Conveyor visual tracking using robot vision

Article · April 2006

Rodney G. Roberts
Florida State University

Conveyor Visual Tracking using Robot Vision

Ik Sang Shin[1, 3], Sang-Hyun Nam[2], Hyun Geun Yu[3], Rodney G. Roberts[3] , Seungbin B. Moon[1]

[1]Dept. of Computer Engineering
Sejong Univ., Seoul, Korea
isshin@eng.fsu.edu,
sbmoon@sejong.ac.kr

[2]Chungnam Human Resource
Development, Institute, Korea
shnam28@korcham.net

[3]Dept. of Electrical & Computer Eng.
Florida State Univ.,Tallahassee, FL
rroberts@eng.fsu.edu,
hyun@eng.fsu.edu

ABSTRACT
Robot conveyor tracking is a task in which a robot follows and obtains objects on a conveyor belt. Prior to obtaining an object from an automation line, a robot needs information about the object, such as its position, orientation, velocity, size, etc. Compared to ultrasonic sensors and infrared ray sensors, vision sensors can provide more information about the object on the conveyor belts. Generally, an object tracking system has several steps: obtaining an image, recognizing objects, and extracting information for object position and orientation. The object tracking process for a conveyor system should be fast enough to support a real-time environment. In this article we describe a robotic tracking system that uses vision information extracted from sequential image frames. For fast extraction of vision information we applied a difference image technique which carries out a subtraction operation between two sequential frames with a small time interval. This simple method is useful for obtaining information about an image when there are minimal background effects caused by the conveyor system. In this article we present some results of some experiments using an integrated robot conveyor, vision system, and robot user interface software.

Keywords
Conveyor visual tracking, vision sensor, object recognition

1. INTRODUCTION
Conveyor tracking consists of tracking and catching an object on a conveyor belt using a robot. For catching objects, a robot needs object information such as position, orientation, velocity, size, etc. The recognition of the objects using vision sensors [1] is applicable to areas such as military applications, medical applications, biology, engineering, education, factory operation, etc. The vision sensor is able to efficiently process the information needed to reorganize and track objects on the conveyor belt and adjust some position error of a robot in the middle of assembling electric parts into products by a robot.

Generally the camera can be a single or stereo type for the recognition and tracking of the objects. The former case has the disadvantage that it cannot obtain depth information by itself but has advantages, that is, simple configuration, no correspondence problem caused in each other relation and also its processing is not complicated while in the latter case, it has advantages which can acquire depth information by itself. Currently researchers in this area use a variety of methods for detecting objects in the sequential image frames including block matching, model-based methods, motion-based methods and methods which use color information.

In this article we introduce a method based on analyzing a differential image. Akec [2] made use of calculating pose parameters of a robot from feature points measured in the image plane of a camera for finding object motion. But this method has a disadvantage in that there is a need for prior object information. Luo [3] calculated the motion of a rigid body using optical flow technique. This works well when several objects have differential velocity. Nomura [4] applied a hybrid Kalman filter for calculating the velocity of end-effector and measuring precisely the velocity of a moving object. Papanikolopoulos calculated the motion of objects using SSD (Sum of Squared Differences) optical flow for searching for vision information to track an object on the conveyor [5][6]. Rembold applied a template matching method in his work to calculate position and orientation of objects and a method which estimates continuously the acquisition region of them by a Kalman filter [7].

In this paper, we introduce a system which tracks objects on a conveyor. We performed experiments about conveyor tracking with the method of analyzing a differential image captured from a camera on the end-effector of a robot and obtained information about the objects moving on the conveyor. This is useful in the case when the background effect is small. Previous methods made the full image to convert a binary image but we did only the special part of the image for real-time realization. In conclusion we describe the development environment and robot vision language in Sections 2 and 3. In Section 4 the steps for visual tracking are described and experimental results are shown in Section 5. In Section 6, we present the conclusions.

2. DEVELOPMENT ENVIRONMENT
As shown in Figure 1 the system proposed here consists of a robot system with a 6-axis robot and controller, a vision system with a stand-alone vision processing PC, a camera on the end-effector, a frame grabber on the PC, and a conveyor system which

moves objects.

