

Network-based Vision Guidance of Robot for Remote Quality Control

Yongjin (James) Kwon¹, Richard Chiou², Bill Tseng³ and Teresa Wu⁴

*¹Industrial and Information Systems Engineering
Ajou University*

Suwon, South Korea, 443-749

²Applied Engineering Technology

Drexel University

Philadelphia, PA 19104, USA

³Industrial Engineering

The University of Texas at El Paso

El Paso, TX 79968, USA

⁴Industrial Engineering

Arizona State University

Tempe, AZ 85287, USA

1. Introduction

A current trend for manufacturing industry is shorter product life cycle, remote monitoring/control/diagnosis, product miniaturization, high precision, zero-defect manufacturing and information-integrated distributed production systems for enhanced efficiency and product quality (Cohen, 1997; Bennis et al., 2005; Goldin et al., 1998; Goldin et al., 1999; Kwon et al., 2004). In tomorrow's factory, design, manufacturing, quality, and business functions will be fully integrated with the information management network (SME, 2001; Center for Intelligent Maintenance Systems, 2005). This new paradigm is coined with the term, e-manufacturing. In short, "e-manufacturing is a system methodology that enables the manufacturing operations to successfully integrate with the functional objectives of an enterprise through the use of Internet, tether-free (wireless, web, etc.) and predictive technologies" (Koc et al., 2002; Lee, 2003). In fact, the US Integrated Circuit (IC) chip fabrication industries routinely perform remote maintenance and monitoring of production equipment installed in other countries (Iung, 2003; Rooks, 2003). For about the past decades, semiconductor manufacturing industry prognosticators have been predicting that larger wafers will eventually lead the wafer fabrication facilities to become fully automated and that the factories will be operated "lights out", i.e., with no humans in the factory. Those predictions have now become a reality. Intel's wafer fabrication facilities in Chandler, Arizona, USA, are now controlled remotely and humans only go inside the facility to fix the problems. All operators and supervisors now work in a control room, load/unload wafers

through commands issued over an Intranet. Within the e-manufacturing paradigm, e-quality for manufacture (EQM) is a holistic approach to designing and embedding efficient quality control functions into the network-integrated production systems. Though strong emphasis has been given to the application of network-based technologies into comprehensive quality control, challenges remain as to how to improve the overall operational efficiency and how to improve the quality of the product being remotely manufactured. Commensurate with the trends, the authors designed and implemented a network-controllable production system to explore the use of various components including robots, machine vision systems, programmable logic controllers, and sensor networks to address EQM issues (see Fig. 1).

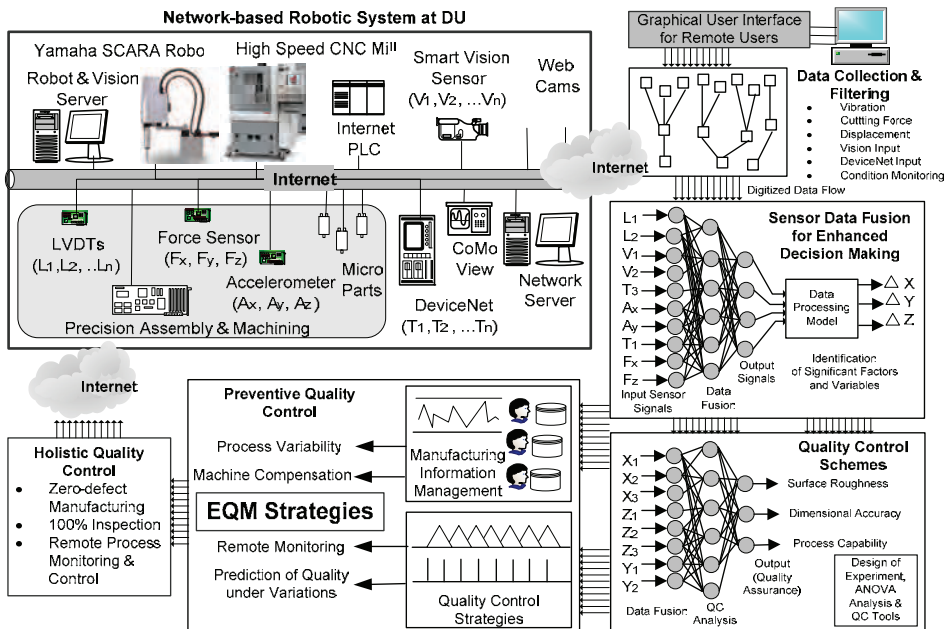


Fig. 1. The proposed concept of EQM within the framework of network-based robotic system developed at Drexel University (DU)

