Object Recognition in Assembly Assisted by Augmented Reality System

Conference Paper · November 2015

DOI: 10.1109/IntelliSys.2015.7361172

2 authors:

Anderson Nishihara
University of São Paulo

Jun Okamoto Jr.
University of São Paulo

SAI Intelligent Systems Conference 2015
November 10-11, 2015 | London, UK

Object Recognition in Assembly Assisted by Augmented Reality System

Anderson Nishihara*, Jun Okamoto Jr.†
Department of Mechatronics and Mechanical Systems Engineering
Polytechnic School of The University of São Paulo
São Paulo, Brazil
Email:*anderson.nishihara@usp.br, †jokamoto@usp.br

***Abstract*—Assembly processes for simple toys or complex machines require instructions to be executed. Traditionally, these instructions are written in the form of paper or digital manuals. These manuals contain descriptive text, photos or diagrams to guide the assembly sequence from the beginning to the final state. To change this paradigm, an augmented reality system is proposed to guide users in assembly tasks. The system recognizes each piece to be assembled through image processing techniques and guides the piece placement with graphic signs. Still, the system checks if the pieces are properly assembled and alerts the user when the assembly has been finished. In the field of assembly assisted by augmented reality systems, many works use some kind of customized device, such as head mounted displays (HMD). Additionally, markers are commonly used to track camera position and to identify assembly parts. These two features restrict the spread of the technology whence, in this work, customized devices and markers to track and identify parts are not used and all the processing is executed on embedded software in an off-the-shelf device without the need of communication with other computers to offload image processing. In this work we present the current results of the proposed system's object recognition process. The implemented object recognition subsystem is part of a system called A^{3}R (Assembly Assisted by Augmented Reality).**

Keywords*—augmented reality; assembly process; image processing; object recognition

I. INTRODUCTION

Several elements need to be addressed in assembly tasks, such as the piece's orientation and the assembly sequence. As a way to assist assembly tasks, several authors have proposed augmented reality systems [1], [2], [3]. They propose that images augmented with instructions may be an alternative to the traditional paper manual and it would improve the assembly time. In these researches, implementations of augmented reality systems were compared to computer assisted instructions, paper manuals and tutorial by an expert. Although augmented reality (AR) system did not improved assembly time compared to a tutorial by an expert [2], it was evidenced that assembly time can be improved in comparison to computer assisted instructions [1], [3] and paper manuals [1], [2], [3]. Yet, when augmented reality system was used the number of errors in the assembly process were decreased [1], [3].

This research consists in development of an augmented reality system to assist the assembly process in real time as a replacement to conventional assembly manuals. The proposed system behavior is like an interactive instruction manual where virtual models are overlaid on a video of the real scene and the user is guided until the final assembled state is reached. To prove the concepts described here, a tablet is used between the user and the workspace in a way that a set of randomly placed assembly pieces can be viewed on the display. From the video feed, the pieces to be assembled are identified through image processing techniques and graphic signs show to the user where the piece should be assembled. Lastly, the system checks the assembly state and alerts the user when the final state is reached.

The system has the following characteristics:

- Fiducial markers are not used: objects are detected by image processing and named from a database;
- the solution is known beforehand: the system is not for finding a solution to the assembly problem;
- the system does not teach the assembly process to the users, its purpose is to be a guide through the assembly process;
- the display is placed between the user and the assembly pieces and it is fixed during all the process;
- the tablet implementation runs an embedded software, only resources from the tablet is used and no network connection is required.

In this paper we present the results of the object recognition process from the proposed AR system for assembly guidance. The object recognition algorithm was implemented on a tablet to identify pieces of a planar puzzle. Object recognition is part of a system called A^{3}R (Assembly Assisted by Augmented Reality).

II. AUGMENTED REALITY IN ASSEMBLY TASKS

One of the first efforts to build an AR tool was shown by [4] to guide the assembly of cable harnesses. Information derived from engineering design of parts and processes come to the factory floor in form of templates, assembly guides, cable lists and location markings. Expenses and delays on manufacturing can be minimized if changes on engineering design of parts and processes can be quickly mirrored to templates, assembly guides, cable lists and location markings. To address this issue, [4] demonstrated an AR system composed of a wearable computer and a head mounted display (HMD). Later, researchers demonstrated applications to guide printer maintenance [5] and car door-lock assembly [6].

Specifically on AR applications in assembly processes, we can analyze researches in this field under 4 main aspects which enables the AR system implementation: display, type of tracking system, user interaction and platform.

A. Display

Different kinds of displays are used on AR systems. There are two different approaches to show information to the user: optical see-through and video see-through. In optical see-through approach a half-transparent mirror is placed in front of the user's eyes. This way the world can be seen though the half-transparent mirror and information is reflected on these half-transparent mirrors, thereby combining the real world and virtual information. Optical see-through displays do not present parallax effect, however, the combination of virtual elements by half-transparent mirror can reduce the brightness and contrast of virtual images. In video see-through approach the users see the world captured by cameras and information is superimposed on the digitized video. Video see-through approaches are cheaper and easier to be implemented. Besides, it is easier to add or remove elements from the display, as the world is already digitized. Still, as the camera and user's eyes have different field of views, video see-through approaches suffer from parallax effect. As for the placement of the display, some devices are placed on the user's head and the display is positioned in front of the eyes (HMD - head mounted displays), projectors can be used to display information on real objects (spatial displays) and some displays are placed on a hand's reach (hand-held displays).

