HUMAN-ROBOT INTERACTION AND TRACKING USING LOW COST 3D VISION SYSTEMS

ENRIQUE MARTINEZ BERTI, ANTONIO JOSÉ SANCHEZ SALMERÓN, FRANCESC BENIMELI

Abstract. This paper describes a platform which allows humans to interact with robotic arms using augmented reality. Low cost "kinect" cameras (Xbox 360) are used for tracking human skeletons and locations of robot's end effectors. The main goal of this paper is to develop robust trackers on this platform. Concretely, a Kalman filter is used for tracking robotic arms using data received from these sensors. It comes to finding a low cost platform for human-robot interactions.

Key words: low cost vision system, Kalman filter, augmented reality, kinematics, human-machine interaction.

1. INTRODUCTION

There is a wide range of industrial processes in which robotic systems are present. Nowadays, the required characteristics for such industrial processes are high efficiency, flexibility and adaptability. Human-robotic systems interaction is a key solution to accomplish these requirements, establishing a synergy between the best features of both robots and humans: robot's precision and high efficiency and human's flexibility and adaptability.

Human-machine interactions have numerous applications such as assembly tasks [3, 12], wheelchair controls through different types of sensors [2, 13], developments of servomechanisms [1], developments of intelligent robots [4], and so on.

This paper provides the basis for human-machine interaction in order to increase efficiency in pieces assembly processes, whose flexibility and adaptability characteristics require a clos interaction between humans and the robotic systems. Interactions between humans and robots improve the efficiency of complex assembly processes, especially when intelligence is required by the system [3]. However, a precondition for this close relationship is human safety. Many research advances have been carried out in this area and now, some surveillance systems based of sensors used to interact with robots in the market can be found. Intelligent

Instituto de Automática e Informática Industrial, Univesitat Politècnica de València, Valencia, Spain

Rom. J. Techn. Sci. – Appl. Mechanics, Vol. 57, N^{os} 2–3 P. 151–168, Bucharest, 2012

assistance devices (IAD) are the basis for introducing human beings in assembly processes in order to use their cognitive and sensory-motor skills to carry out assemblies with high flexibility.

The main goal of this research consists on creating a platform which can be used as a basis for developing applications with interactions between humans and machines. A simple practical case of human-robot interaction has been implemented to check this platform.

1.1. RELATED RESEARCH

We consider the problem of estimating and tracking 3D configurations of complex articulated objects from images, *e.g.*, for applications requiring 3D robot arms pose, human body pose and hand gesture analysis. There are two main schools of thought on this. Model-based approaches presuppose an explicitly known parametric articulated object model and estimate the pose either by directly inverting the kinematics (which has many possible solutions and which requires known image positions for each part [26]) or by numerically optimizing some form of model-image correspondence metric over the pose variables, using a forward rendering model to predict the images (which is expensive and requires a good initialization, and the problem always has many local minima [24]). An important subcase is model-based tracking, which focuses on tracking the pose estimate from one time step to the next starting from a known initialization based on an approximate dynamical model [17, 23]. In contrast, learning-based approaches try to avoid the need for explicit initialization and accurate 3D modeling and rendering, instead capitalizing on the fact that the set of typical articulated object poses is far smaller than the set of kinematically possible ones and learning a model that directly recovers pose estimates from observable image quantities. In particular, example-based methods explicitly store a set of training examples whose 3D poses are known, estimating pose by searching for training image(s) similar to the given input image and interpolating from their poses [15, 19, 22, 25].

There is a good deal of prior work on articulated objects pose analysis, but relatively little on directly learning 3D pose from image measurements. Brand [16] models a dynamical manifold of human body configurations with a Hidden Markov Model and learns using entropy minimization, Athitsos and Sclaroff [14] learn a perceptron mapping between the appearance and parameter spaces, and Shakhnarovich *et al.* [22] use an interpolated-k-nearest-neighbor learning method. Human pose is hard to ground truth, so most papers in this area [14, 16, 19] use only heuristic visual inspection to judge their results. However, Shakhnarovich *et al.* [22] used a human model rendering package (POSER from Curious Labs) to synthesize ground-truthed training and test images of 13 degrees of freedom upper body poses with a limited (±40º) set of random torso movements and viewpoints. Several publications have used the image locations of the center of each body joint

as an intermediate representation, first estimating these joint centers in the image, then recovering 3D pose from them. Howe *et al.* [18] develop a Bayesian learning framework to recover 3D pose from known centers, based on a training set of pose-center pairs obtained from resynthesized motion capture data. Mori and Malik [19] estimate the centers using shape context image matching against a set of training images with prelabeled centers, then reconstruct 3D pose using the algorithm of [26]. These approaches show that using 2D joint centers as an intermediate representation can be an effective strategy.