Each component in the system has the access to the network and can be monitored and controlled from a remote site. Such circumstance presents unprecedented benefits to the current production environment for more efficient process control and faster response to any changes. The prototype system implemented enables this research, that is, improving the remote quality control by tackling one of the most common problems in vision-based robotic control (i.e., difficulties associates with vision calibration and subsequent robotic control based on vision input). The machine vision system, robot, and control algorithms are integrated over the network in the form of Application Control Interface (ACI), which effectively controls the motion of robot. Two machine vision systems track moving objects on a conveyor, and send the x, y, z coordinate information to the ACI algorithm that better estimates the position of moving objects with system motion information (speed,

acceleration, etc.). Specifically, the vision cameras capture 2D images (top and side views of part) and combine to get the 3D information of object position. The position accuracy is affected by the distance of parts to the image plane within the camera field of view. The data, such as image processing time and moving speed of the conveyor, can be combined to approximate the position. The study presented in this paper illustrates the benefits of combining e-manufacturing with information-integrated remote quality control techniques. Such concept (i.e., the combination of EQM and e-manufacturing) is novel, and it is based on the prediction that the industry will need an integrated approach to further enhance its production efficiency and to reduce operational costs. Therefore, this study manifests the future direction of e-quality integrated, networked production system, which is becoming a mainstay of global manufacturing corporations.

2. Review of Related Literature

2.1 Technical challenges in network-based systems integrated with EQM

E-manufacturing allows geographically separated users to have their designs evaluated and eventually produced over the network. Using a web browser, a remote operator can program and monitor the production equipment and its motions with visual feedback via the network in real-time. EQM may represent many different concepts in automated production environment, such as 100%, sensor-based online inspection of manufactured goods, network-based process control, rule-based automatic process adjustment, and remote, real-time monitoring of part quality. Industrial automation processes, primarily for pick-and-place operations involving robots, use various sensors to detect the movement of product on a conveyor and guide the robot for subsequent operations. Such system requires precise calibration of each sensors and tracking devices, usually resulting in a very complex and time consuming setup. In Bone and Capson's study (2003), automotive components were assembled using a vision guided robot without the use of conventional fixtures and jigs, which saved the time and cost of production operations. In the study, a host computer was connected through the LAN (local area network) to control the robot, vision system, robot gripper, system control architecture, etc. The coordinates of workcell devices were measured by the vision sensor and the information was transmitted to the robot over the LAN. The loss rate and temporal communication delays commonly occurring in the Internet have been clearly outlined in the study. The same was true in Lal and Onwubolu (2008), where the customized, three tiered web-based manufacturing system was developed. For hardware, a three-axis computer numerically controlled drilling machine was remotely operated and the results showed that a remote user's submission job time was largely dependent on the bandwidth. In fact, the communication time delay problem has been the active research topic since the 1960s, and abundant study materials exist in dealing with delay related problems (Brady & Tarn, 2000). However, the communication linkage between the operator and remote devices are limited by the bandwidth, thus, time-varying delays can only be reduced to some extent. Even a small delay can seriously degrade the intuition of remote operators. Therefore, Brady and Tarn (2000) developed a predictive display system to provide intuitive visual feedback to the operators. Existing industry practices, e.g., network-based factory and office automation usually require a three-layer architecture of information communication, including device-connection layer, equipment-control layer, and information-management layer. The time and cost encountered for setting up the

layered communication system prohibits the broader applications. As a result, Ethernet technologies (e.g., Fast Ethernet and Ethernet Switch) are becoming a mainstay of factory automation networking to replace the traditional industrial networks (Hung et al., 2004). Development of web-based manufacturing system is also abundant in literature, mainly dealing with the integration architecture between devices, equipment, servers, and information networks in distributed shop floor manufacturing environment (Wang & Nace, 2009; Kang et al., 2007; Xu et al., 2005; Wang et al., 2004; Lu & Yih, 2001; Smith & Wright, 1996). Lal and Onwubolu (2007) developed a framework for three-tiered web-based manufacturing system to address the critical issues in handling the data loss and out-of-order delivery of data, including coordinating multi-user access, susceptibility to temporal communication delays, and online security problems. The study well illustrated the problems in Internet-based tele-operation of manufacturing systems. The same problems have been well outlined in Luo et al. (2003), which predicted that the use of Internet in terms of controlling remote robots will ever increase due to its convenience. However, despite the numerous studies conducted by many field experts, solving the delay related problems of the Internet would remain as a challenging task for many years (Wang et al., 2004).

