance transformation. Upper finger parts are identified by iterating over the hand's contour points, consequent points with a distance above a certain threshold from the inner circle's center (Fig. 19.b). Finally, by finding the local distance maxima of the respective contour segments the fingertips appear on the video stream (Fig. 19.c).

Fig. 16. (a) System framework (b) System setup. [28]

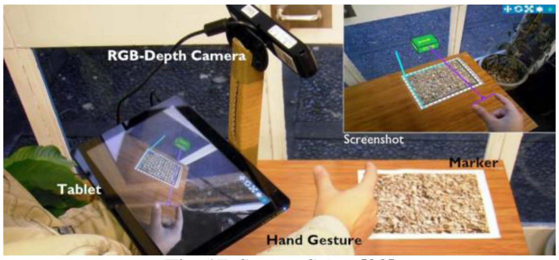


Fig. 17. System Setup [28]

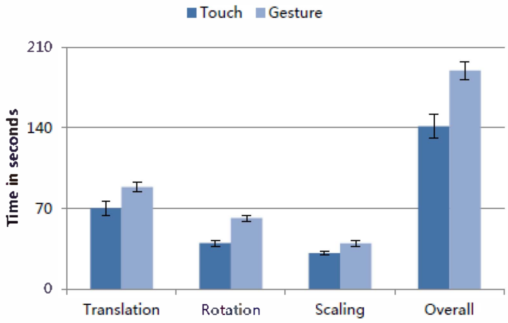
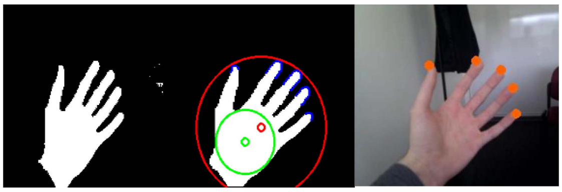


Fig. 18. Wilcoxon Signed-Rank Test. [28]

In 2013, Chun and Höllerer [31] introduced a methodology for marker-less real-time handling interaction of users with the virtual objects appearing on the mobile phone screen. In this paper, three gestures are handled; translation, scaling and adjusting the transparency of the object as the user's hand come closer or moves away from it. Those gestures were specifically chosen to optimize the learn ability as they are similar to touch screen gestures. The methodology is illustrated in Fig. 20. First, the hand is detected after background subtraction rather than detecting it from raw input. They used RGB color space to avoid the cost of color space conversion, then they performed RGB normalizing. Authors assumed the hand will be around the AR marker (virtual object) hence, a search window of a size three times the marker area is chosen to save computational costs. The search window is divided to 4x4 grids (Fig. 21.d) to track hand movement within the area by computing the percentage of skin-color pixels $P(skin_n)$. Two different interaction modes are implemented: one is discrete event detection (Fig. 21.a and 21.b) where a threshold is set manually for each cell, such that if $P(skin_n)$ is greater than the threshold value, $grid_n$ is hit. The second is continuous value adjustment (Fig. 21.c) where they recorded how much the hand occludes that grid. In this case, each cell is not compared to a threshold, but the amount of hand occlusion in each cell, $P(skin_n)$ is used to change a value dynamically, for instance opacity.



(a) (b) (c)

Fig. 19. Fingertip detection steps. (a) Hand segmentation through thresholding. (b) Contour finding and fingertip analysis (c) Fingertip visualization in the live video. [27]

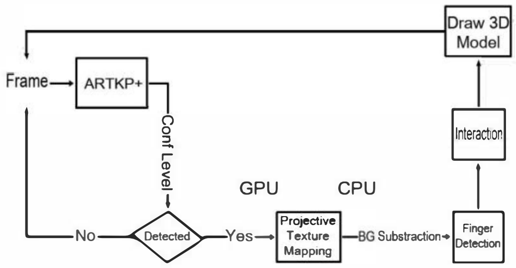


Fig. 20. Flow Diagram of the application. [31]

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

Fig. 21. (a) and (b) show translation and scaling interaction (c) shows continuous value adjustment, applied to box transparency. In this experiment, the user attempts to match a target opacity value (d) shows (as debug information) the 4*4 grid board used to track the hand. [31]

IV. Comparison between Interaction Techniques

In this section, we present a comparison between all the introduced techniques in terms of hardware, software used, and highlight the pros, cons, and the results of each technique. It is also based on the tangible versus intangible paradigm as stated before.