With regard to tracking, some approaches have learned dynamical models for specific human motions [20, 21]. Particle filters and MCMC methods have been widely used in probabilistic tracking frameworks, *e.g.* [23, 27]. Most of these methods use an explicit generative model to compute observation likelihoods.

1.2. OVERWIEV OF THE APROACH

Using the same philosophy as for tracking human skeleton, former approaches can be applied to track robot arms. We propose to use a low cost vision system which requires a discrete Kalman filter. This allows tracking join variables at each instant of time.

1.3. ORGANIZATION

Section 2 describes the global system. Section 3 describes de low cost 3D vision system and calibration system. Section 4 presents the robot used and descrive the kalman filter uses to track a robot arm. Section 5 describes a human skeleton tracking approach. Section 6 describes an example to human-robot interaction. Finally, Section 7 concludes with some discussions and directions of future work.

2. GLOBAL SYSTEM DESCRIPTION

A platform for human-machine interaction using augmented reality [8] has been performed between robotic systems and human beings. This platform is a distributed system where processes can communicate easily between them. The main functionalities offered by this platform are:

1) Communications between processes via XML [5].
2) Safety controls.
3) Tracking of robot arm poses.
4) Tracking of human skeletons.
5) Handle of augmented reality scenes.

A practical case of human-robot interaction has been implemented to check this platform. Figure 1 shows the distribution of the physical components (robot and camera) of this application.

Fig. 1 – Low cost 3D vision system.

In this case, the distributed system is composed by two processes to perform interactions between a human and a robot. One process realizes the monitoring of human skeleton. The other process realizes the monitoring of the robot arm pose. The data information for human and robot state estimation is obtained by a low-cost 3D camera.

The XML messages set developed ad-hoc for an application use special communication software. This software is called RT-SCORE [6] and it is a system (based on a blackboard system) that allows to assign a communication channel between processes. The "channel" concept is similar to a "hall" in a chat communication system (the chat communication is RT-SCORE); so, only the entities connected to a channel receive the information sent into this channel.

Safety regulations require introducing guardrails, so that humans do not have direct access to industrial robots workspaces. To achieve a human–robot interaction a safety protocol has been established that allows such interaction without risk of serious damages. The safety control system calculates the human and robotic arms location, so that the closer they get, the slower the robot moves.

MatLab [10] has been used in order to implement both processes.

The robot arm monitor tracks the end effector and joint angles of the robot. However, it is needed a Kalman filter for tracking robot arm poses.

The human skeleton monitor uses third party libraries with functions to estimate locations of each body part. Some human skeleton poses are used to handle virtual objects in the augmented reality scenario. Additionally these virtual objects can be shown on real RGB images captured by the camera.

3. LOW COST 3D VISION SYSTEM

The vision system used in this practical case is the "Kinect" camera, which consists of two optical sensors whose interaction allows a three-dimensional scene analysis. One of the sensors is an RGB camera which has a video resolution of 30 fps. The image resolution given by this camera is 640×480 pixels. The second sensor has the aim of obtaining depth information corresponding to the objects found at the scene. The working principle of this sensor is based on the emission of an infrared signal which is reflected by the objects and captured by a monochrome CMOS sensor. A matrix is then obtained which provides a depth image of the objects in the scene, called DEPTH.

Calibration is needed to relate both camera and robot coordinates reference systems. Therefore objects located by the camera can be handled by the robot.

3.1. 3D VISION SYSTEM CALIBRATION

The intrinsic and extrinsic parameters of the two Kinect optical sensors are different. Therefore, it is necessary to calibrate one optical sensor (RGB) with respect to the other (DEPTH) in order to relate the corresponding pixels in both images, as shown in Fig. 2.

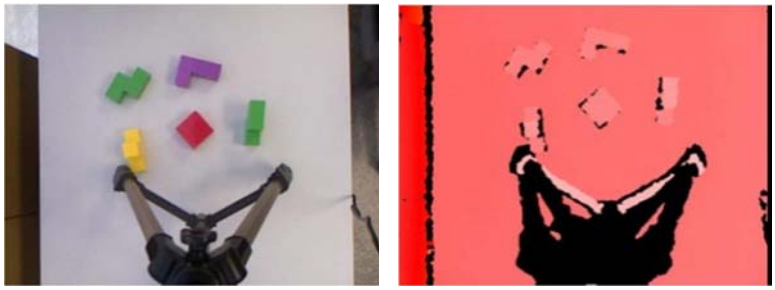

Fig. 2 – Left – RGB image; right – DEPTH image.

Fig. 3 – Overlap of RGB and DEPTH images.

