FPGA-based Power Efficient Interactive Augmented Reality Learning Applications for Children

Abdul-Rahman Algharbi, Farida Hamid
School of Communication and Electronics
October University for Modern Sciences and Arts
Giza, Egypt
e-mail: aralgharbi@gmail.com
e-mail: faridaaddala@gmail.com

Noha Younis
National Center for Radiation Research and
Technology, Atomic Energy Authority
Cairo, Egypt
e-mail: nyounis@msa.eun.eg

***Abstract*— Most human-computer interaction systems, specifically, Augmented Reality, are designed based on general purpose processors. Consequently, their power consumption is considerably high, as systems work at Gigahertz rates. In this paper, power efficient interactive augmented reality learning applications for children are designed and implemented. Interaction is performed by hand gestures and markers. The power consumption of the proposed system is reduced by developing and implementing the recognition and tracking processes on a Field Programmable Gate Array platform to exploit its parallelism feature. This enables the system to work, portably, at lower operating frequencies, without violating the required real-time performance. The most suitable five hand gestures for 3 to 10-year-olds were determined, then the implemented system was tested by 100 children. Implementation results revealed that only 25 MHz are sufficient for the applications to run in real-time at 30 fps, with a recognition rate of 93.2% on average. This significantly reduces the overall power consumption of the proposed system, comparing with other systems.**

Keywords; Augmented Reality (AR); hand gestures; AR markers; FPGA; Human Computer Interaction (HCI)

I. INTRODUCTION

Augmented Reality (AR) is a direct or indirect combination between real and virtual worlds in real-time, by using computer generated graphics, sounds, or videos. AR is used for various applications, such as medicine, entertainment, and marketing as a kind of Human Computer Interaction (HCI). Augmented reality is also used in educational fields, as it can improve the pedagogical methodology by enhancing children's concentration [1]- [5]. However, such systems do not offer a direct virtual interaction between the students' body and the virtual object, which is necessary to keep them attracted and to enrich their imagination. Most interactive AR systems use markers that are represented by either barcode [6], or certain objects, attached to data gloves [7]. Interaction can, also, be achieved by means of bare hand gestures [8][9], or by hand gestures together with markers [10] [11]. The main challenge in such interactive AR systems is detecting, recognizing and tracking markers and/or gestures in real-time, with high accuracy. Object detection is developed, in some systems, using hardware devices such as gloves, supported with sensors, to digitize the detected hand-gesture/marker [7] [12] [13]. Though such systems are robust and provide high recognition rates, their power consumption is high. Also, gloves are not user friendly as they limit the user's movement. This is in addition to the inflexible size of gloves that cannot fit all human hands' sizes, specifically, children's. Other systems use depth cameras, such as Kinect, which provides HCI systems with the hand skeleton and depth such that the remaining detecting steps become easier [14] [15]. However, it is expensive and power consuming, and is only compatible with Windows operating systems. Moreover, there is another part of the power, consumed by the software program that is developed to execute the remaining recognition and tracking operations. Such power consumption is considerably high as the program is executed at gigahertz rates to achieve a real-time performance. In [10], a mobile AR system was proposed for teaching and learning, using a low power low processing device, to control simple hand gestures that in turn control presentation slides. Another effective method to reduce the power consumption is implementing the high computationally complicated functions on a Field Programable Gate Array (FPGA) platform, where a wearable backpacked computer and tracking gloves are used [16]. However, the system does not provide a high recognition rate, since it is built in Handel-C that was later converted to a Hardware Descriptive Language. Also, the user should wear gloves and a heavy computer while using it, which isn't suitable for children. Another FPGA-based hand gesture HCI is proposed in [17], where an Artificial Neural Network (ANN) was implemented on FPGA to recognize hand gestures. Again, a data glove was used to detect the proposed gestures.

To reduce the power consumption, an FPGA-based interactive AR system is proposed, where the high computationally complicated hand-gesture/marker detection, recognition, and tracking processes are designed and implemented on an FPGA. The power consumption is, then, reduced by operating the implemented system at lower frequencies. This is achieved by gaining the benefit of the parallelism feature from the FPGA to execute more than one function in parallel, in a way that the required real-time performance is not affected. As a result, the proposed low power consuming design can be used portably anywhere that suits children, especially in Kindergarten and primary levels.

The implemented applications have been applied on normal and autistic children to find how far it improves their concentration.