2.2 Calibration for vision-guided robotics in EQM

Technical complications in vision-guided robotics stem from the challenges in how to attain precise alignment of image planes with robot axes, and the calibration of image coordinates against corresponding robot coordinates, which involve expensive measuring instruments and lengthy derivation of complex mathematical relationships (Emilio et al., 2002; Emilio et al., 2003; Bozma & Yal-cin, 2002; Maurício et al., 2001; Mattone et al., 2000; Wilson et al., 2000). Generally speaking, robot calibration refers to the procedure during start-up for establishing the point of reference for each joint, from which all subsequent points are based, or the procedure for measuring and determining robot pose errors to enable the robot controller to compensate for positioning errors (Greenway, 2000; Review of techniques, 1998; Robot calibration, 1998). The latter is a critical process when (1) robots are newly installed and their performance characteristics are unknown, (2) the weight of end-of-arm tooling changes significantly, (3) robots are mounted on a different fixture, and (4) there is a need for robot performance analysis. To position a robot at a desired location with a certain orientation, a chain of homogeneous transformation matrixes that contain the joint and link values mathematically model the geometry and configuration of the robot mechanism. This kinematic modeling, which is stored in a robot controller, assumes the robot mechanism is perfect, hence no deviations are considered between the calculated trajectories and the actual robot coordinates (Greenway, 2000; Review of techniques, 1998). In addressing kinematic modeling, robot calibration research has three main objectives related to robot errors: (1) robot error parameter modeling, (2) a measurement system for collecting pose error measurements, and (3) parameter identification algorithms. Despite extensive research efforts, most calibration systems require complicated mathematical modeling and expensive measuring devices, both of which entail special training, lengthy setups and substantial downtime costs on companies (Greenway, 2000; Robot calibration, 1998; Review of techniques, 1998). Even after errors are mathematically modeled, calibration becomes susceptible to slight changes in setup. Consequently, only a few calibration methods have been practical, simple, economical and quick enough for use with industrial robots (Maurício et al., 2001; Albada et al., 1994; Emilio et al., 2003; Hidalgo & Brunn, 1998; Janocha

& Diewald, 1995; Lin & Lu, 1997; Meng & Zhuang, 2001; Meng & Zhuang, 2007; Young & Pickin, 2000). Another compounding factor is introduced when a robot's position is controlled via a machine vision system. The transformation of camera pixel coordinates into corresponding robot coordinate points requires not only a precise alignment between the robot and vision systems' primary axes, while maintaining a fixed camera focal length, but also a precise mapping between the camera field of view and the robot workspace bounded by the field of view. Slight discrepancies in alignment and boundary delineation usually result in robot positioning errors, which are further inflated by other factors, such as lens distortion effects and inconsistent lighting conditions. Indeed, visual tracking has been the interests of industry and academia for many years, and still an active area of research (Bozma & Yal-cin, 2002). These studies commonly investigate the optimal way of detecting the moving object, separating them from the background, and efficiently extracting information from the images for subsequent operations (Cheng & Jafari, 2008; Bouganis & Shanahan, 2007; Lee et al., 2007; Golnabi & Asadpour, 2007; Tsai et al., 2006; Ling et al., 2005; Stewart, 1999; Yao, 1998). Note vision related studies suffer from technical difficulties in lighting irregularities, optical distortions, calibration, overlapped parts, inseparable features in the image, variations in settings, etc. Even though many years of efforts have been dedicated to the vision research, finding an optimal solution is highly application dependent and no universal model exists in motion tracking.