Ref.	Technique	Hardware	Software	Remarks
Tangible methods				
[21] - 2012	Freeze-View Touch	Samsung Galaxy S2	• AR Tracking: C++ with Android NDK • Touch Interaction and UI layout: Java with Android SDK • Natural Feature based tracking library: BRISK algorithm imported to OpenCV4Andr-oid, FASTCV to optimize performance.	Compared to gesture-based [21], it has been found to be better in usability, not stressful, and easier to learn.
[22] – 2012	Device-based interaction with midair virtual objects	MotorolaDroid/Milestone phone	• OS: Android version 2.1	Device based interaction achieved the best results in the shortest time for translation tasks and was highly ranked by the users for this purpose.
[22] – 2012	Touch-based interaction with midair virtual objects	MotorolaDroid/Milestone phone	• OS: Android version 2.1	Compared to both device-based [22] and finger-based [22], touch screen is the fastest approach for selection tasks.
Intangible methods				
[22] – 2012	Interaction with objects related to the real world (game board)	HTC Desire HD smartphone.	• OS: Android version 2.2.3 • Natural feature tracking framework: Qualcomm Augmented Reality (QCAR) SDK	Interaction with objects attached to real world recorded to have better performance than midair ones [22]. Moreover, it proved to be engaging for gaming applications.
[22] – 2012	Finger-based interaction with midair virtual objects	MotorolaDroid/Milestone phone	• OS: Android version 2.1	Compared to touch-based [22] and device-based [22], finger-based is not the best in performance, yet it is the most engaging as it resembles the real world.
[21] – 2012	Finger Gesture Interaction	Samsung Galaxy S2	• AR Tracking: C++ with Android NDK • Finger Tracking: C++ with Android NDK • Touch Interaction and UI layout: Java with Android SDK • Natural Feature based tracking library: BRISK algorithm imported to OpenCV4Andr-oid, FASTCV to optimize performance.	Compared to freeze-view touch-based [21], it has been found to better in entertainment and engaging applications. Moreover, authors stated that more work is required for better accuracy in the gesture-based interaction.
[25] - 2008	One-Handed Interaction with Virtual Objects (Visual Interaction)	UMPC (Sony, VGN-UX2LN)	*Not mentioned*	The proposed palm pose estimation method resulted in a relatively accurate estimation.

[26] – 2011	Bare-Hand-Based Augmented Reality Interface	Samsung Galaxy S2	• OS: Android version 2.3	Through various optimizations such as reducing floating point operations and parallel processing, the proposed interface works more than three times faster than the original one [25]. Also, it has a recognition rate of 83% on average.
[31] – 2013	Real-time Hand Interaction	Nokia smart phone	• OS: Maemo 5 • Prototype built on ARToolKitPlus [32]	Authors addressed the problem of potential camera motion during gesture detection successfully using projective texture mapping and background subtraction. Acceptable correctness rates were obtained for simple scenes while maintaining good frame rates. They also provided a thorough performance analysis of user studies through collecting task performance data and user impressions.
[28] – 2013	Gesture-Based Interaction using Client-Server	Microsoft Kinect, Desktop PC, Android Smart phone and wireless connection	OS: Client: Android 4.1 Server: windows 7 • AR Tracking: OPIRA library • Obtaining images from depth sensor: OpenCV and OpenNI • Object rendering and graphics: OpenGL	Due to the error in the Kinect (less than 0.5 cm when placed around 75 cm from the marker plane), the manipulation accuracy of the system is within 1 cm under the physical WCS. This indicates that the prototype is not suitable for high-precision mobile AR applications. However, it meets the basic requirements of general interaction tasks.
[28] – 2013	Gesture-Based Interaction using a Tablet	Primesense depth sensor and Tablet	• OS: windows 7 • AR Tracking: OPIRA library • Obtaining images from depth sensor: OpenCV and OpenNI object • Rendering and graphics: OpenGL	The framework differs from gesture-based interaction using client-server [28] in that the PC server system and the mobile client device are combined in the tablet. In addition, as opposed to the Kinect camera, the PrimeSense sensor used provides a short detection range so that users can operate naturally.
[29] – 2011	Marker-less Visual Fingertip Detection	HTC Desire phone: 1 GHz processor, 576 MB RAM, and a 5 megapixel camera.	• OS: Android version 2.1 • OpenCV • Android's Native Development Kit. • SWIG wrappers.	