To correct the gap between pixels from the two images, the following procedure is used:

1. Detect a 2D point *prgb* =(*xrgb, yrgb*) on the RGB image.
2. Project *pi* into three-dimensional space *PRGB* = (*XRGB, YRGB,*1) through the following equations:

$$X_{RGB} = \left(\frac{x_{rgb} - u_0}{\alpha_x} \right) \tag{1}$$

$$Y_{RGB} = \left(\frac{y_{rgb} - v_0}{\alpha_y} \right) \tag{2}$$

$$Z_{RGB} = 1, \tag{3}$$

where:

u_0 – x coordinate of the main point of the RGB camera;

v_0 – y coordinate of the main point of the RGB camera;

α_x – focal length in pixels;

α_y – focal length in pixels.

For the present case of study, the applied intrinsic values of the camera $\left(u_0, v_0, \alpha_x, \alpha_y\right)$ are presented in [9].

3. Transform point P_{RGB} into three-dimensional point P_D with reference to the coordinate system of the DEPTH image.

$$\begin{bmatrix} X_D \\ Y_D \\ Z_D \\ 1 \end{bmatrix} = \begin{bmatrix} R & t \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X_{RGB} \\ Y_{RGB} \\ Z_{RGB} \\ 1 \end{bmatrix}. \tag{4}$$

The transformation matrix is composed of: rotation matrix R and translation vector t. The rotation matrix was obtained as described in [9]. The translation vector was obtained experimentally by overlapping the DEPTH and RGB images. The resulting transformation matrix is:

$$M = \begin{bmatrix} 0.99984 & 0.00126 & -0.01748 & 0.03 \\ -0.00147 & 0.99992 & -0.01225 & -0.01 \\ 0.01747 & 0.01227 & 0.99977 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{5}$$

4. Once the three-dimensional point P_D is obtained, it is projected into the DEPTH image plane, hence obtaining a 2D point $p_d = (x_d, y_d)$.

$$x_d = \left(\frac{X_D * \alpha_x}{z = 1} \right) + u_0 \tag{6}$$

$$y_d = \left(\frac{Y_D * \alpha_x}{z = 1} \right) + v_0 \tag{7}$$

The implementation of this procedure allows to considerably reduce the gap between the images. Figure 4 shows the calibration results:

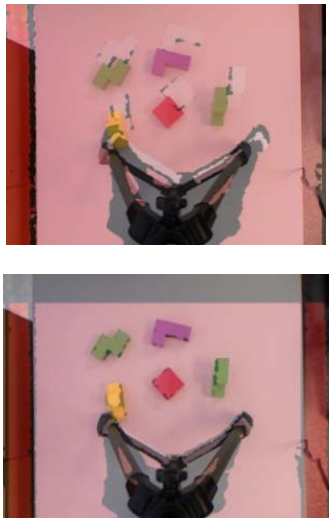

Fig. 4 – Left – overlap of RGB and DEPTH images before calibration;
right – overlap of RGB and DEPTH images after calibration.

3.2. COORDINATE SYSTEM UNIFICATION

As the robotic device and the vision system must be integrated, it is necessary to implement a calibration process in order to obtain the transformation matrix that relates the coordinate systems of both elements. The calibration process is performed as follows.

A set of 3D points whose location with respect to the robot coordinate system is known, are processed and their locations in reference to the camera coordinate system determined.

The point to be detected by the camera is assumed to be the Tool Center Point (TCP) of the tool applied in the process, which also coincides with the center

of the object grasped by the tool. This is shown in Fig. 5, where the white point corresponds to center of the object identified by camera.

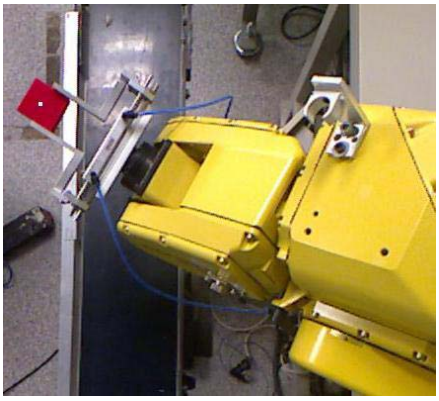

Fig. 5 – White point – TCP with vision system.

As mentioned in section 3.1, 3D representation of a point in a picture can be performed through equations (1) and (2). The Z_{RGB} coordinate of a pixel in the RGB image is obtained from the value of the interesting pixel in the DEPTH image (white point in Fig. 5), according to [28], as follows:

$$Z_{RGB} = 0.1236\left(\tan\left(\frac{R_d}{2842.5} + 1.1863\right)\right), \tag{8}$$

where R_d (Raw Disparity) is a value of depth provided by the Kinect.