3. System Setup

At Drexel University, the network-based robotic systems have been under development in the last five years. The aim is to develop robotic, vision, and micro machining systems integrated with sensor networks, which can be accessed and controlled through the Internet. Each equipment has own IP address for network-based data transfer and communication. Some of the constituents include micro/macro-scale robots (Yamaha SCARA YK-150X, YK-220X & YK-250X), a high speed computer numerical control micro milling machine with an Ethernet card (Haas Office Mini CNC Mill), a micro force transducer (Kistler Co.), ultra precision linear variable displacement sensors (LVDTs), Internet programmable logic controllers (DeviceNet Allen Bradley PLCs), a SmartCube network vacuum controller (Festo Co.) for robot end-effector, network computer vision systems (DVT Co.), and a CoMo View remote monitor/controller (Kistler Co.), D-Link DCS-5300 web cameras, network cameras, a BNT 200 video server, and web servers. The SmartImage vision system from DVT Company is Internet-based and self-contained with a lens, a LED ring lighting unit, FrameWork software, and an A/D converter. The camera can be accessed over the network through its IP/Gateway addresses. Any image processing, inspection and quality check can be performed remotely and instant updates on system parameters are possible. The camera contains a communication board with eight I/O ports, which can be hardwired for sending and receiving 24-V signals based on inspection criteria (i.e., Fail, Pass, and Warning). Also, descriptive statistics can be sent over the network in the form of text string using a data link module. The two SmartImage Sensors used are DVT 540 (monochrome) and 542C (color). The first one is a gray-scale CCD camera with a pixel resolution of 640 x 480 and the CCD size being 4.8 x 3.6 mm. This camera is used to provide the height of the moving object for robot's Z-axis control. The 542C camera has a pixel resolution of 640 x 480 with a CCD size of 3.2 x 2.4mm. The 542C is used to find the exact center location of moving objects on a

conveyor and is placed horizontally above the conveyor in the X & Y plane. A Kistler CoMo View Monitor has connectivity with sensors, including a high sensitivity force transducer for micro-scale assembly force monitoring and a LVDT for dimensional accuracy check with one micron repeatability. The CoMo View Monitor contains a web server function with a variety of process monitoring and control menus, enabling Internet-based sensor networks for process control. The Yamaha YK 250X SCARA (selective compliance assembly robot arm) robot is specifically configured to have a high accuracy along the horizontal directions in the form of swing arm motions (Groover 2001). This renders the robot particularly suitable for pick and place or assembly operations. The robot has the repeatability along horizontal planes of ± 0.01 mm (± 0.0004 -in.). For part handling, a variable speed Dorner 6100 conveyor system is connected with robot's I/O device ports in order to synchronize the conveyor with the motion of robot (Fig. 2).

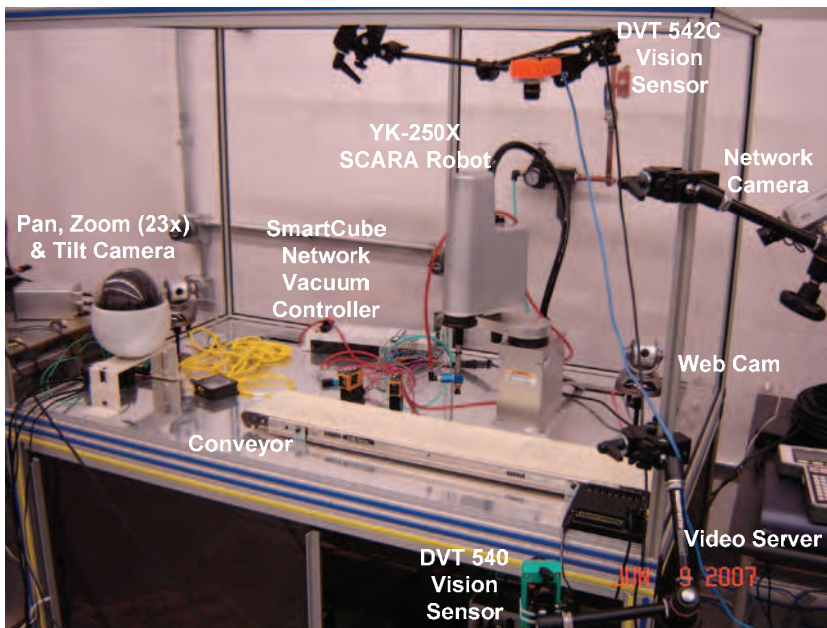


Fig. 2. Experimental setup for network-based, vision-guided robotic system for EQM

The robot's RCX 40 controller is equipped with an onboard Ethernet card, an optional device for connecting the robot controller over the Internet. The communications protocol utilizes TCP/IP (Transmission Control Protocol/Internet Protocol), which is a standard Internet Protocol. PCs with Internet access can exchange data with the robot controller using Telnet, which is a client-server protocol, based on a reliable connection-oriented transport. One drawback to this approach is the lack of auditory/visual communications between the robot and the remotely situated operators. To counter this problem, the Telnet procedure has been included in the Java codes to develop an Application Control Interface (ACI), including windows for the robot control, data, machine vision, and web cameras (Fig. 3).