The remainder of the paper is organized as follows: Section II demonstrates an overview of the proposed hand-gesture/marker AR system. The software and hardware

ISBN: 978-1-61208-538-8

architectures are explained in Sections III, and IV, respectively. In Section V, the Hardware-Software interface is described, whereas the practical implementation results are discussed in Section VI. Finally, the conclusion and future work are presented in Section VII.

II. SYSTEM OVERVIEW

The proposed system supports four different learning applications for children, as shown in Figure 1. The first and second applications use hand gestures to control and interact with virtual objects, displayed on the Personal Computer (PC) monitor. The third and fourth applications use markers, instead. In the first application -called "Animal Homeland"- the child grabs a virtual animal using his/her hand and places it on its home land. In the second application -called “Planting”- the child learns about the planting phases by planting virtually, using hand gestures. In the third application -called “Machinery”- the child uses different mono-color markers to assemble machinery objects, such as an airplane. The fourth application, “Atom System”, helps children to realize the main particles of the atom (i.e., protons, neutrons and electrons). In this application, children move multi-colored markers that represent the atom particles, to virtually display their rotations around the atom.

To minimize the computational complexity, caused by recognizing and classifying different objects of various applications, the system was designed such that the user selects an application, first. This is done by means of the Graphical User Interface (GUI), depicted in Figure 1, which has been developed using Unity, the game engine [18]. Then, the application number is sent from the PC to the FPGA, where classification and tracking are done for only the gestures/markers of the selected application. Figure 2 illustrates the system setup, where the system input is the successive frames, captured by a CMOS webcam, which is connected to the PC. The scene includes a white background, on top of which, a hand gesture/marker moves. 'A' and 'B' represent the areas of the detected object at zero and maximum heights, respectively, which are used to get the object depth, as explained later. The output of the FPGA, which represents the recognized gesture/marker and its 3D position, is sent to the PC. Then, the monitor displays the gesture/marker, after mixing it with a virtual object that is determined, based on the received data from the FPGA. Figure 3 shows the block diagram of the proposed system that consists of Software (SW) and FPGA – based architectures. Both architectures are discussed in the following sections.

I. SOFTWARE-BASED ARCHITECTURE

Captured frames are automatically stored in the PC RAM, since the camera is, directly, connected to the PC to display the user's hand on its monitor, while interacting with virtual objects. Consequently, to minimize the size of the FPGA utilized RAMs, it is more efficient to carry out the first step of detection, which is color segmentation, in the PC. For hand-gesture-based applications, a skin color filter is applied, that range is selected using MATLAB. For Marker-based applications, a color filter is applied instead, based on the colors of the markers. Afterwards, a binary conversion of the segmented frame is executed. Both color segmentation and binary conversion are developed using Unity. Figures 4 and 5 show the skin color segmentation of the applied five hand gestures and the markers, respectively. The five gestures were selected after visiting several schools; the children were given small figures, to hold by hand, then they were asked to perform several gestures. Accordingly, the most feasible five

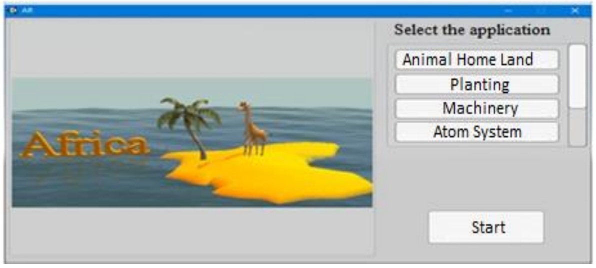


Figure 1. GUI of the proposed multi-application interactive AR system.

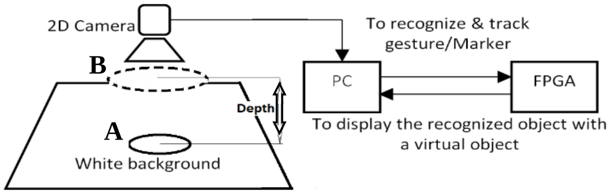

2D Camera
To recognize & track gesture/Marker
B
PC
FPGA
Depth
A
White background
To display the recognized object with a virtual object

Figure 2. System setup.

