Kalman Filter for Tracking Robotic Arms Using low cost 3D Vision Systems

Enrique Martinez Berti
Instituto de Automática e Informática Industrial
Univesitat Politècnica de València
Valencia, Spain
e-mail: enmarbe1@etsii.upv.es

Antonio José Sanchez Salmerón
Instituto de Automática e Informática Industrial
Univesitat Politècnica de València
Valencia, Spain
e-mail: asanchez@isa.upv.es

Francesc Benimeli
Instituto de Automática e Informática Industrial
Univesitat Politècnica de València
Valencia, Spain
e-mail: frabean@ai2.upv.es

***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.**

Keywords—low cost vision system; Kalman filter; augmented reality; kinematics; Human–machine interaction.

I. 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 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.

A. 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-nearestneighbor 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.

B. 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.

C. Organization

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

II. 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.

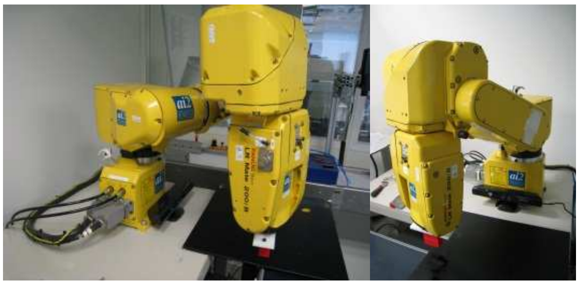

Figure 1. Low Cost 3D Vision System.

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.

III. 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 640x480 pixels. The second sensor has the aim of obtaining depth information corresponding to the objects

ISBN: 978-1-61208-177-9

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.

The calibration process of this camera can be seen in [9]. Calibration is needed to relate both camera and robot coordinates reference systems. Therefore objects located by the camera can be handled by the robot.

IV. 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:

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

2) Prediction of error covariance:

$$P^*_t = AP_{t-1}A^T + Q \tag{2}$$

3) Calculate the constant gain K:

$$K_t = P^*_t H^T \left(HP^*_t H^T + R \right)^{-1} \tag{3}$$

4) Update:

$$\hat{X}_t = \hat{X}^*_t + K_t \left(Z_t - H\hat{X}^*_t \right) \tag{4}$$

5) Update error covariance:

$$P_t = \left(I - K_t H \right) P^*_t \tag{5}$$

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 (1) and (2) the z_d value and covariance is predicted to the next step. Equations (3), (4) and (5) are the equations to correct the discrete Kalman filter. In (3), a new gain of Kalman is calculated. Equations (4) and (5) 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 2 shows a pose estimated during the robot movement. It can be seen the three points detected on the end effector.

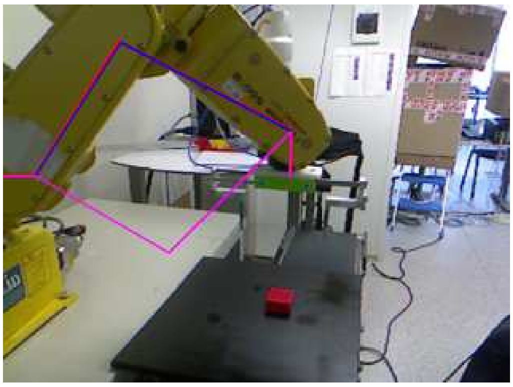

Figure 2. Robot arm pose tracking.

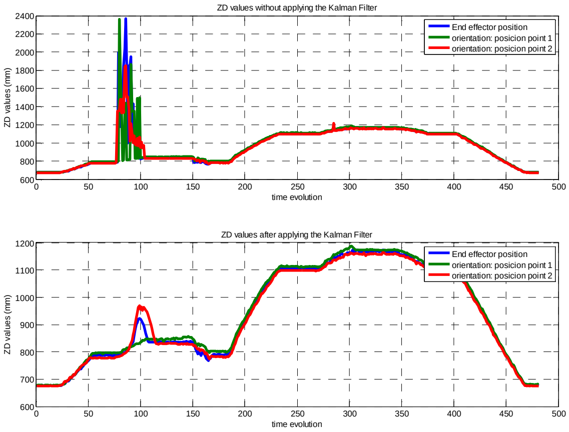

Figure 3. Kalman Filter evolution.

Figure 3 shows the evolution of depth information for comparing results obtained with Kalman filter and without applying the Kalman filter. The first graph shows parameter z_d over time without applying the Kalman filter. The second graph shows z_d over time applying the Kalman filter. At time 100, incorrect z_d 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.

V. ROBOT FANUC 200IB CONTROL

DH (Denavit-Hartemberg) is used to solve direct and inverse kinematics problems. Figure 4 shows coordinate systems and articulation axes used for this Fanuc robot.