On literature, optical see-through ([3], [7]) and video see-through HMD ([8], [1], [9], [10], [11], [12], [13], [7]) are commonly used on AR systems as they enable the use of both user's hands on assembly processes. However, HMD available on the market are bulky, expensive or requires connection to a PC computer.

Spatial displays use projectors to directly display information on real objects. This is the most integrated technology to the environment and several users can interact simultaneously. [14], [15], [16] implement projector-based AR systems. However, only [15] uses the projector as a spatial display to guide the user to a solution of a 3D puzzle. Although spatial systems enable more immersion on the task than other alternatives, it is required a controlled environment illumination and preparation of workspace parameters to correctly superimpose information on assembly pieces.

Smartphones, tablets and PDA represent hand-held displays. [17] has shown a system composed of a smartphone and PC computer to guide assembly process of 3D puzzle. While the PC computer made all the image processing, the smartphone was used to display images augmented with assembly information.

B. Tracking

When augmented reality is used only to show information to the user and it is not required to superimpose information aligned to real world, this is called augmented reality without context. [3], [8], [18], [15] demonstrated AR systems without context. In this kind of AR, images, animations and texts are shown to the user on a display similarly to an electronic manual. The main difference between an electronic manual and AR without context is the faster access to assembly information without change of attention, as the information is on the same display as the real world seen by the user.

Augmented reality with context requires the knowledge of camera and object position. In some cases, camera and object position are not known beforehand, therefore, methods to extract this information are needed. These methods are often called tracking. The main challenge to tracking methods is the depth information which is lost when a camera captures the world. Some AR systems use a combination of camera and another sensor to acquire the depth information. Magnetic sensors [1], infrared sensor [19] and multiple cameras [20] have been used to extract the depth data.

Specifically on AR uses on assembly processes, the main approach to tracking is the application of computer vision techniques on images taken by a singular camera. Two different approaches using computer vision techniques are found on the literature: tracking with fiducial markers and marker-less tracking. Fiducial markers make the tracking and recognition more robust, however, it requires the preparation of the workspace. [11], [9], [10], [21], [17], [12], [22], [23], [16] demonstrate AR systems which use fiducial markers tracking approach.

There are mainly two kinds of marker-less tracking: frame-to-frame and tracking by detection. Frame-to-frame methods estimate the current position by using information from previous frames. Tracking by detection estimates position by matching extracted features from images to features extracted from a known object model. [24] has shown an AR system that uses tracking by combining frame-to-frame and tracking by detection methods. While frame-to-frame tracking is achieved by an implementation of a particle filter algorithm, tracking by detection is made by matching features from several 2D synthetic views of the 3D geometric features generated by moving a virtual camera along a sphere.

C. User interaction

The analyzed AR systems provided different kinds of user interface. Some researches use mouse and PC keyboard [9], [10], [16] or smartphone keyboard [17] to enable interaction while other researches propose interface with selectable panels by using a 3D stick tool [21], hand and gesture recognition methods [12] and haptic interfaces [20].

Two kinds of tools are used to interact in AR environments: computer vision tools and haptic tools. [10] demonstrated a system controlled by hand gestures. The detection of hands is made by color segmentation and virtual menus open by holding the hands on an activation area. [21] presents a stick tool detected by image segmentation to interact with virtual panels. Following [21] work, [25], [26] improved the segmentation technique to recognize hands and gestures, respectively.

[10], [27] presented voice control. The voice control is made by a connection from the AR system with a stand-alone voice command system which makes the voice recognition of simple English commands like next, previous, next phase and previous phase.

Yet, [20] demonstrated an AR system with a haptic device where users wear a device to manipulate virtual objects as it were real objects, which enables a more integrated and natural way to interact with the virtual elements.

D. Platform

Several AR systems are implemented by a combination of PC computer and HMD/monitor [3], [8], [1], [10], [27], [12], [24], [14], [11], [15], [28], [16], [13]. The PC computer versatility, connection diversity which enables several devices connected at the same time, wide availability on market and processing power are the main reasons for choosing PC computers as platform for AR systems. Still, there are restrictions on connections to HMD, as it requires a connection to a PC computer.

One main requirement on AR systems are near real time response. Manipulation of 3D virtual objects, tracking and object recognition require suitable processing power and memory. Due to this requirement, few approaches have been developed on mobile devices and markers for object recognition or tracking have been used to reduce the algorithm complexity. [17] showed an AR system using a mobile phone as a client and a PC computer as a server on client/server architecture. A mobile phone is used to take photos of the workspace with markers and the images are sent to a server. The images are processed on the server and then the assembly information is sent back to the mobile phone. On the research of [24] a marker-less tracking subsystem for AR systems was described. Images at 640x480 resolution were used to estimate camera pose in hundreds of milliseconds on a PC computer equipped by Intel Core i7-860 CPU and 3GB of RAM. To eliminate the need to move bulky hardware components, [24] propose the development of an AR system with the same architecture as proposed by [17] so that the mobile device's computational requirements can be lowered. Up until this work, only one project implemented on a tablet [23] was found on the literature, mainly due to limitations on processing power and memory. Nowadays, mobile devices are becoming increasingly powerful platforms. The iPad is a mobile device that stands out for computing power, which can permit execution of image processing algorithms in real time and may be used for AR systems [29].