The operating system for the robot controller is LynxOS, which is operated in real-time. As shown in Figures 2 and 3 the vision system consists of a stand-alone PC for vision processing and a vision sensor attached to a robot arm to improve system open architecture. Figure 2 shows the block diagram of the robot control part and the vision part.

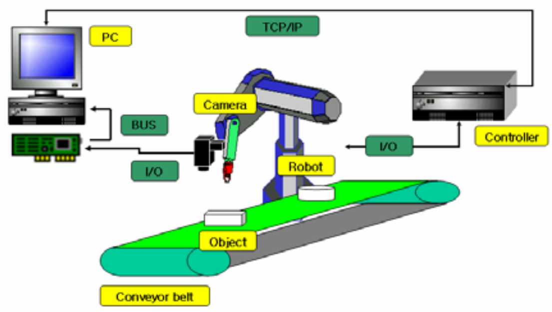


Figure 1. Experiment environment

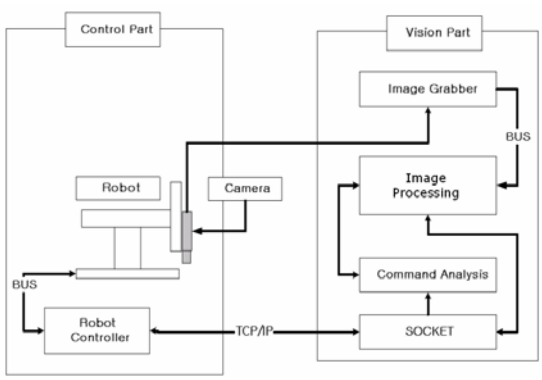


Figure 2. System architecture

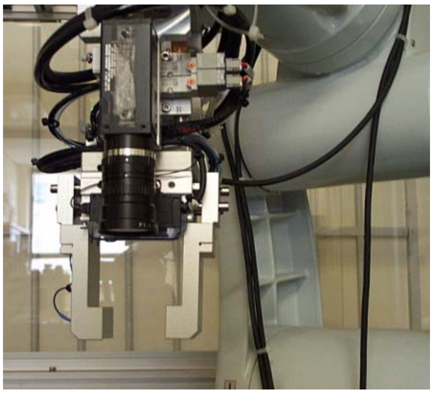


Figure 3. Eye-in-hand vision sensor

As shown in Figure 2, once the robot controller part requests vision processing, the vision system decodes the vision command and performs the service routine and then transmits the result to the robot part through TCP/IP using a communication protocol developed for this project.

3. DEVELOPMENT OF ROBOT VISION LANGUAGE

For designing vision systems more easily, we developed a robot language architecture which is consistent with previous robot vision languages and is able to interface with vision functions efficiently. The main goal of this development is to have a smaller number of commands than previous languages and to interface functions with the arguments of commands so as to support a variety of vision libraries. This architecture handles the current language well and can easily interface with additional libraries later.

As shown in Figure 4, the vision I/F(Interface) language consists of command sets, that is, VINT, VPIC, VCOM, VTRA and VOUT. Also each command set includes many functions. VINIT performs vision initialization such as the image grabber, vision library. VPIC saves images acquired from a camera on memory and shows still images or sequential images to the user. VCOM is in charge of image processing such as preprocessing, object recognition, graphic processing, etc.

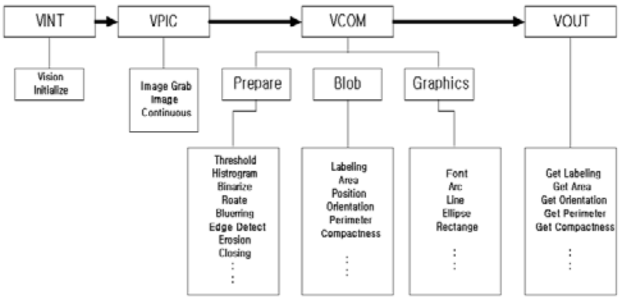


Figure 4. Vision Language

To make the image processing more efficient, unnecessary components are reduced in the preprocessing step, which is composed of Threshold, Histogram, Binaries, etc. [8]. In the object recognition step, object information such as Labeling, Area, and Perimeter are extracted. In the graphic step, object information is properly displayed by characters or figures. Similarly VTRA language is used in calculating conveyor speed, transferring object position to a robot, and starting and pausing vision processing. Finally VOUT is a language which obtains and saves the resulting data to the special variable indicated by the user and computer memory.