Once the locations of the points with respect to the camera coordinate system are obtained, the transformation matrix can be computed. Therefore, from at least 4 points, the linear equation system (9) is obtained and solved by least squares.

$$Ax = b, \tag{9}$$

where A contains information from the 3D points in reference to the camera coordinate system and "b" contains information from the 3D points in reference to the robot coordinate system. Solution $\breve{x}$ to this equation can be estimated using:

$$A^{\mathrm{T}} A\breve{x} = A^{\mathrm{T}} b \qquad \therefore \qquad \breve{x} = \left(A^{\mathrm{T}} A\right)^{-1} A^{\mathrm{T}} b, \tag{10}$$

which results in the following transformation matrix

$$M = \begin{bmatrix} -0.0004 & 1.0026 & -0.0627 & -0.1025 \\ 1.0076 & 0.0203 & -0.0146 & -0.5670 \\ -0.0052 & -0.0445 & -0.9311 & 0.5168 \\ 0 & 0 & 0 & 1 \end{bmatrix}. \tag{11}$$

This matrix allows us to estimate a 3D location of a point in space in reference to the robot coordinate system (PR) starting from its location in reference to the Kinect coordinate system (P_K):

$$\begin{bmatrix} X_R \\ Y_R \\ Z_R \\ 1 \end{bmatrix} = [M] \begin{bmatrix} X_K \\ Y_K \\ Z_K \\ 1 \end{bmatrix}. \tag{12}$$

4. ROBOT FANUC 200IB CONTROL

DH (Denavit-Hartemberg) is used to solve direct and inverse kinematics problems. Figure 2 shows coordinate systems and articulation axes used for this Fanuc robot.

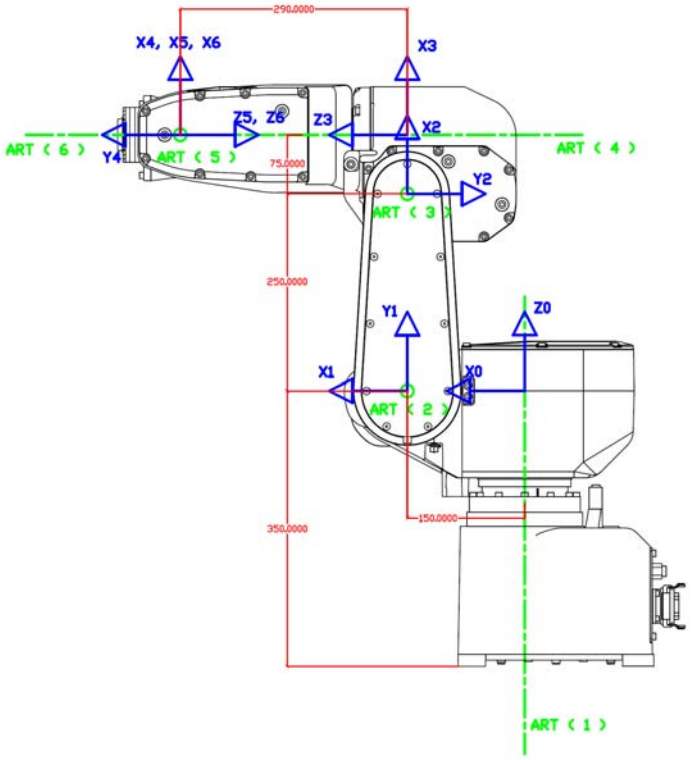

Fig. 6 – Coordinate systems and articulation axes.

Hence, the following DH parameter is obtained (Table 1):

Table 1

Denavit-Hartemberg

ART	θ	d_i	a_i	α_i
1	0º	0	150	90º
2	90º	0	250	0
3	0º	0	75	90º
4	0º	290	0	90º
5	0º	0	0	90º
6	0º	0	0	0

The direct and inverse kinematics problems are solved using these parameters. These kinematic models allow tracking the robot by using the Kinect, so that the end effector position is identified on the image and the state of the robot joints is calculated. The Kalman filter is necessary to filter the information captured by vision sensor and realize robot tracking.

4.1. KALMAN FILTER

The Kalman filter [11] is used in sensor fusion and data fusion. Typically real time systems produce multiple sequential measurements rather than making a single measurement to obtain the state of the system. These multiple measurements are then combined mathematically to generate the system's state at that time instant.

Data fusion using a Kalman filter can assist computers to track objects in videos with low latency (not to be confused with a low number of latent variables). The tracking of objects is a dynamic problem, using data from sensor and camera images that always suffer from noise. This can sometimes be reduced by using higher quality cameras and sensors but can never be eliminated, so it is often desirable to use a noise reduction method.

The iterative predictor-corrector nature of the Kalman filter can be helpful, because at each time instance only one constraint on the state variable needs to be considered. This process is repeated considering a different constraint at every time instance. All the measured data are accumulated over time and help in predicting the state.

Video can also be pre-processed, using a segmentation technique, to reduce computation and hence latency.