III. System Description

The proposed AR system is being developed to run an embedded software on an off-the-shelf tablet with a camera, display, gyroscope and accelerometer sensors. It uses a video see-through approach to show the assembly space to the user and graphical signs to guide him or her to the solution. Consequently, no customized device as a HMD is used and no connection to another computer is required. The parts to be assembled are identified by object recognition techniques in a way that markers are not used as an aid to the parts identification process and environment preparation is not needed. Instead of markers, models are used to identify the parts to be assembled.

The system has an offline and an online phase. On the offline phase, or initialization step, models for object recognition are loaded with assembly information which enables the assembly process. Initially, the system is implemented to identify planar objects and only information about the object's edges positions are needed to produce the models. This way, the models for object recognition can be produced from a CAD model or camera device. Features invariant to translation, rotation and scale are extracted from the models and stored in a database. Alternatively, as only position of edges from assembly parts are needed to produce the object recognition models, manufacturers can prepare the models of assembly parts and assembly information in a way that users can download them from the internet as parameters of the embedded software. Yet, on offline phase the intrinsic camera parameters and lens distortion are estimated to enable, on online phase, the estimation of object position and undistortion of the camera image view, respectively.

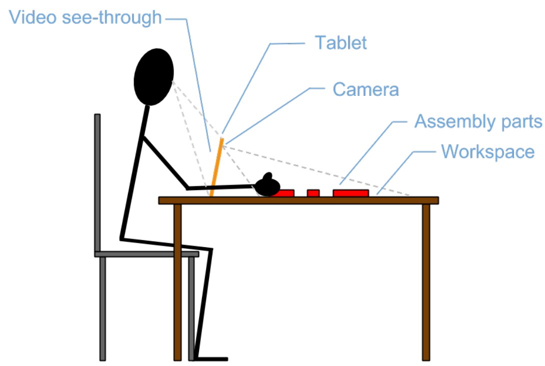


Fig. 1: System setup

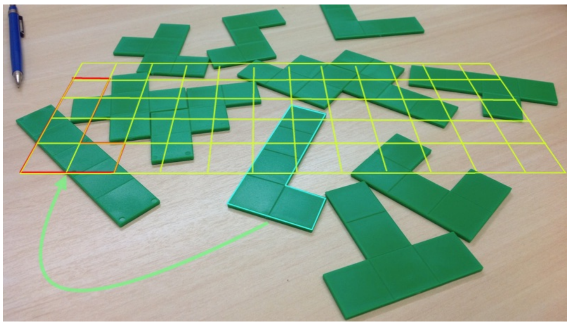


Fig. 2: Simulated user view on tablet's display

On the online phase the tablet device is placed between the user and the assembly parts in a way that the user can see the assembly parts on the tablet's display as a video see-through display, as described by figure 1. The system shows information to guide the user on assembly process on the tablet's display (figure 2). For each image frame the following process occurs: the image of the workspace taken by the device is undistorted using the distortion lens parameters estimated on offline phase. After the undistortion, objects on workspace are segmented. Segmented parts are rectified to remove the perspective distortion as the segmented parts are in perspective due the the tablet's camera position. Features are extracted and then invariant features relative to rotation and scale are computed from the rectified segmented parts. These invariant features are necessary to make the matching possible with models in the database. As pieces are matched to the models, each piece is labeled and one of them is chosen. The chosen labeled piece model is projected on the display on the top of the camera image (piece highlighted in blue on figure 2). An assembly diagram is projected (yellow layout on figure 2) and the place of the chosen labeled piece is highlighted on the assembly diagram (highlighted in red on figure 2). After the user manipulates the piece, the system compares the position and orientation of the chosen piece to the assembly diagram. If the chosen piece is detected on the planned assembly diagram place, the system chooses a new labeled piece and the procedure is repeated until all the pieces are placed on its desired place on the assembly diagram. When the system has verified that all pieces are placed correctly on assembly diagram, it alerts to the user that the final assembly state has been reached. figure 3 shows a block diagram for the online phase algorithm.