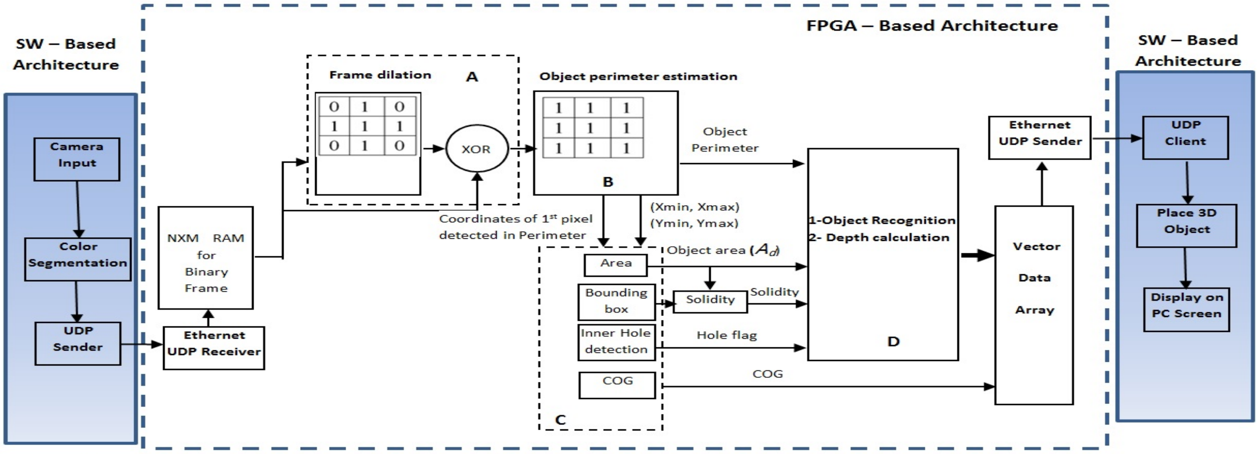
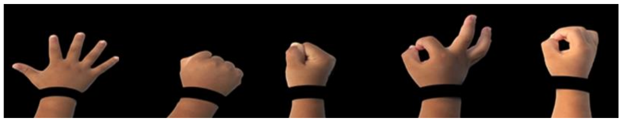
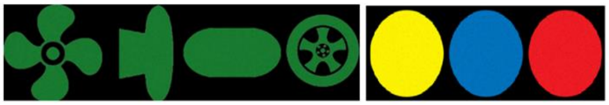


Figure 3. SW / FPGA-based architectures of the proposed multi-application interactive AR system.



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

Figure 4. Skin color segmentation of gestures, used in the 1st and 2nd applications: a) Open hand, b) Horizontal fist, c) Vertical fist, d) Three fingers, e) With hole.



(a) (b)

Figure 5. Color segmentation of the AR Markers, used in (a) 3rd application, and (b) 4th application.

gestures -shown in Figure 4- for 3-to-10-year-olds were determined. The gestures, shown in Figure 4 are performed without any rotation or twisting during interaction. To isolate the detected hand from the rest of the arm, a green ribbon is worn on the child's wrist, if he/she is not wearing long sleeves. The binary frame is, then, sent to the FPGA for recognizing and tracking. N×M represents the camera resolution. For marker-based applications, the colors of the markers are also sent to the FPGA to ease classification, as children may use markers of other applications that have appropriate features as those used in the selected application.

II. FPGA-BASED ARCHITECTURE

Since the application is first selected, a limited number of hand gestures or markers should be recognized. Hence, shape-based features are selected for classification, to reduce the computational complexity. The extracted features are the object perimeter, area, bounding box, solidity, Center of Gravity (CoG), and the object depth.

A. *Object perimeter extraction*

The received binary frame is stored in an N×M RAM, implemented in the FPGA. The perimeters of all segments in the frame are, first, extracted by dilating each shape in the original binary frame with the structuring element, shown in Block A of Figure 3. The dilated frame is then XORed with the original binary frame, to give a new N×M array that includes only perimeters [19]. Then, the structuring element, shown in block B of Figure 3, is used to go through all shapes in the frame, and the number of binary-one-pixels for each connected segment is accumulated, to get the perimeter length. The largest estimated perimeter represents the marker/gesture perimeter. Other segments are considered noise, and therefore deleted, except the ones enclosed inside the object perimeter, as they are used to detect the inner hole of the gesture, if any, as explained later. Also, the minimum and maximum X and Y coordinates of the object perimeter -(X_{min}, X_{max}) (Y_{min}, Y_{max})- are saved to extract other features.

B. *Parellel extraction of featurs*

To gain the benefit of the parallelism feature from the FPGA, the calculation of the area, bounding box, CoG, as well as the inner hole detection of the gestures, shown in Figure 4 (d) and (e), are processed in parallel. This reduces the power consumption and processing time, considerably.