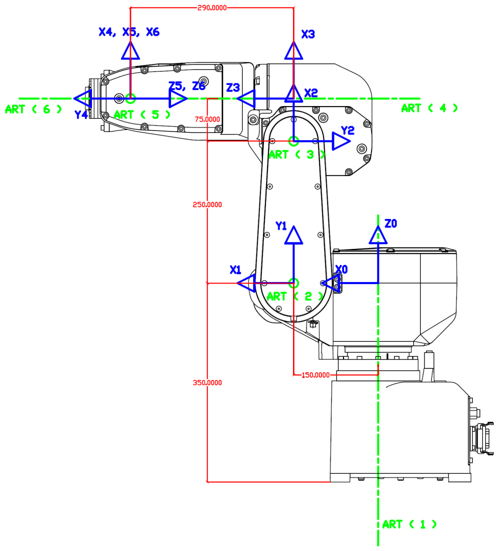

Figure 4. Coordinate systems and articulation axes.

Hence, the following DH parameter table is obtained:

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.

VI. 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. Figure 6 shows the gesture to pick up a virtual object and Figure 7 shows the gesture to place the object in a fixed position.

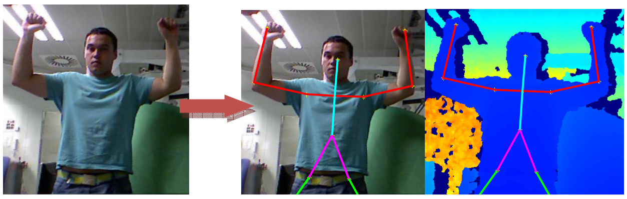

Figure 5. Skeleton pose control.

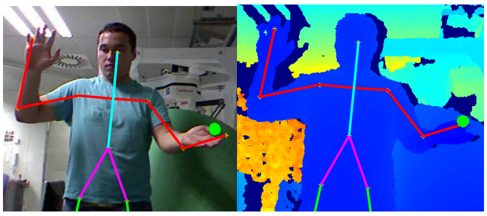

Figure 6. Take object.

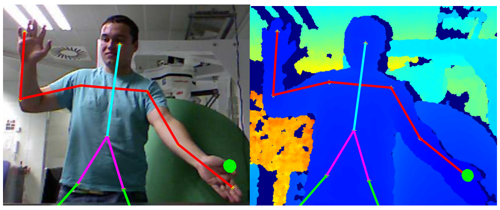

Figure 7. Place object.

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.

VII. 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.

Acknowledgment

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 (2005) “Desarrollo de una cabeza robótica con capacidad de seguimiento visual e identificación de personas” XXVI Jornadas de automática.

[2] J. Gonzalez, C. Galindo, J.A. Fernandez, J.L. Blanco, A. Muñoz and V. Arevalo (2008) “La silla robótica SENA. Un enfoque basado en la interacción hombre-máquina”, Revista Iberoamericana de Automática e Informática Industrial, pp. 38-47.

[3] J. Kruger, T.K. Lien and A. Verl (2009) “Cooperation of human and machines in assembly lines” CIRP Annals-Manufacturing Technology, pp. 628-646.

[4] F. Ramirez (2010) “Avances en interacción hombre-máquina” ECIPERU Revista del encuentro científico internacional pp. 29-36.

[5] E. Olmos, M. Bosch and A. Sánchez (2009) “Programación de Robots a través de XML”, XXX Jornadas de Automática, Valladolid, España.

[6] J.L. Posadas, P. Pérez, J.E. Simó, G. Benet and F. Blanes (2002). “Communications Structure for Sensory Data in Mobile Robots”. Engineering Applications of Artificial Intelligence, vol. 15, no. 3, pp. 341-350.

[7] Prime Sensor™ NITE 1.3 Algorithms notes. http://www.primesense.com.

[8] PrimeSense TM. NITE Controls User Guide. http://www.primesense.com.

[9] E. Martinez Berti, D. Hernandez Campos, and A. Sánchez Salmerón, “Visión 3D de Bajo Coste para la interacción humano-robot utilizando filtro de partículas”. Jornadas de Automática, Sevilla 2011.

[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, n^{o}. 1 (2001): 1-16

[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, n^{o}. 5 (October 2000): 505-516

[13] F. Leishman, O. Horn, and G. Bourhis, “Smart wheelchair control through a deictic approach”, Robotics and Autonomous Systems 58, no. 10 (October 31, 2010): 1149-1158

[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, pp. 1237-1244, 1999.

[17] C. Bregler, and J. Malik, “Tracking People with Twists and Exponential Maps,” Proc. Int’l Conf. Computer Vision and Pattern Recognition, pp. 8-15, 1998.

[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, vol. 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, pp. 894-900, 2000.

[21] V. Pavlovic, J. Rehg, and J. MacCormick, “Learning Switching Linear Models of Human Motion,” Neural Information Processing Systems, pp. 981-987, 2000.

[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, pp. 50-59, 2001.