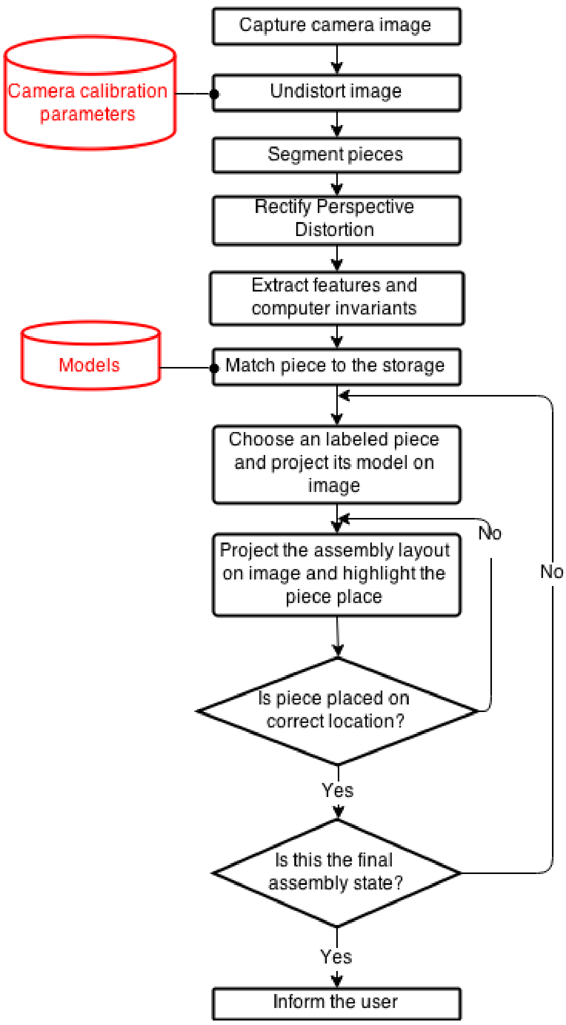


Fig. 3: Block diagram of the online phase of the system

IV. Implementation

The implemented system uses an iPad Air 2 as platform with a video see-through approach. An 18 mm focal distance wide angle lens was placed on the tablet camera to broaden the field of view. Xcode was used to implement the software and the OpenCV library were used to develop the image processing algorithms. All implemented routines were written with Objective-C and C++ languages.

As an use case, the system is being implemented to guide the assembly process of a pentomino puzzle [30]. This puzzle is composed of 12 pieces and each piece is the result of 5 square concatenated by edges.figure 4shows the pentomino puzzle pieces and the given name of each piece. A common puzzle is to tile the pieces in a rectangular box with an area of 6x10 squares (figure 5). This rectangular box can have 2339 possible solutions. The application was selected because it represents a generic assembly process with a known solution where the user can place the pieces in a well defined position from a random distribution of pieces on the workspace. Besides, it can be described by a simple model, simplifying the solution structure.

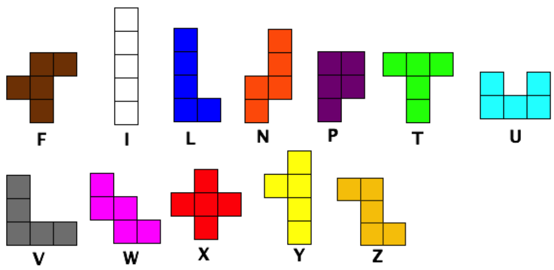


Fig. 4: Pentomino pieces identified by letters

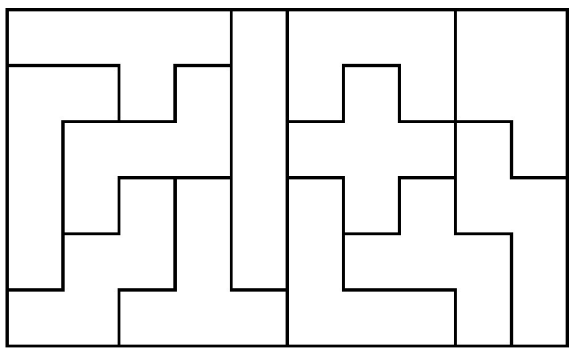


Fig. 5: One possible solution for the 6x10 tiling of pentominoes puzzle

On the offline phase, the calibration is made using the method implemented in the OpenCV library[31]. Using the tablet's camera with the wide angle lens attached, twenty images were taken from a chessboard on unknown orientations to estimate camera intrinsic matrix and lens distortion parameters. If the sum of the squared distances between the observed projection points and projected points (re-projection error) is less than 0.5, the camera intrinsic matrix and lens distortion parameters are stored for later use. If estimated re-projection error is not less than 0.5, the calibration process is repeated.

Assembly parts images were segmented in three steps. Firstly, the Canny Edge algorithm [32] is used to detect edges. A horizontal element of size 1x5 pixels and a vertical element of size 5x1 pixels are used on a morphology closing process to close open borders on the detected edges. Border Following algorithm, as described by [33], is used to join the borders in vectors which represents the border points in counter-clockwise order.

In this particular case of the 2D puzzle solution guidance, it is necessary to make the rectification of perspective distortion to compute assembly pieces as a top-down view. Given that the pieces on puzzle is on the plane $Z = 0$, we can relate a point d on image to a point D on plane $Z = 0$:

$$sd = HD \tag{1}$$

where H is the homography matrix; s is an arbitrary scale.

In our setup, as shown on figure 6, it is required to compute the image points as if the image was taken by a virtual camera placed on $C_v = (0, 0, h_v)_g$ on global coordinate in O. Using the equation (1) a point in the coordinate system O can be represented in the coordinate system C as:

$$s_c d_c = A \begin{bmatrix} -cos(\theta) & 0 & h.\frac{sin^2(\theta)}{cos(\theta)} \\ 0 & 1 & 0 \\ sin(\theta) & 0 & h.sin(\theta) \end{bmatrix} D = H_c D \tag{2}$$

and a point in the coordinate system O can be represented in the coordinate system V as:

$$s_v d_v = A \begin{bmatrix} cos(\pi) & 0 & 0 \\ 0 & 1 & 0 \\ -sin(\pi) & 0 & h_v \end{bmatrix} D = H_v D \tag{3}$$

where the d_c and d_v are the points on the real camera image and virtual camera, respectively; A is the camera intrinsic matrix; D is the point on the plane of assembly parts; H is the homography matrix of the camera or virtual camera; θ is the device inclination angle; h is the distance between the camera and the device opposite side; h_v is the height of the virtual camera.