To increase the recognition rate of the gestures of Figure 4 (d) and (e), their inner hole is detected after extracting the object perimeter. This is done by comparing the perimeter of shapes that are bordered by (X_{min}, X_{max}) and (Y_{min}, Y_{max}). The segment represents an inner hole of the gesture if the ratio between the perimeters of the candidate hole and the object is greater than 0.01%. Otherwise, it is considered noise. Figure 6 shows the inner hole, enclosed inside the hand gesture of Figure 4 (d).

The Object area is calculated by (1), where max x_y and min x_y represent the maximum and minimum X coordinates, respectively, which exist on the object edge at a specific y.

In parallel, the coordinates of the CoG, (X_C, Y_C), are calculated by (2) [20]. Also, the width, *W*, and length, *L*, of the bounding box are calculated using (3) and (4), respectively.

Another feature, to be extracted, is the solidity. It is used since it is not affected by the distance between the object and the camera, as it represents the ratio between the object area and the convex area. In the implemented system, the *bounding box* is used instead of the convex area to reduce the computational complexity. The solidity, S, is then estimated by (5).

$$A_d = \sum_{y=Y\min}^{Y\max} (\max x_y - \min x_y) \qquad (1)$$

$$X_C = \frac{X_{max}-X_{min}}{2}, Y_C = \frac{Y_{max}-Y_{min}}{2} \qquad (2)$$

$$W = X_{max} - X_{min} \qquad (3)$$

$$L = Y_{max} - Y_{min} \qquad (4)$$

$$S = \frac{A_d}{(WxL)} \qquad (5)$$

C. *Object recognition and 3D pose localizations*

The perimeter, area, and solidity of the five gestures were pre-calculated for 140 different hands to define the upper and lower limits of their thresholds. In Figure 7, the five gestures from "Open" to "With hole" represent the five gestures of Figure 4, from Figure 4(a) to Figure 4(e). Figure 7 shows that the (open) and (3 Fingers), gestures can be recognized based on any of the three features, whereas the (Vertical) gesture can be recognized based on the area and the perimeter features. Though the (With hole) and (Horizontal) gestures are overlapped in all features, the inner hole detection is used to distinguish them. Similarly, for marker recognition, the colors and the normalized values of the area and solidity of all utilized markers were calculated, and listed in Table I.

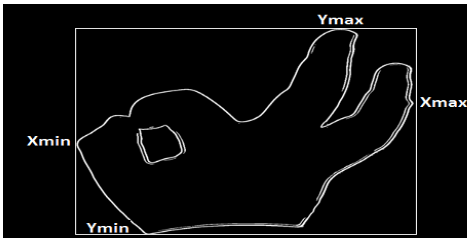


Figure 6. The edges of the three fingers-gesture and its inner hole.

For 3D pose localization, more than a single 2D cameras, or depth sensors are usually used [14] [15]. This is mandatory if the utilized gestures/markers rotate or twist during interaction. In the proposed system, neither rotation nor twist is required. Thus, the object depth can be calculated using only one 2D camera, based on the change of the object area at different heights, with respect to the camera, as shown in Figure 2. For any different user, both A and B areas shown in Figure 2, are calibrated once, for all gestures, such that after object recognition they are used to estimate its depth, Z, by applying (6). A is the biggest possible hand area in pixels, whereas B is the smallest possible hand area in pixels. H, represents the height at which the camera is fixed, with respect to the background, and 'Δ' is the difference between the areas A and B that is independent of A_d. However, 'Δ' varies if the distance between the camera and the background is changed.

$$Z = (A_d - A) * H/\Delta \tag{6}$$

III. HARDWARE - SOFTWARE INTERFACE

To interface the FPGA with the PC, Ethernet and User Datagram Protocol (UDP) are used. UDP is selected because the it transfers data directly without dividing them into chunks, and it does not depend on a certain Operating System (OS). The N×M binary frame is sent to the FPGA via Ethernet, followed by an m-bit data that represent the application number, and the marker color. For the proposed system, 4 bits are adequate to represent m. On the other side, a data vector is sent from the FPGA to the PC, via Ethernet, after executing the recognition and tracking processes. This vector carries the required information of the recognized object and its 3D position. The first Q bits of the data vector represent the recognized gesture/marker, where Q is determined according to the maximum number of different objects used in one application. In the proposed system, the maximum number of objects in one application is five; hence, Q is equal to three bits. The remaining parts of the data vector are 24 bits that represent the 3D position of the detected object, (X_C, Y_C, Z), where each coordinate is represented with eight bits. Wireshark analyzer monitored the packets sent via Ethernet to the FPGA. Also, it monitors the packets received from the FPGA, and provides error check methods, which in turn, diagnose and correct errors that can result from the VHDL code. A C# program was developed using Unity to capture the received data and check if they passed by all network layers and the OS accepted it. After that, the user's hand is combined with the 3D virtual model that is selected and located on the screen according to the received data vector.