V. Conclusion and Future Work

In this paper, we presented an elaborate survey on selected interaction techniques categorized into tangible and intangible. According to the subjective questionnaires in most of the studies presented, tangible techniques proved to be easier to use and non-stressful compared to the intangible techniques. On the other hand, intangible techniques proved to be more engaging and fun to the users, and closer to the real world interaction than tangible ones.

Moreover, gesture-based had some issues that need to be taken into consideration. In case of object selection, it is a hard task to detect when the selection started and to feedback the user that the object has been successfully selected [22] [28]. Also, lighting conditions in handling fingertip detection both indoors and outdoors affects the performance considerably. The problem of the user's hand always covered by the virtual objects overlaid in the video frames [28] is also a matter of concern.

From our extensive reviews, we also concluded that interaction techniques have no performance evaluation benchmark that enables comparing the techniques according to a standard criteria. Accordingly, most published comparisons mainly depend on subjective questions like if the technique is engaging, mentally stressful, physically stressful, easy to use, etc. Perhaps the most solid evaluation criteria is with respect to the response time, which is application dependant, and varies heavily from one task to another.

References

[1] A. Dix, J. Finlay, G. D. Abowd and R. Beale. "Human Computer Interaction,", 1996.

[2] R. T. Azuma. "A survey of augmented reality," in Presence-Teleoperators and Virtual Environments, 6(4): pp. 355–385, August 1997.

[3] H. Kaufmann. "Collaborative Augmented Reality in Education". Keynote Speech at Imagina Conference, 2003.

[4] F. Zhou, H. B.-L. Duh, and M. Billinghurst, "Trends in augmented reality tracking, interaction and display: A Review of Ten Years of ISMAR " in Proc. IEEE ISMAR, Cambridge, UK, 2008, pp. 193-202.

[5] T. H¨ollerer and S. Feiner. "Mobile augmented reality," in *Telegeoinformatics: Location-Based Computing and Services. Taylor and Francis Books Ltd., London, UK*, 2004.

[6] S.Yuen, G. Yaoyuneyong, E. Johnson. "Augmented reality: An overview and five directions for AR in education," in Journal of Educational Technology Development and Exchange, 4:1, 2011, pp. 119-140.

[7] C. Kirner, C. Cerqueira and T. Kirner. "Using augmented reality artifacts in education and cognitive rehabilitation" in Virtual Reality in Psychological, Medical and Pedagogical Applications, 2012, pp. 248 – 270.

[8] D. van Krevelen and R. Poelman. "A survey of augmented reality technologies, applications and limitations". IJVR, 9(2): 2010, pp.1 –20.

[9] C. Arth, L. Gruber, R. Grasset, T. Langlotz, A. Mulloni, D. Schmalstieg and D. Wagner. "The History of Mobile Augmented Reality". September 2015.

[10] J. Carmigniani and B. Furht. "Augmented reality: an overview" in Handbook of Augmented Reality, USA, 2011.

[11] M. Heilig, "Sensorama simulator", U.S. Patent, August 28, 1962.

[12] I. E. Sutherland. "A head-mounted three dimensional display," in Proc. of the December 9-11, 1968, Fall Joint Computer Conference, Part I, AFIPS '68, New York, USA, pp. 757-764, 1968.

[13] P. Milgram and A.F. Kishino, "Taxonomy of mixed reality visual displays" in IEICE Transactions on Information Systems, E77-D(12), 1994, pp. 1321–1329.

[14] B. Thomas, B. Close, J. Donoghue, J. Squires, P. De Bondi, M. Morris and W. Piekarski, "ARQuake: An outdoor/indoor augmented reality first person application," in Proc. 4th Int'l Symp. Wearable Computers (ISWC 2000), 2000, pp. 139-146.