Relating the equation (2) and equation (3), we can relate an image point to a point as taken by a virtual camera:

$$\alpha d_v = \begin{bmatrix} -1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & h_v \end{bmatrix} \begin{bmatrix} -cos(\theta) & 0 & h.\frac{sin^2(\theta)}{cos(\theta)} \\ 0 & 1 & 0 \\ sin(\theta) & 0 & h.sin(\theta) \end{bmatrix}_c^{-1} d_c \tag{4}$$

or

$$\alpha d_v = H_v H_c^{-1} d_c \tag{5}$$

where α is an arbitrary constant scale.

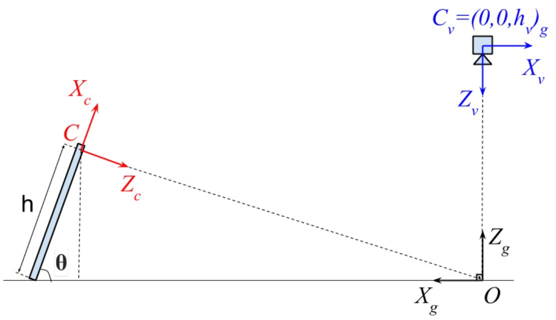


Fig. 6: Perspective effect removal: camera coordinate system C, global coordinate system O and virtual camera coordinate system V

The distance h of the camera and the opposite side of the tablet is known and we used the gyroscope and accelerometer sensor of the tablet to compute the inclination angle θ.

Following the rectification, we used the Ramer-Douglas-Peucker algorithm ([34], [35]) implemented on OpenCV to approximate the segmented contours to polygons, which compressed the edge points and reduced noise. Precision of 1% of contour length was used on the algorithm.

The approximated contours of the segmented parts change if the assembly part is rotated or if they are farther or near from the camera. The parts on the image are recognized by matching to the models of a database, so it is necessary to extract features which are invariant to translation, rotation and scale to make the comparison possible between segmented parts and a model in the database. [36] presents the turning function concept, an approach to change a shape contour to a representation invariant to translation, rotation and scale. Suppose a polygon represented by a curve, we can re-scale the curve in a way to make the total length equal to 1. The Turning Function $\phi(s)$ measures the change of the angle ϕ along the curve as a function of the curve length s. The angle increases on counter-clockwise turn and decreases on clockwise turn as the turns are accumulated along the curve. The turning function have the following features: it is translation invariant, scale invariant, the rotation is represented by a shift on accumulated turns and the choice of initial starting point corresponds to a shift on curve length.

The matching between two polygons is done by calculating a distance as discussed in [37]. The method used is the Polygonal Method. The choice of the initial starting point affects the turning functions, so it is necessary to compute the minimum distance from all possible turning functions computed by different initial starting points. The following metric is used to make the curve matching between two closed polygons A and B given their turning functions $\phi_1(s)$ and $\phi_2(s)$:

$$D(A, B) = \min_{\substack{\alpha \in \Re \\ u \in [0,1]}} [\int_0^1 (\phi_1(s) - \phi_2(s + u) + \alpha)^2 ds]^{\frac{1}{2}} \tag{6}$$

The optimal α is:

$$\alpha = \int_0^1 [\phi_1(s) - \phi_2(s)]ds - 2\pi u \tag{7}$$

where u is the choice of a starting point and α is the rotation difference between the two polygons.

For each segmented object it is computed the turning function. Then we compute the minimum distance between the segmented object and each model in the database by the use of equation (6). The minimum is achieved when the best orientation between the curves and starting point is selected and a matching is made when the minimum distance computed to the model is less than a selected threshold of 0.2. Due to the nature of pentomino pieces, some flipped pieces (mirrored pieces) are not recognized as the same object. We solved this problem by calculating the turning function of the mirrored models too, this way each model have two possible turning functions.

Assembly parts are unique on pentominoes puzzle, therefore it is necessary to treat repeated identified parts. If a model is matched to more than one contour, the segmented image object with the minimum distance to the model is held as a match to the model. Yet, due to the noise on captured image the segmentation algorithm can detect the noise as a small contour. Elimination of this noise contour is made by comparing the arc length of contours matched to the same model. If the arc length of one contour is less than 20% of the comparing contour, the contour with the minor arc length is discarded as a possible match to the model.

V. Results and discussion

We processed 35 images with the 12 pieces on random orientation and position without occlusion to evaluate segmentation and object recognition algorithms.

The table I summarizes the piece segmentation result. From a total of 420 pieces the algorithm segmented 413 pieces, thus, the segmentation ratio is 0.98.