Figure 8 illustrates the functions, designed and developed by Unity, where the data base of the application, selected by the GUI, is passed to the UDP receiver that also receives the data vector from the FPGA. The received data packet is then stored temporarily in the PC’s RAM, such that each new packet deletes the old one. Afterwards, the data are analyzed according to the selected application and the received 3D coordinates, where the center of the detected object is combined with that of the virtual 3D object in the Integration Unit. The continuous response of the virtual object is controlled by the Controller Unit, based on the data vector.

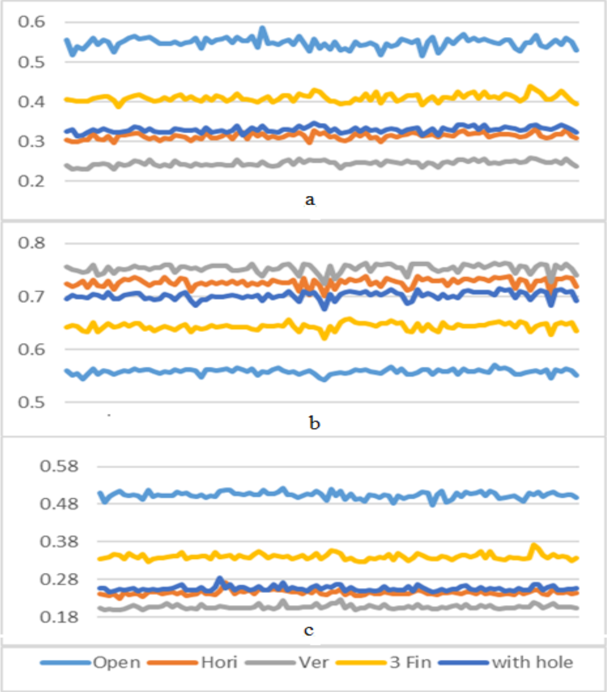


Figure 7. Normalized values of (a) Area, (b) Solidity, and (c) Perimeter of 140 different hands for the applied five hand gestures, in the system.

TABLE I. NORMALIZED VALUES OF THE MARKERS FEATURES

Marker	Color	Area	Solidity
Motor	Green	0.30421875	0.65847
Tale	Green	0.225143229	0.84043
Wheel	Green	0.317083333	0.5021
Wing	Green	0.331158854	0.91021
Proton	Blue	0.323385417	0.66774
Electron	Red	0.36265625	0.74883
Neutron	Yellow	0.300859375	0.62123

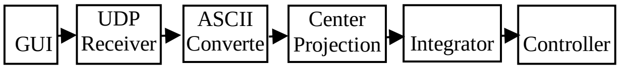


Figure 8. Software algorithm, executed by Unity game developing

IV. IMPLEMENTATION RESULTS AND DISCUSSION

Several visits were paid to nurseries and schools to evaluate the practical accuracy of the implemented system, and to see how well kids interact with it. The practical testing setup consisted of a webcam with a 640×480 resolution. The camera is fixed at height of 70 cm above a 50 cm × 25 cm white background. The software–based architecture runs on a 2.2 GHz processor.

Table II lists the recognition rate of each hand gesture after testing it on 100 children, and that the average recognition rate is 93.2%. Also, it can be noticed that the lowest recognition rate was of the “3 Fingers” gesture. This is because the gesture was slightly difficult for kids who are less than 5 years old to steadily perform, especially the first time they practice the applications. On the other side, the accuracy of the utilized markers approaches 100% because their shapes, features, and colors are constant, unlike gestures.