The discrete Kalman filter [11] is implemented as follows:

1) State prediction:

$$\hat{X}_t^* = A\hat{X}_{t-1}. \tag{13}$$

1) Prediction of error covariance:

$$P^*_t = AP_{t-1}A^{\mathrm{T}} + Q. \tag{14}$$

2) Calculate the constant gain K:

$$K_t = P^*_t H^{\mathrm{T}} \left(HP^*_t H^{\mathrm{T}} + R \right)^{-1}. \tag{15}$$

3) Update:

$$\hat{X}_t = \hat{X}^*_t + K_t \left(Z_t - H\hat{X}^*_t \right). \tag{16}$$

4) Update error covariance:

$$P_t = \left(I - K_t H \right) P^*_t. \tag{17}$$

The Kalman filter has been applied to depth information. The values returned by depth images are not always right. This happens because the sensor does not detect the depth correctly when the infrared light is not properly reflected on the object. In this case, the input value to the Kalman filter is the depth value of z_d (state) corresponding to the distance between the camera and the object. In (13) and (14) the z_d value and covariance is predicted to the next step. Equations (15), (16) and (17) are the equations to correct the discrete Kalman filter. In (15), a new gain of Kalman is calculated. Equations (16) and (17) calculate a new value of z_d predicted, and new covariance of error, respectively.

Three Kalman filters have been implemented: one for each of the three points used to locate (position and orientation) the end effector. Figure 7 shows a pose estimated during the robot movement. It can be seen the three points detected on the end effector.

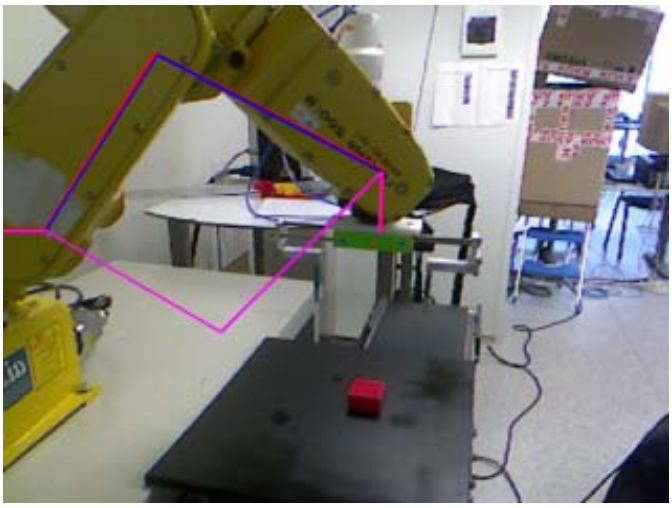

Fig. 7 – Robot arm pose tracking.

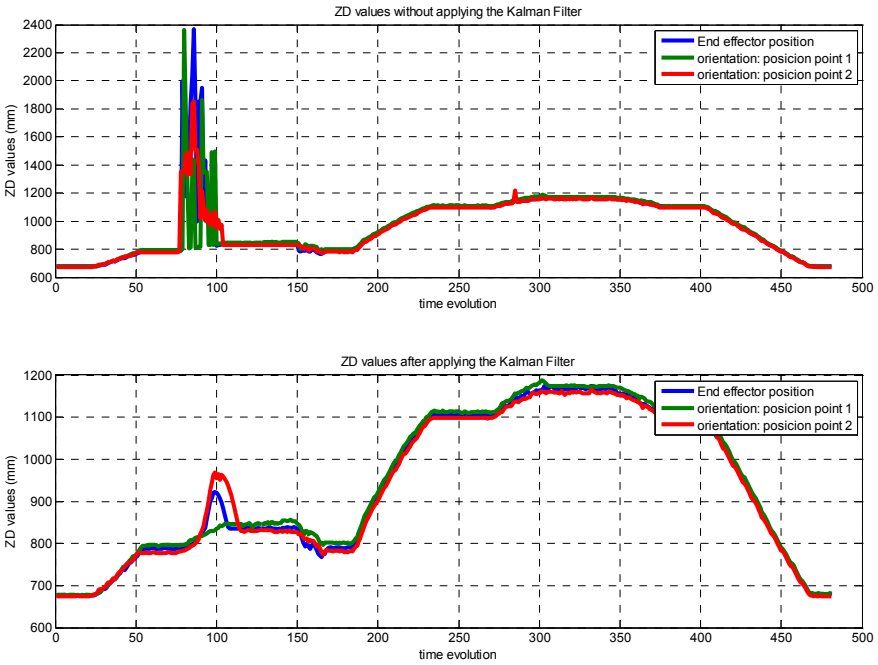

Fig. 8 – Kalman Filter evolution.

Figure 8 shows the evolution of depth information for comparing results obtained with Kalman filter and without applying the Kalman filter. The first graph shows parameter zd over time without applying the Kalman filter. The second graph shows zd over time applying the Kalman filter. At time 100, incorrect zd values can be observed when not applying the filter because the camera does not get properly information. It can be seen that the Kalman filter makes a correction of these values.