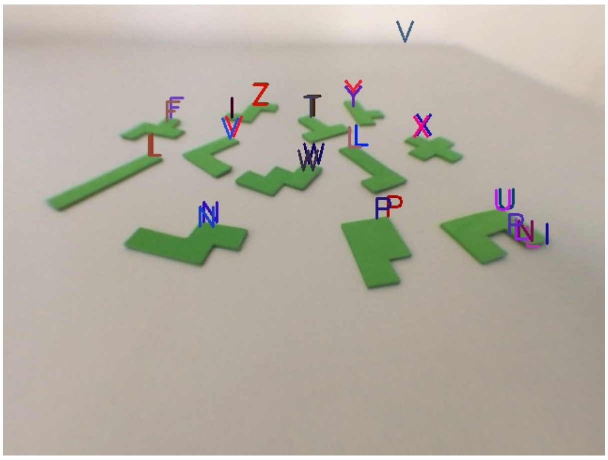
Analyzing the pieces which could not be segmented, we noticed that the segmentation failed due to not fully closed borders. The morphology closing was not sufficient to close some borders and the border following algorithm accumulated the points of the border as an open contour. The segmentation result may be improved if a larger element is used on the morphology closing. However, larger elements may deform the shape, which makes the identification task more challenging.

On our first tests on object recognition using the turning function distance and only selecting distances less than a threshold of 0.2, the object recognition algorithm recognized several contours as pentomino pieces and there were many false positives due to noise contours, as we can see an example on figure 7a.

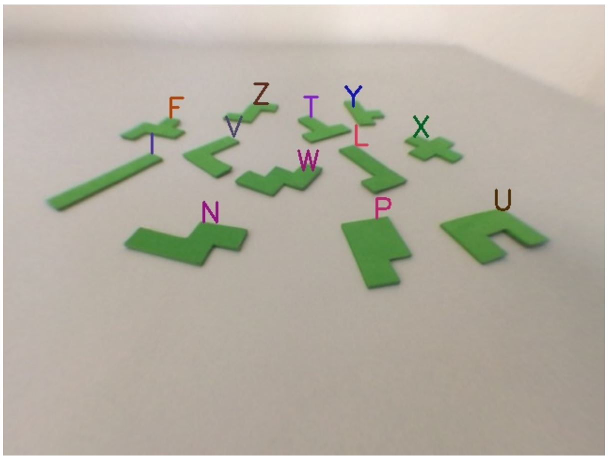
Object recognition improvement has been achieved when we assumed unique pentomino pieces and filtered segmented contours proportionally small in comparison to others as we described on section §IV. We can see an example of the result of the algorithm with this additional filter on figure 7b.

TABLE I. Segmentation result

Piece	Number	Segmented	Segmentation rate
F	35	34	0.97
I	35	35	1
L	35	34	0.97
N	35	34	0.97
P	35	35	1
T	35	35	1
U	35	35	1
V	35	35	1
W	35	34	0.97
X	35	35	1
Y	35	35	1
Z	35	32	0.97
Total	*420*	*413*	*0.98*



(a) Object recognition using turning function distance with a threshold filtering



(b) Recognition assuming unique pentominoes and filtering small contours

Fig. 7: Example of pentomino parts recognition

Object recognition results assuming unique pentomino pieces and with small contours filter are summarized on table II. As object recognition depends directly from segmentation result, we only counted segmented parts for object recognition. From a total of 413 segmented parts, 407 were correctly identified. We noticed that some of the unrecognized

TABLE II. OBJECT RECOGNITION RESULT

Piece	Segmented	Identified	Identification rate
F	34	33	0.97
I	35	35	1
L	34	33	0.97
N	34	34	1
P	35	34	0.97
T	35	35	1
U	35	33	0.94
V	35	35	1
W	34	34	1
X	35	35	1
Y	35	34	0.97
Z	32	32	1
Total	*413*	*407*	*0.98*

TABLE III. ALGORITHMS EXECUTION TIME

	Time (ms)	
Algorithm	***Mean***	***Standard deviation***
Undistortion	33.5	2.71
Segmentation	27.9	2.52
Perspective rectification	0.340	0.0452
Object recognition	165	17.0
Total	*227*	*8.68*

puzzle parts were near the edge of camera image. They were affected by lens distortion correction that acts more aggressively on pixels far from the center of image, which deforms the piece shape. Yet, in some unrecognized cases, false positive matches have been made to some small contours detected inside the pieces.

Near real time performance is desired for the implemented AR system so it is important to evaluate the execution time of each algorithm. We placed pentomino pieces in a way that the camera can capture all the 12 pieces and executed the algorithms 500 times. Execution time for each process is summarized on table III. We can observe that object recognition execution time had a great significance on the whole process execution time. It is desired to focus on object recognition if we desire to improve the whole process significantly.

The desired performance has not been achieved yet but we are working in several algorithms optimizations such as color segmentation in the segmentation phase and reuse of equivalent turning functions in object recognition phase. Color segmentation reduces the number of segmented contours, and consequently improvement on object recognition execution time can be achieved. Still, on implemented system we mirrored all the pentomino models to represent two different turning functions. However, the V, I, X, U and W pentomino pieces are equivalent to the mirrored pieces if rotated, therefore, its turning functions are equivalent. Improvement on object recognition performance can be achieved if we reduce the number of mirrored pentominoes turning functions to the strictly necessary to compute the turning function distance.