4. STEPS FOR CONVEYOR VISUAL TRACKING

Conveyor visual tracking requires two stages: (1) object recognition and (2) object tracking. There are several approaches to perform object recognition. One could use self-organizing maps [9], template matching [10], recognition using color information [11], and the method of using the time difference of two sequential image samples. Self-organizing maps are too slow to be operated with real-time. Template matching requires a priori knowledge of object information to match objects. Unfortunately, due to the required computation load, this method cannot be performed in real time. The method by color information covers the former two disadvantages but cannot be used for binary images. The object recognition method using the difference of two images is useful when the environment does not change quickly over time.

[Reference creating Steps for object recognition]

Step 1: Initialize vision functions.

Step 2: Create binary image.

Step 3: Specify a range which includes objects.

Step 4: If a specific object is pointed among objects probed by user, save the object information.

For object tracking we need to measure object speed, i.e., the speed of the belt. In this work we considered two approaches, one using an encoder and the other using a vision sensor. The former is exactly computable but not only requires the device to be attached to the conveyor system but is inflexible. The vision sensor has additional advantages since it can provide a variety of information such as the area of an object [6], [12].

The equation for the conveyor belt speed is given by

$$V_c = \frac{X(T^k) - X(T^{k-1})}{T^k} \tag{3}$$

where V_c is the speed of an object (conveyor belt), X is the position of an object, T^k is the *k-th* time. The position of the object is defined to be

$$X = (\bar{x}, \bar{y}) \tag{4}$$

where the center of gravity ($\bar{x}$, $\bar{y}$) of the object is given by

$$\bar{x} = \frac{\sum_{i=initial}^{n} \sum_{j=initial}^{m} jB[i,j]}{A} \tag{5}$$

$$\bar{y} = \frac{\sum_{i=initial}^{n} \sum_{j=initial}^{m} iB[i,j]}{A} \tag{6}$$

where *m, n* define the maximum region of an object; *i, j* is the current *x, y* coordinates on the image plane, and initial value is the initial point of each object. The binary image is presented by *B* [*i*,*j*]. The area of each object *A* is given by

$$A = \sum_{i=initial}^{n} \sum_{j=initial}^{m} B[i,j]. \tag{7}$$

By equation (3) the future position of objects is calculated and then sent to the robot controller. Using this information, a robot can acquire an object and according to the user command, move the object to another location.

[Steps for object tracking]

Step 1: Recognize objects entry by calculating the difference of two images.

Step 2: Measure the speed of the object and identify whether it is defined by user or not, If it is ok, go to Step 3; otherwise, go to Step 1 and the robot waits.

Step 3: Send the information to the robot by TCP/IP communication.

Step 4: Conduct object tracking by the robot.

5. RESULTS

As shown in Figure 5, as the conveyor belt moves from left to right, the robot chases a moving object.

Figure 5. A robot and conveyor system

As shown in Figure 6, we designed a GUI consisting of a menu bar, output window, communication mode, object information, etc. As shown in Figure 6, the menu bar has vision libraries and the current image acquired from a camera is displayed on an output window(view port). Additionally, results form the object recognition process and the tracking situation is displayed for user. Also we added a communication mode on the GUI to communicate with the robot, debug, and develop programs conveniently.

Figure 7 shows the object recognition steps for one experiment case. First of all, the Figure 7(a) step shows the initial step, i.e., to initialize the vision functions. Figure 7(b) step is the preprocessing step; the image is converted into a binary image using a threshold.