5. MONITORING HUMAN SKELETON

To monitor the skeleton of a human being, the toolbox [7] and [8] has been used in Matlab. The NITE tracking human body module aim is based on extracting the most important features of the human skeleton and following them over time. Figure 5 shows the result of the skeleton detection by the kinect.

Tracking the human skeleton is necessary to control the actions in order to interact with the robot. In the present case study, it is used to insert virtual parts in the scene. A virtual piece will be created on the hand using a set gesture and with another set gesture the object will be fixed in that position. Fig. 10 shows the gesture to pick up a virtual object and Fig. 11 shows the gesture to place the object in a fixed position.

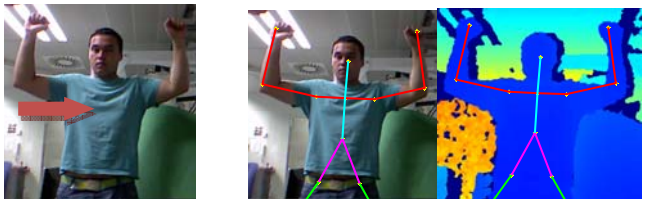

Fig. 9 – Skeleton pose control.

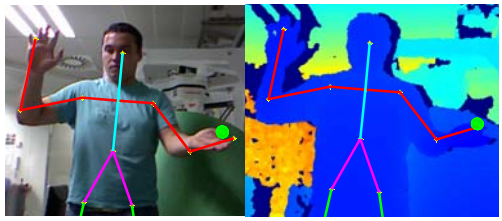

Fig. 10 – Take object.

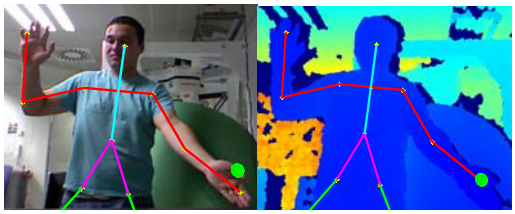

Fig. 11 – Place object.

The act of creating a virtual object is performed by placing the arm and forearm in an angle of 90 degrees. Once the piece appears on the image, it follows the arm movements until the subject performs the gesture for placing the piece, which consists on stretching out the arm to the desired position.

6. EXAMPLE

The platform used to perform the human-robot interaction is described below.

The main application in the platform performs a hybrid assembly operation [3]. This operation is performed by a robot arm and a human. Tracking of the robot arm and the human skeleton is performed using a Kalman filter.

The necessary information from the environment is obtained through two low cost 3D cameras (Kinect), each one with a different function. The first one detects and identifies the objects in the assembly and provides their corresponding 3D locations (Kinect-1). The second one tracks the robot arm and human locations using information about space (Kinect-2). The objects in the assembly can be real

objects or virtual objects. Virtual objects are visualized in the RGB image from Kinect-2 using augmented reality. Figure 12 shows the distribution of the components in the physical system.

Fig. 12 – Low cost 3D vision system.

Figure 13 shows a scheme describing the system integration.

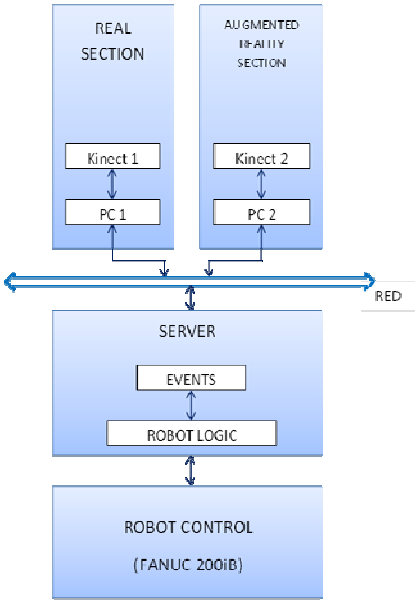

Fig. 13 – Integration scheme.

According to the type of objects in the space, two different sections can be considered in the system: a section with real objects and a section with virtual objects generated through augmented reality. The information for each one of these sections is provided by the corresponding Kinect.

In order to develop the example a multiplatform scheme, involving Matlab and Visual Studio is applied. Matlab is used to track the robot arm (augmented reality section). The communication with the robot and the detection of real objects is implemented in Visual Studio (real objects section).

6.1. LOCALIZATION OF REAL OBJECTS

The real objects section is responsible for acquiring images of the environment where the real objects are and sending the necessary commands to the robot.

1. Identification of real objects: acquiring and processing images from the environment in order to identify real objects that correspond to objects to be assembled. Both the orientation and 3D location are obtained.
2. Interaction with real objects: necessary calculations are performed so that the robot moves to positions where real objects are detected, grasps the detected objects and finally leaves them in the assembly using a default position and location.