VI. CONCLUSION

This paper presented an AR system that is being developed to guide assembly processes. Segmentation and object recognition are two important aspects of this project. Assembly parts segmentation is made by a combination of Canny Edge algorithm, morphology closing and border following algorithm. Segmentation rate of 0.98 has been achieved. Segmented assembly parts are identified by computing the turning function and the distance between the turning functions of segmented assembly parts and models. We made the assumption that all the 12 pentomino parts are present on the evaluated images and a simple filter algorithm have been used to reduce the likelihood of noise contour being incorrectly labeled as a pentomino part. We achieved a rate of 0.98 for object recognition with this process.

We evaluated the execution time of the 4 system's main processes: undistortion, perspective rectification, segmentation and object recognition. It was achieved a mean of approximately 227 ms to the whole process execution time and an execution time of 165 ms for object recognition of 12 pentomino pieces. As an AR system that works in near real time is desired, our next step is to improve the performance of the object recognition process. Currently, optimizations are being evaluated to improve the system and achieve near real time performance. There are two approaches to follow: reduce the number of segmented contours on images and reduce the number of turning functions computed from mirrored models.

REFERENCES

[1] A. Tang, C. Owen, F. Biocca, and W. Mou, "Comparative Effectiveness of Augmented Reality in Object Assembly," in *Proceedings of the SIGCHI Conference on Human Factors in Computing Systems*, (New York, New York, USA), pp. 73–80, ACM Press, 2003.

[2] S. Wiedenmaier, O. Oehme, L. Schmidt, and H. Luczak, "Augmented reality (AR) for assembly processes design and experimental evaluation," *International Journal of HumanComputer Interaction*, vol. 16, no. 3, pp. 497–514, 2003.

[3] K. M. Baird and W. Barfield, "Evaluating the effectiveness of augmented reality displays for a manual assembly task," *Virtual Reality*, vol. 4, pp. 250–259, Dec. 1999.

[4] T. Caudell and D. Mizell, "Augmented reality: an application of heads-up display technology to manual manufacturing processes," *Proceedings of the Twenty-Fifth Hawaii International Conference on System Sciences*, pp. 659–669 vol.2, 1992.

[5] S. Feiner, B. Macintyre, and D. Seligmann, "Knowledge-based augmented reality," *Communications of the ACM*, vol. 36, pp. 53–62, July 1993.

[6] D. Reiners, D. Stricker, G. Klinker, and S. Müller, "Augmented reality for construction tasks: Doorlock assembly," *Proc. IEEE and ACM IWAR*, 1998.

[7] B. Odenthal, M. P. Mayer, W. Kabuß, and C. M. Schlick, "A Comparative Study of Head-Mounted and Table-Mounted Augmented Vision Systems for Assembly Error Detection," *Human Factors and Ergonomics in Manufacturing and Service Industries*, vol. 24, no. 1, pp. 105–123, 2014.

[8] A. Boud, D. Haniff, C. Baber, and S. Steiner, "Virtual reality and augmented reality as a training tool for assembly tasks," in *1999 IEEE International Conference on Information Visualization (Cat. No. PR00210)*, pp. 32–36, IEEE Comput. Soc, 1999.

[9] J. Zauner, M. Haller, A. Brandl, and W. Hartman, "Authoring of a mixed reality assembly instructor for hierarchical structures," in *The Second IEEE and ACM International Symposium on Mixed and Augmented Reality, 2003. Proceedings.*, pp. 237–246, IEEE Comput. Soc, 2003.

[10] T. Salonen, J. Sääski, M. Hakkarainen, T. Kannetis, M. Perakakis, S. Siltanen, A. Potamianos, O. Korkalo, and C. Woodward, "Demonstration of assembly work using augmented reality," in *Proceedings of the 6th ACM international conference on Image and video retrieval - CIVR '07*, (New York, New York, USA), pp. 120–123, ACM Press, 2007.

[11] S. Henderson and S. Feiner, "Exploring the Benefits of Augmented Reality Documentation for Maintenance and Repair," *IEEE Transactions on Visualization and Computer Graphics*, vol. 17, no. 10, pp. 1355–1368, 2011.

[12] Z. B. Wang, S. K. Ong, and a. Y. C. Nee, "Augmented reality aided interactive manual assembly design," *The International Journal of Advanced Manufacturing Technology*, vol. 69, pp. 1311–1321, June 2013.

[13] B. M. Khuong, K. Kiyokawa, A. Miller, J. J. La Viola, T. Mashita, and H. Takemura, "The effectiveness of an AR-based context-aware assembly support system in object assembly," *2014 IEEE Virtual Reality (VR)*, pp. 57–62, Mar. 2014.

[14] H.-K. Chiang, Y.-Y. Chou, L.-C. Chang, C.-Y. Huang, F.-L. Kuo, and H.-W. Chen, "An augmented reality learning space for PC DIY," in *Proceedings of the 2nd Augmented Human International Conference*, (New York, New York, USA), p. 12, ACM Press, 2011.