[15] H.-K. Wu, S. W.-Y. Lee, H.-Y Chang and J.-C. Liang. "Current status, opportunities and challenges of augmented reality in education" in Computers & Education, 62, 2013, pp. 41–49.

[16] P. Bonnet, P. Ducher and A. Kubiak. "A Brief Introduction to Augmented Reality," in Advances in Embedded Interactive Systems, vol. 2, Issue 4. ISSN, 2014, pp: 2198-9494.

[17] D. Koller, G. Klinker, E. Rose, D. Breen, R. Whitaker, and M. Tuceryan. "Real-time vision-based camera tracking for augmented reality applications," in ACM Symp. on Virtual Reality, Software and Technology (VRST-97), 1997.

[18] A.I. Comport, E. Marchand, M. Pressigout, and F. Chaumette. "Realtime markerless tracking for augmented reality: the virtual visual servoing framework," in IEEE Transactions on Visualization and Computer Graphics, 12(4): pp: 615–628, July 2006.

[19] U. Neumann and S. You, "Natural Feature Tracking for Augmented Reality," IEEE Trans. Multimedia, vol. 1, no. 1, pp. 53-64, March 1999.

[20] D. Wagner, G. Reitmayr, A. Mulloni, T. Drummond, and D. Schmalstieg, "Real-time detection and tracking for augmented reality on mobile phones," in IEEE Trans. Visual Comput. Graph, vol. 16, no. 3, 2010, pp. 355–368.

[21] H. Bai, G. A. Lee, and M. Billinghurst, "Freeze view touch and finger gesture based interaction methods for handheld augmented reality interfaces," in Proc. of the 27th Conference on Image and Vision Computing New Zealand, 2012, pp. 126–131.

[22] W. Hürst and C. Wezel, "Gesture-based interaction via finger tracking for mobile augmented reality," in Multimedia Tools and Applications, vol. 62, no. 1, pp. 233– 258, Jan. 2012.

[23] J. Callahan, D. Hopkins, M. Weiser, and B. Shneiderman. "An empirical comparison of pie vs. linear menus," in CHI'88, 1988, pp. 95–100.

[24] S. Leutenegger, M. Chli, and R. Siegwart. „BRISK: Binary Robust Invariant Scalable Keypoints," in Proc. of the 13th IEEE

International Conference on Computer Vision, ICCV '11, Barcelona, Spain, pp. 2548–2555, November 2011.

[25] B.-K. Seo, J. Choi, J.-H. Han, H. Park, and J. Park, "One-handed interaction with augmented virtual objects on mobile devices," in Proc. of 7th International Conference on Virtual Reality Continuum and Its Applications in Industry, 2008.

[26] J. Choi,H. Park, J.Park, and J.-I1. Park., "Bare-hand-based augmented reality interface on mobile phone," in Mixed and Augmented Reality (ISMAR), 2011 10th IEEE International Symp., 2011, pp. 275–276.

[27] T. Lee and T. Hollerer, "Handy AR: markerless inspection of augmented reality objects using fingertip tracking," in Proc. of 11th IEEE International Symp. on Wearable Computers, 2007, pp. 83-90.

[28] L. Gao. "Natural gesture based interaction for handheld augmented reality," M.S. thesis, Dept. Comput.Sci., Canterbury Univ., 2013.

[29] M. Baldauf, S. Zambanini, P. Fr¨ohlich, and P. Reichl. "Markerless visual fingertip detection for natural mobile device interaction," in Proc. of the 13th International Conference on Human Computer Interaction with Mobile Devices and Services, MobileHCI '11, Stockholm, Sweden, pp. 539–544, September 2011.

[30] J. Kovac, P. Peer and F. Solina, "Human skin colour clustering for face detection," in in Proc. of Int. Conf. on Computer as a Tool, 2003, pp. 144–148.

[31] W. H. Chun and T. H¨ollerer. "Real-time hand interaction for augmented reality on mobile phones". 2013.

[32] D. Wagner and D. Schmalstieg. "ARToolKitPlus for pose tracking on mobile devices," in Proc. of 12th Computer Vision Winter Workshop (CVWW'07), pp. 139-146, 2007.