TABLE II. RECOGNITION RATE OF THE UTILIZED HAND GESTURES

Gesture	Open	Hori	Ver	3Fin	With Hole
Recognition (%)	96	93	92	90	95

Table III lists the FPGA utilized resources of the proposed system that was estimated by Xilinx Integrated Software Environment (ISE) tool, after implementing it on the low power Spartan3, S1600e-4fg320 device. The FPGA chip is mounted on a MicroBlaze development Kit-Spartan3E-1600E that features an Ethernet interface to the PC. Table III indicates that the implemented system uses less than 5% of the FPGA resources, which allows further modifications to improve the recognition rate and increase the number of gestures and markers. In addition, the implementation reports revealed that the maximum operating frequency at which the FPGA-implemented system can work is 102.817 MHz. However, the implemented system successfully works in real-time at only 25 MHz, when the camera works at 30 fps.

The power consumption of the implemented system was estimated, using XPower analyzer, provided by Xilinx. It is found that the FPGA–based implemented part consumes 5.65 mW at 25 MHz. Using higher frame rates is not demanded, since hand speed varies from medium to slow rates. Hence, it is more efficient to use lower frame rates, such as 15 fps, to reduce the operating frequency required for a real-time performance. This, in turn, optimizes power consumption much better.

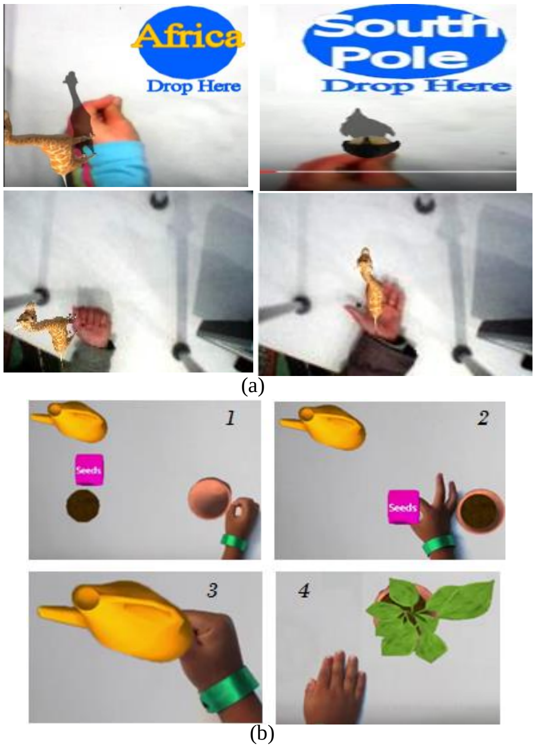
In addition, practical testing showed that normal children were very excited while practicing the applications without finding any difficulty performing the selected hand gestures. On the other side, autistic children were more interested in marker based applications, but they were hardly obeying their teachers when told to do the selected hand gestures. However, after an adequate effort from the teachers, autistic children could, so far, interact with the applications by gestures.

Figure 9 (a) shows kids interacting with animals by dragging them to their homeland, using different hand gestures. Figure 9 (b) illustrates the planting steps starting with holding the planting pot by the "With hole" gesture, putting a seed inside the pot using the "three figures" gesture, dropping water by the "vertical fist" gesture, and finally growing the plant by the "open hand" gesture. In Figure 10 (a) the child holds and drags different shaped green markers to assemble an airplane that flies at the end. In Figure 10 (b) the child uses three different colored markers that represent the main particles of the protons, neutrons and electrons. The child moves the colored markers and locates them on the circles in the middle, that represent the atom structure. Afterword, the electrons -red balls- start rotating around the atom in the fourth step.

To determine whether the learning outcomes of the applications have been successfully received by the students and how far such interactive applications enhance their concentration, some matching quizzes were given to 25 normal children ranging from 4 to 8-year-olds. The quizzes results differed according to the child's age and the application, however, the results were 75% on average, after the first time of practicing the applications.

TABLE III. FPGA UTILIZED RESOURCES OF THE IMPLEMENTED SYSTEM

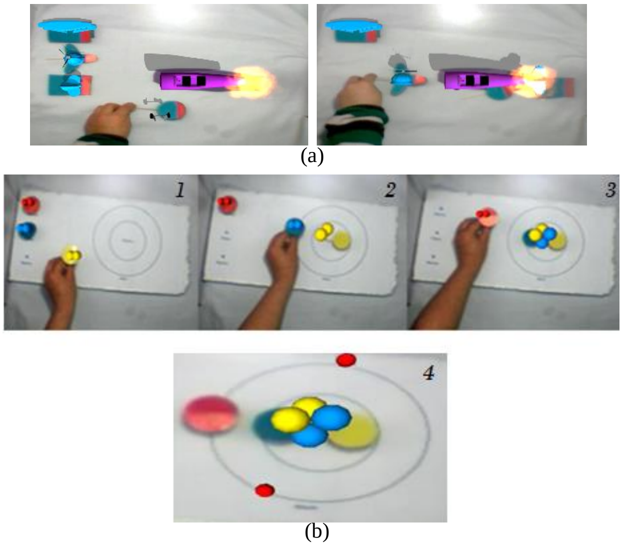
Logic Utilization	Used	Available	Utilization
Number of Slice Flip Flops	412	29,504	1.3 %
Number of 4 input LUTs	927	29,504	3.1 %
Number of occupied Slices	553	14,752	3.7 %
Number of RAMB16s	1	36	2.77 %
Number of BUFGMUXs	1	24	4.1 %