[15] M. Kitagawa and T. Yamamoto, "3D puzzle guidance in augmented reality environment using a 3D desk surface projection," in *2011 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 133–134, IEEE, Mar. 2011.

[16] M. Fiorentino, A. E. Uva, M. Gattullo, S. Debernardis, and G. Monno, "Augmented reality on large screen for interactive maintenance instructions," *Computers in Industry*, vol. 65, pp. 270–278, Feb. 2014.

[17] M. Hakkarainen, C. Woodward, and M. Billinghurst, "Augmented assembly using a mobile phone," *2008 7th IEEE/ACM International Symposium on Mixed and Augmented Reality*, pp. 167–168, Sept. 2008.

[18] M. Nakanishi, M. Ozeki, T. Akasaka, and Y. Okada, "Human factor requirements for Applying Augmented reality to manuals in actual work situations," *2007 IEEE International Conference on Systems, Man and Cybernetics*, pp. 2650–2655, 2007.

[19] Y. Li, Y. Kameda, and Y. Ohta, "AR replay in a small workspace," in *2013 23rd International Conference on Artificial Reality and Telexistence (ICAT)*, pp. 97–101, IEEE, Dec. 2013.

[20] K. Murakami, R. Kiyama, T. Narumi, T. Tanikawa, and M. Hirose, "Poster: A wearable augmented reality system with haptic feedback and its performance in virtual assembly tasks," in *2013 IEEE Symposium on 3D User Interfaces (3DUI)*, pp. 161–162, IEEE, Mar. 2013.

[21] M. L. Yuan, S. K. Ong, and A. Y. C. Nee, "Augmented reality for assembly guidance using a virtual interactive tool," *International Journal of Production Research*, vol. 46, pp. 1745–1767, Apr. 2008.

[22] J. Novak-Marcincin, J. Barna, M. Janak, L. Novakova-Marcincinova, and J. Torok, "Visualization of intelligent assembling process by augmented reality tools application," in *2012 4th IEEE International Symposium on Logistics and Industrial Informatics*, pp. 33–36, IEEE, Sept. 2012.

[23] J. Serván, F. Mas, J. Menéndez, and J. Ríos, "Assembly Work Instruction Deployment Using Augmented Reality," *Key Engineering Materials*, vol. 502, pp. 25–30, Feb. 2012.

[24] H. Alvarez, I. Aguinaga, and D. Borro, "Providing guidance for maintenance operations using automatic markerless Augmented Reality system," *2011 10th IEEE International Symposium on Mixed and Augmented Reality*, pp. 181–190, Oct. 2011.

[25] Z. Wang, Y. Shen, S. K. Ong, and A. Y.-C. Nee, "Assembly Design and Evaluation Based on Bare-Hand Interaction in an Augmented Reality Environment," in *2009 International Conference on CyberWorlds*, pp. 21–28, IEEE, 2009.

[26] S. Ong and Z. Wang, "Augmented assembly technologies based on 3D bare-hand interaction," *CIRP Annals - Manufacturing Technology*, vol. 60, pp. 1–4, Jan. 2011.

[27] J. Sääski, T. Salonen, M. Hakkarainen, S. Siltanen, C. Woodward, and J. Lempiäinen, "Integration of Design and Assembly Using Augmented Reality," *Micro-Assembly Technologies and Applications.*, vol. 260, pp. 395–404, 2008.

[28] G. Westerfield, *Intelligent Augmented Reality Training for Assembly and Maintenance*. Master of science, University of Canterbury, 2012.

[29] J. Carmigniani, B. Furht, M. Anisetti, P. Ceravolo, E. Damiani, and M. Ivkovic, "Augmented reality technologies, systems and applications," *Multimedia Tools and Applications*, vol. 51, pp. 341–377, Dec. 2010.

[30] S. W. Golomb, *Polyominoes: puzzles, patterns, problems, and packings*. Princeton University Press, 2 ed., 1996.

[31] Z. Z. Z. Zhang, "Flexible camera calibration by viewing a plane from unknown orientations," *Proceedings of the Seventh IEEE International Conference on Computer Vision*, vol. 1, no. c, pp. 666–673, 1999.

[32] J. Canny, "A computational approach to edge detection.," *IEEE transactions on pattern analysis and machine intelligence*, vol. 8, pp. 679–98, June 1986.

[33] S. Suzuki and K. Abe, "Topological structural analysis of digitized binary images by border following," *Computer Vision, Graphics, and Image Processing*, vol. 30, pp. 32–46, Apr. 1985.

[34] U. Ramer, "An iterative procedure for the polygonal approximation of plane curves," 1972.

[35] D. H. Douglas and T. K. Peucker, "Algorithms for the reduction of the number of points required to represent a digitized line or its caricature," *Cartographica: The International Journal for Geographic Information and Geovisualization*, vol. 10, no. 2, pp. 112–122, 1973.

[36] E. M. Arkin, L. P. Chew, D. P. Huttenlocher, K. Kedem, and J. S. B. Mitchell, "An efficiently computable metric for comparing polygonal shapes," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 13, pp. 209–216, Mar. 1991.

[37] D. Cakmakov and E. Celakoska, "Estimation of curve similarity using turning functions," *International Journal of Applied Mathematics*, vol. 15, pp. 403–416, 2004.