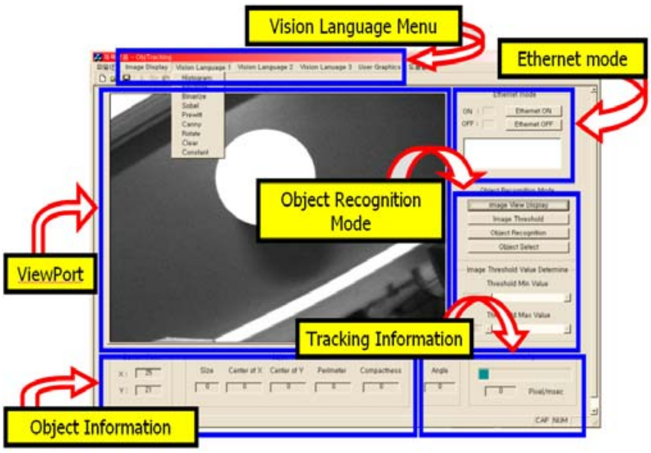
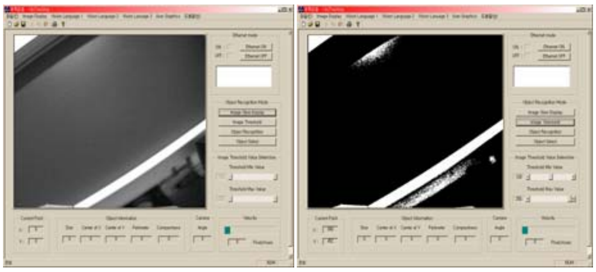


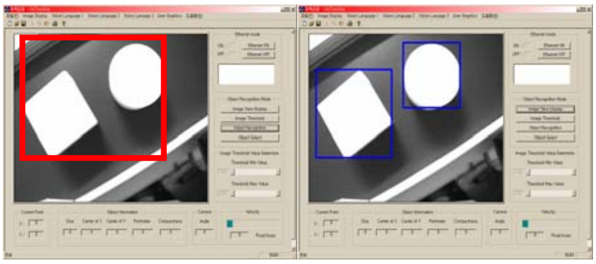
Figure 6. Vision GUI

This step is needed to improve the speed of operation and separate objects from the background. In Figure 7(c) step, the range of objects is chosen in the binarized image and within that range, labels are assigned to each object using a special labeling algorithm. As shown in Figure 7(d) step, the last step is the object recognition stage, e.g., once the user selects an object using the mouse, information (area, position, perimeter, compactness, etc.) about that object is saved to a structure assigned for object information.

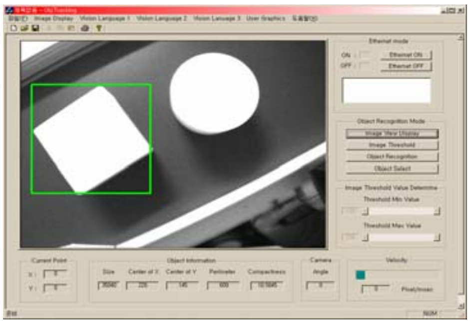
The steps for conveyor visual tracking are shown in Figure 8. Figure 8(a) step shows that the vision system recognizes object entry by calculating the difference of two sequential images and then calls the tracking language. Figure 8(b) step shows that the system measures object speed and decides whether or not the object is identical to the reference object. If the object is matched to the reference, the robot starts to track the object, otherwise the robot remains at its nominal configuration, and the vision system remains in state (a). In the next step, shown in Figure 8(c) step, the system transmits object information to the robot.



(a) **(b)**



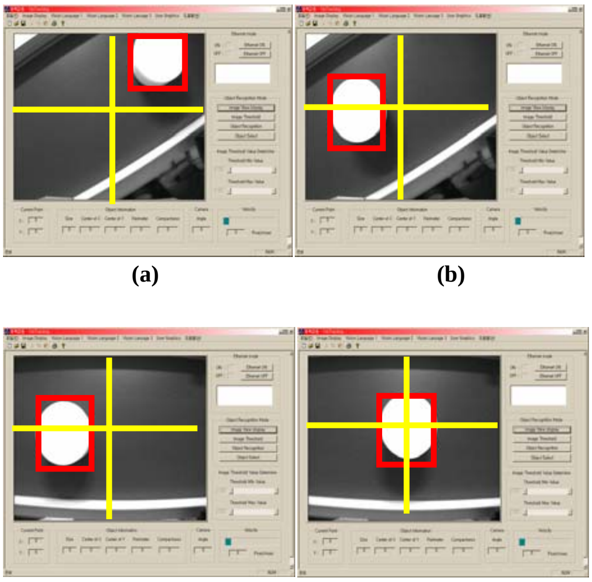
(c)



(d)

Figure 7. Object recognition steps: (a) Vision initialization (b) Image preprocessing (c) Set object range (d) Classify objects

As shown in Figure 8(d) step, the system predicts the next position of the object using the current position and speed and then delivers the information to the robot to track objects with respect to camera-centered position. After this step, the robot catches the object using its gripper and places the object at a point defined by user. After all steps are completed, the robot returns to the first step in which the robot is ready and vision system is initialized.