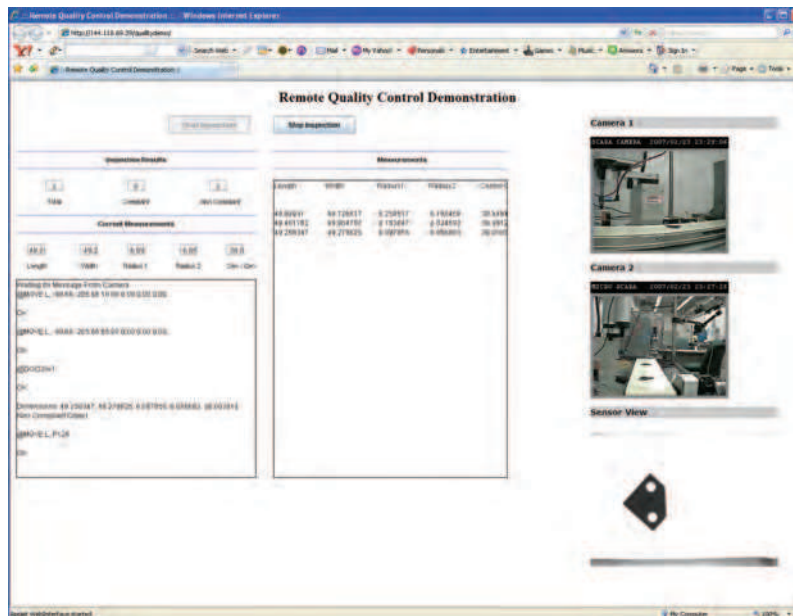


Fig. 3. Web Interface ACI for the remote quality experiment

The basic workings are as follows. The user first tells the ACI to connect to the robot controller. The connection between the application and the robot controller is established by the utilization of Winsock control on port 23 and other control functions that communicate through IP addresses. The robot follows a Telnet type connection sequence. Once connected, the ACI automatically sends the robot to a starting position. The ACI then starts the conveyor belt, which is activated by a digital output from the robot controller. The user establishes a contact with the DVT cameras using DataLink control, and the live images are displayed with the help of DVTSID control. When an object is detected by the ObjectFind SoftSensor in the camera, the x and y coordinates of the object are passed to the ACI from the DVT 542C SmartImage Sensor. The z-coordinate, which is the height of the object, is also passed to the ACI. The robot then moves to the appropriate location, picks up the object and places it in a predefined location, off from the conveyor. The whole process then starts again. The ACI improves not only the visualization of robot operations in the form of an intuitive interface, but also provides enhanced controllability to the operators. The ACI can verify the robot coordinate points, once the robot has been driven to the vision guided locations. The ACI monitors the current robot position, and calculates the shortest approach as the vision sends the part coordinates to the robot. In addition to the web camera, for a high degree of visual and auditory communications, three fixed focus network cameras and one PZT (pan, zoom, and tilt) camera with 23 x optical zoom are connected with a BNT 200 video server, through which remotely located operators can observe the robot operations over the secure web site. This enhanced realism in the simulated environment guarantees the higher reliability of the performance and confidence about the remote operation of the system (Bertoni et al., 2003).

4. Calibration of Vision System

Vision calibration for robotic guidance refers to the procedure for transforming image coordinates into robot Cartesian coordinates (Gonzalez-Galvan et al., 2003; Motta et al., 2001). This procedure is different than the robot calibration, which describes (1) the procedure during start-up for establishing the point of reference for each joint, from which all subsequent points are based, or (2) the procedure for measuring and determining robot pose errors to enable the robot's controllers to compensate for the errors (Bryan, 2000; Review of techniques, 1998; Robot calibration, 1998). Vision calibration is a critical process when (1) robots are newly installed, (2) camera optics and focal length have changed significantly, (3) robots are mounted on a different fixture, and (4) there is a need for vision guidance. To position a robot at a desired location, pixel coordinates from the camera have to be converted into corresponding robot coordinates, which are prone to many technical errors. The difficulties stem from the facts that (1) lens distortion effects, (2) misalignment between image planes and robot axes, and (3) inherent uncertainties related to the defining of image plane boundaries within the robot work space. Precise mathematical mapping of those inaccuracies are generally impractical and computationally extensive to quantify (Amavasai et al., 2005; Andreff et al., 2004; Connolly, 2005; Connolly, 2007). Most calibration procedures require complicated mathematical modeling and expensive measuring devices, both of which entail special training and lengthy setups, hence imposing substantial downtime costs on companies (Bryan, 2000; Robot calibration, 1998; Review of techniques, 1998). Even after mathematical modeling, calibration becomes susceptible to slight changes in setup. Consequently, only a few calibration methods have been practical, simple, economical and quick enough for use with industrial robots (Abderrahim & Whittaker, 2000; Hosek & Bleigh, 2002; Meng & Zhuang, 2001; Meng & Zhuang, 2007; Pena-Cabrera et al., 2005; Perks, 2006; Young & Pickin, 2000; Zhang & Goldberg, 2005; Zhang et al., 2006).