(a)

(b)

Figure 9. Kids practice (a) the first and (b) the second applications.



(a)

(b)

Figure 10. Kids practice (a) the third and (b) the fourth applications.

TABLE IV. COMPARISON BETWEEN THE PROPOSED SYSTEM AND OTHER INTERACTIVE AR AND HCI SYSTEMS

System	G	M	R (%)	FR	F_{OP} (MHz)	P_{HW} (mw)	Technology Used
[9]	2	None	90	U	1600	> 12000 (for Kinect [21])	Intel Core i5 processor & Kinect camera.
[10]	4	3 colored ribbons	93.3	U	U	U	Low Processor (U).
[15]	1	9	U	25	2800	> 12000 (for Kinect and Video Camera)	Intel Core i5 processor, Kinect & Video cameras.
[16]	None	2 color balls	U	25	U	6000	Celoxica RC200 reconfigurable computer, Virtex II, Head worn PAL resolution camera, and Tracked glove.
Proposed system	5	7	93.2	30	25 at 30 fps	(5.65: for FPGA-based design)	General purpose processor & FPGA (UR < 5 %).

Table IV summarizes a comparison between the proposed system and some other hand gesture and marker based HCI systems, where G and M are the number of used gestures and markers, respectively, R is the recognition rate, FR is the frame rate, F_{OP} is the applied operating frequency for real-time recognition and tracking, P_{HW} is the amount of power, consumed by the hardware equipment, UR is the FPGA Utilized Resources, and U stands for undefined. From Table IV it is noticed that for real-time performance, the operating frequency and power consumption of the proposed system are lower than those of the other systems. Another important point is that, the markers, used in the proposed system, are handmade, unlike the complicated barcode markers, used in [7], [12] and [16]. Using such simple and cheap markers makes children feel included as they participate in creating the 3D objects, and it makes the system more reliable, as its reusable resources are affordable.

V. CONCLUSION AND FUTURE WORK

A low power interactive AR learning system for children was proposed. The high computationally complicated recognition and tracking functions were implemented on an FPGA to minimize the operating frequency without violating real-time performance. This helps using the proposed applications anywhere that fits children. Implementation and testing results show that the recognition rate of the implemented system is 93.2% on average. Comparing the proposed implemented system to some other systems, it is found that the proposed system is more efficient in terms of power consumption, and reliability. For future work, a general form is currently being finalized by the authors, to get the optimal FPGA operating frequency, at which the system can work in real-time, as a function of the frame rate. This will enable working at rates even lower than 25 Mhz. Also, more hand gestures and markers can be recognized by adding additional features as only 5% of the FPGA resources are utilized. In addition, a custom Printed Circuit Board (PCB) will be manufactured that contains only the necessary components for the implemented system to minimize size and cost.

ACKNOWLEDGMENT

Authors thank Prof. Khayri Abdel Hamid for his sincere support, and for breaking many barriers the authors faced, during working in this manuscript.

REFERENCES

[1] M. Mafner, M. Schoning, M. Antczak, A. Demenko, and K. Hameyer, “Methods for computation and visualization of Magnetic Flux Lines in 3D,” *IEEE Trans. Magn.*, vol. 46, no. 8, pp. 3349 – 3352, Aug. 2010.A.

[2] S. Matsutomo, T. Miyauchi, S. Noguchi, and H. Yamashita, “Real-Time visualization system of magnetic field utilization Augmented Reality Technology for education,” *IEEE Trans. Magn.*, vol. 48, no. 2, pp. 531-534, Feb. 2012.

[3] M. Sugimoto, “A mobile mixed-reality environment for children’s storytelling using a handheld projector and a robot,” *IEEE Trans. Learn. Technol.*, vol. 4, no. 3, pp. 249 – 260, Sep. 2011.