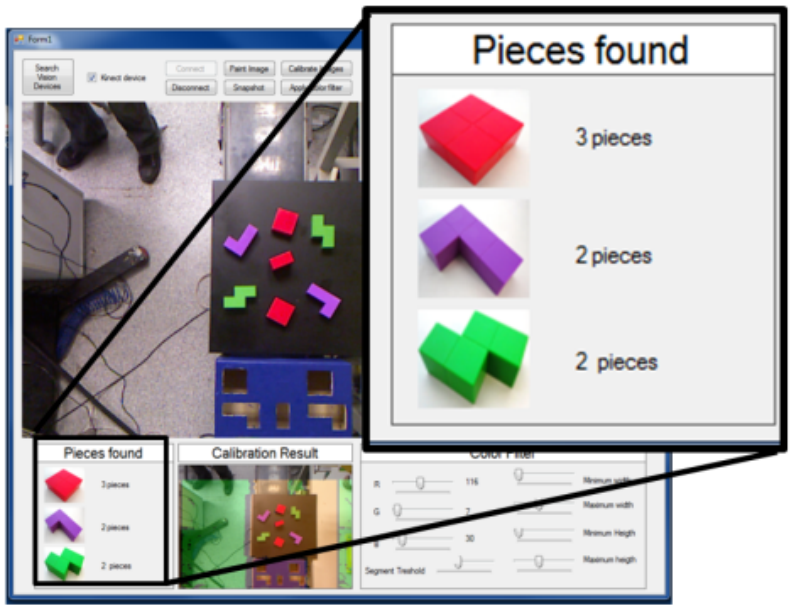

Fig. 14 – Visualization real objects detected.

6.2. HUMAN-ROBOT INTERACTION WITH VIRTUAL OBJECTS

The main functions of this platform are:

1. Tracking joint variables of the robot arm: using a Kalman filter according to the values provided by Kinect-2.
2. Tracking human skeleton: using a library of functions to estimate the location of each body part.
3. Create the augmented reality scene, where real objects and virtual objects are visualized.

The use of virtual parts allows the simulation of assembly operations where the possibility of collision with the environment or with other parts is critical.

The safety standard requires the introduction of guardrails ensuring that humans never invade the workspace of an industrial robot. In this case, the aim is allowing that a human can interact with the robot. Therefore, a security protocol has been established that provides such interaction without risk of serious accidents. The system determines the locations of the human and the robot arm, so that as they approach the robot speed is reduced.

6.3. COMUNICATION WITH THE ROBOT

For this study, a 200iB Fanuc robot has been used which performs the functions of localized assembly of pieces using one of the two available vision systems.

Given that the communication is a critical component of the system, it is performed via an Ethernet network using the RT-Score system. RT-Score [6] is a real time communication system based on the distributed blackboard model, which is able to send and receive messages with ease.

As can be seen in Fig. 13, computers in a distributed system are connected to the internal network. Communication between applications running on these computers and the server is done by applying RT-Score. The management information input to the server is controlled by the "event handler" application. The applications responsible for the two sections send messages to the robot, corresponding to real and virtual parts in the assembly. The event handler, whose main function is to manage the information that the applications send to the robot, assigns priorities to eventually simultaneous messages according to the logic of integration. Once a message has been received by the event handler, it is interpreted and sent to the "software robot" application, which is responsible for executing the corresponding command on the robot control unit. The robot software is a generic robot programming interface that allows to program for different types of robots using XML [5]. This interface allows configuration instructions to be send in order to move and control robot units connected through an Ethernet network.

7. CONCLUSION

A platform that serves as the basis for developing applications which establish interaction between humans and robots has been created.

Using this platform we have carried out a simple case study of interaction between a human being and a robotic system that allows the handle of virtual and real parts between a human and a robot.

A discrete Kalman filter is used to reduce noise in data get it from the low cost vision system which allows tracking robot arms. A human body can be modeled like some interconnected robots. Therefore this method can be extrapolated for tracking human skeletons.

In a future work, we will explore new methods to track articulated objects based on efficient robot models, like screw theory instead of DH model.

Acknowledgments. This work was partially funded by the Universitat Politècnica de València research funds (PAID 2010, Project no. 2566) and by Spanish government and the European Community under the project DPI2010-20814-C02-02 (FEDER-CICYT) and DPI2010-20286 (CICYT).