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

Figure 8. Object tracking steps : (a) Object entry in view of point of a camera (b) Calculate speed and discriminate objects (c) Transmit the information to the robot (d) Object tracking

6. CONCLUSION

In this article, we presented a conveyor system for catching an object. This included the development of a vision language, a vision library, steps for object recognition and tracking for conveyor visual tracking. Object tracking can be applied to security systems, entertainment robots, automated plant operations, etc., but we focused on catching an object in an automated factory. To achieve fast tracking and catching of an object, it needs to decide on an optimal robot trajectory and interception point. For remote control of a robot [13], vision data are transported by Ethernet communication and a differential image method was applied to object recognition for real-time calculation. We believe that these studies will be needed to improve the efficiency of a repetitive task by a robot in an automated line.

7. REFERENCES

[1] Denker, A., Sabanovic, A., and Kaynak, O. Vision-controlled robotic tracking and acquisition. *Intelligent Robots and systems '94. In Proceedings of the IEEE/RSJ/GI nternational Conference on* vol. 3, 12-16 (Sept. 1994) 2000 – 2006.

[2] Akec, J. A. Steiner, S. J., and Stenger, F. An experimental visual feedback control system for tracking applications using a robotic manipulator. *Industrial Electronics Society, 1998. IECON '98. Proceedings of the 24th Annual Conference of the IEEE* vol 2, (31 Aug.-4 Sept. 1998) 1125 – 1130.

[3] Luo, R. C., Mullen, R. E., and Wessell, D. E. An adaptive robotic tracking system using optical flow, *Robotics and Automation, Proceedings, IEEE International Conference* on vol. 1, (1998) 568-573.

[4] Papanikolopoulos, N. P. and Khosla, P. K. Feature based robotic visual tracking of 3-D translational motion, *Proc. of the 30th IEEE Conf Decision and Control*, (1991)1877-1882.

[5] Papanikolopoulos, N. P., Khosla, P.K., and Kanade, T. Visual tracking of a moving target by a camera mounted on a object: a combination of control and vision, *Robotics and Automation, IEEE Trans.*, vol.9 (1993) 14-35.

[6] Rembold, D., Zimmermann, U., Langle, T., and Worn, U. Detection and handling of moving objects, *IEEE IECON* , (1998) 1332-1337.

[7] Chowdhury, M. H. and Little, W. D, Image thresholding techniques, *Communications, Computers, and Signal, Proceedings, IEEE Conference on*, (1995) 585-889.

[8] Kohenen, T. The self-organizing map, *Proc. of the IEEE*, vol.78, No.9. (1998) 1464-1479.

[9] Rembold, D., Zimmermann, U., Langle, T., and Worn, H. Detection and Handling of Moving Object, *Industrial Electronics Society, 1998. IECON '98. Proceedings of the 24th Annual Conference of the IEEE* vol.3, (1998) 1332 - 1337.

[10] Seungmin, B. and Taeyong, K. An implementation of real-time color tracking vision system using a low cost overlay board, *In Proc. of The 14th KACC*, vol. 14, (1999) 443-445.

[11] Sitti, M., Bozma, I., and Denker, A. Visual tracking for moving multiple objects: an integration of vision and control, *Industrial Electronics, 1995, ISIE '95., Proceedings of the IEEE International Sympositon on* vol. 2, (1995) 535-540.

[12] Safaric, R., Jezernik, K., Calkin, D., W., and Parkin, R., M. Telerobot control via Internet, *Industrial Electronics, ISIE '99. Proceedings of the IEEE International Symposium on* vol. 1, (Feb., 1999) 298 – 303.