[4] D. B. Chin, I. M. Dohmen, and D. L. Schwartz, “Young children can learn Scientific Reasoning with teachable agents,” *IEEE Trans. Learn. Technol.*, vol. 6, no. 3, pp. 248 – 257, Sep. 2013.

[5] L. Escobedo, M. Tentori, E. Quintana, J. Favela, and D. Garcia-Rosas, “Using Augmented Reality to help children with Autism stay focused,” *IEEE Pervasive Computing*, vol.13, no.1, pp. 38-46, Mar. 2014.

[6] B. Zhen, A. F. Blackwell, and G. Coulouris, “Using Augmented Reality to elicit pretend play for children with Autism,” *IEEE Trans. Vis. Comput. Graph.*, vol. 21, no. 5, pp. 598-610, May 2015.

[7] S. Prasad, P. Kumar, and K. P. Sinha, “A wireless dynamic gesture user interface for HCI using hand data glove,” *in International Conference on Contemporary Computing (IC3)*, Noida, India, pp. 62-67, Aug. 2014.

[8] L. Figueiredo, V. Teichrieb, and R. Anios, “Bare hand natural interaction with augmented objects, " *In IEEE International symposium on Mixed and Augmented Reality (ISMAR)*, Adelaide, Australia, pp. 1-6, Oct. 2013.

[9] J. M. Teixeira, B. Reis, S. Macedo, and J. Kelner, “Open/Closed hand classification using Kinect data,” *in 14th Symposium on Virtual and Augmented Reality (SVR)*, Adelaide, Australia, pp. 18-25, May 2012.

[10] S. Prasad, S. K. Peddoju, and D. Ghosh, “Mobile Augmented Reality based interactive teaching & learning system with low computation approach,” *in IEEE Symposium on Computational Intelligence in Control and Automation (CICA)*, Singapore, pp. 97-103, Apr. 2013.

[11] S. Jinwook, K. Minje, Y. Yoonsik, S. Jonghoon, and H. Tack-Don, "Interactive features based Augmented Reality authoring tool," *in IEEE International Conference on Consumer Electronics (ICCE)*, Las Vegas, USA, pp. 47-50, Jan. 2014.

[12] L. Dipietro, A. M. Sabatini, and P. Dario, “A survey of Glove-Based Systems and their applications*,” in IEEE Trans. on Systems, Man, and Cybernetics*, vol. 38, no. 4, pp. 461-482, Jul. 2008.

[13] Z. Huang, W. Li, C. Peylo, and P. Hui, "CloudRidAR: A cloud-based architecture for mobile Augmented Reality," *in Telekom Innovation Laboratories*, Berlin, Germany, 2012.

[14] R. Lo, et al., "Augmented Reality system based on 3D camera self-gesture sensing," *in IEEE International Symposium on Technology and Society (ISTAS)*, Toronto, Canada, pp. 20-31, Jun. 2013.

[15] C. Stritzke and R. Radkowski, "Interactive hand gesture-based assembly for Augmented Reality applications," *in Advances in*

Computer-Human Interactions, Paderborn, Germany, PP. 303-308, 2012.

[16] W. Piekarski, R. Smith, G. Wigley, B. Thomas, and D. Kearney, "Mobile hand tracking using FPGAs for low powered augmented reality," *in 8th International Symposium on Wearable Computers (ISWC'04)*, Mawson Lakes, Australia, pp. 190-191, Oct.2004.

[17] S. Oniga, A. Tisan, D. Mic, A. Buchman, and A. Vida-Ratiu, "Hand postures recognition system using Artificial Neural Networks implemented in FPGA," *in 30th International Spring Seminar on Electronics Technology*, Cluj-Napoca, Romania pp. 507-512, May 2007.

[18] Unity, <https://unity3d.com/ > 2016.11.16

[19] R. E. Woods and R. C. Gonzalez, *Digital Image Processing*, 3rd ed., Miami, Pearson International Edition, 2007.

[20] M. Kolsch, M. Turk, "Fast 2D hand tracking with Flocks of Features and Multi-Cue integration," *in Conference on Computer Vision and Pattern Recognition Workshop (CVPRW '04.)*, Washington, USA, pp. 158-158, Jun. 2004.

[21] <http://www.datasheet-pdf.com/datasheet-html/K/i/n/Kinectsensor_MicrosoftCorporation.pdf.html> > 2016.11.

ISBN: 978-1-61208-538-8