REFERENCES

1. S. Domínguez, E. Zalama and J. Gomez, *Desarrollo de una cabeza robótica con capacidad de seguimiento visual e identificación de personas*, XXVI, Jornadas de automática, 2005.
2. J. Gonzalez, C. Galindo, J.A. Fernandez, J.L. Blanco, A. Muñoz and V. Arevalo, *La silla robótica SENA. Un enfoque basado en la interacción hombre-máquina*, Revista Iberoamericana de Automática e Informática Industrial, 2008, pp. 38-47.
3. J. Kruger, T.K. Lien and A. Verl, *Cooperation of human and machines in assembly lines*, CIRP Annals-Manufacturing Technology, 2009, pp. 628-646.
4. F. Ramirez, *Avances en interacción hombre-máquina* ECIPERU Revista del encuentro científico internacional, 2010, pp. 29-36.
5. E. Olmos, M. Bosch and A. Sánchez, *Programación de Robots a través de XML*, XXX, Jornadas de Automática, Valladolid, España, 2009.
6. J.L. Posadas, P. Pérez, J.E. Simó, G. Benet and F. Blanes, *Communications Structure for Sensory Data in Mobile Robots*, Engineering Applications of Artificial Intelligence, **15**, *3*, pp. 341-350, 2002.
7. * * *, Prime Sensor™ NITE 1.3 Algorithms notes. http://www.primesense.com.
8. * * *, PrimeSense TM. NITE Controls User Guide. http://www.primesense.com.
9. Burrus, Nicolas, *Kinect calibration*, 2011; URL:http://nicolas.burrus.name/index.php/research/kinectcalibration.
10. * * *, MatLab: http://www.mathworks.es/products/matlab/index.html (November, 2011).
11. Greg Welch and Gary Bishop, *An Introduction to the Kalman Filter*, Design, **7**, *1*, 1-16, 2001.
12. T. Itoh, K. Kosuge, and T. Fukuda, *Human-machine cooperative telemanipulation with motion and force scaling using task-oriented virtual tool dynamics*, IEEE Transactions on Robotics and Automation, **16**, *5*, 505-516, 2000.
13. F. Leishman, O. Horn, and G. Bourhis, *Smart wheelchair control through a deictic approach*, Robotics and Autonomous Systems, **58**, *10*, 1149-1158, 2010.

14. V. Athitsos, and S. Sclaroff, *Inferring Body Pose without Tracking Body Parts*, Proc. Int'l Conf. Computer Vision and Pattern Recognition, 2000.
15. V. Athitsos, and S. Sclaroff, *Estimating 3D Hand Pose from a Cluttered Image*, Proc. Int'l Conf. Computer Vision, 2003.
16. M. Brand, *Shadow Puppetry*, Proc. Int'l Conf. Computer Vision, 1999, pp. 1237-1244.
17. C. Bregler, and J. Malik, *Tracking People with Twists and Exponential Maps*, Proc. Int'l Conf. Computer Vision and Pattern Recognition, 1998, pp. 8-15.
18. N. Howe, M. Leventon, and W. Freeman, *Bayesian Reconstruction of 3D Human Motion from Single-Camera Video*, Neural Information Processing Systems, 1999.
19. G. Mori, and J. Malik, *Estimating Human Body Configurations Using Shape Context Matching*, Proc. European Conf. Computer Vision, **3**, pp. 666-680, 2002.
20. D. Ormoneit, H. Sidenbladh, M. Black, and T. Hastie, *Learning and Tracking Cyclic Human Motion*, Neural Information Processing Systems, 2000, pp. 894-900.
21. V. Pavlovic, J. Rehg, and J. Maccormick, *Learning Switching Linear Models of Human Motion*, Neural Information Processing Systems, 2000, pp. 981-987.
22. G. Shakhnarovich, P. Viola, and T. Darrell, *Fast Pose Estimation with Parameter Sensitive Hashing*, Proc. Int'l Conf. Computer Vision, 2003.
23. H. Sidenbladh, M. Black, and L. Sigal, *Implicit Probabilistic Models of Human Motion for Synthesis and Tracking*, Proc. European Conf. Computer Vision, Vol. 1, 2002.
24. C. Sminchisescu, and B. Triggs, *Kinematic Jump Processes for Monocular 3D Human Tracking*, Proc. Int'l Conf. Computer Vision and Pattern Recognition, June 2003.
25. B. Stenger, A. Thayananthan, P. Torr, and R. Cipolla, *Filtering Using a Tree-Based Estimator*, Proc. Int'l Conf. Computer Vision, 2003.
26. C. Taylor, *Reconstruction of Articulated Objects from Point Correspondences in a Single Uncalibrated Image*, Proc. Int'l Conf. Computer Vision and Pattern Recognition, 2000.
27. K. Toyama, and A. Blake, *Probabilistic Tracking in a Metric Space*, Proc. Int'l Conf. Computer Vision, 2001, pp. 50-59.
28. Magnenat, Stephane, *Imaging information*, 2011, Open Kinect website, URL: http://openkinect.org/wiki/imaginginformation.