In this context, the methodology developed in this study emulates production environment without the use of highly complicated mathematical calibrations. The image captured by the camera and the robot working space directly over the conveyor are considered as two horizontal planes. Two planes are considered parallel, hence any point on the image plane (denoted as a_i and b_i) can be mapped into the robot coordinates. By operating individual values of a_i and b_i with the scale factors (S_x and S_y), the image coordinates (pixel coordinates) can be translated into the robot coordinates using the following functional relationship (Wilson et al., 2000):

$$f : \mathbf{P}_i \triangleq \mathbf{R}_i + \mathbf{S}_i \cdot \mathbf{v}_i + \boldsymbol{\varepsilon}_i \quad (1)$$

where \mathbf{P}_i = the robot state vector at time i , \mathbf{R}_i = the robot coordinate vector at the origin of the image plane, \mathbf{S}_i = the scale vector with 2×2 block of the form $\begin{bmatrix} S_x & 0 \\ 0 & S_y \end{bmatrix}$,

$\mathbf{v}_i = [a_i, b_i]^T - [a_0, b_0]^T$, a_0 and b_0 = the image coordinate zero, and $\boldsymbol{\varepsilon}_i$ = a zero mean Gaussian error vector due to coordinate mapping. The robot state vector assumes a form:

$$P_i = [x, \dot{x}, y, \dot{y}, z, \dot{z}, \alpha, \dot{\alpha}, \beta, \dot{\beta}, \gamma, \dot{\gamma}]^T = [x_i, y_i, z_i]^T \tag{2}$$

where x, y, z = the translated robot coordinates (mm) from the pixel or image coordinates, α, β, γ = the relative orientation described by roll, pitch, and yaw angles, and $\dot{x}, \dot{y}, \dot{z}, \dot{\alpha}, \dot{\beta}, \dot{\gamma}$ = the relative velocities. Considering the work area as a 2D surface, the scale factors for each axis can be represented as:

$$S_x = \sqrt{\frac{(x_1 - x_r)^2 + (y_1 - y_r)^2}{(a_1 - a_0)^2 + (b_1 - b_0)^2}}; S_y = \sqrt{\frac{(x_2 - x_r)^2 + (y_2 - y_r)^2}{(a_2 - a_0)^2 + (b_2 - b_0)^2}} \tag{3}$$

The goal is to minimize the transformation error caused by lens distortion and other minor misalignments between the planes:

$$\Theta_{\min} \leq \varepsilon_i(x, y) \approx \max[(\hat{P}_i(x) - P_i(x)), (\hat{P}_i(y) - P_i(y))] \leq \Theta_{\max} \tag{4}$$

where P_i = the true location, \hat{P}_i = the observed robot position over the networks, and Θ_{\min} & Θ_{\max} = the preset limits for the magnitude of errors in accordance with the focal length. Vision calibration was conducted by dividing the region captured by the camera into a 4 x 4 grid, and applying separate scaling factors for a better accuracy (Fig. 4). The division of image plane into equally spaced blocks increases the accuracy of the system by countering the problems in (High-Accuracy Positioning System User’s Guide, 2004): (1) the image plane cannot be perfectly aligned with the robot coordinate axes, which is the case in most industrial applications, (2) the perfect alignment requires a host of expensive measuring instruments and a lengthy setup, and (3) the imperfections caused by optics and image distortion. Initially, it was tried with the calibration grid from Edmund Optics Company, which has 1 micron accuracy for solid circles on a precision grid. The circles were, however, too small for the camera at the focal length. Sixteen, solid-circle grid was designed with AutoCAD software and printed on a white paper. The diameter of circle is equivalent to the diameter of the robot end-effector (i.e., vacuum suction adaptor), of which radius being 10.97mm. Once the grid is positioned, the robot end-effector was positioned directly over each circle, and corresponding robot coordinates were recorded from the robot controller. This reading was compared with the image coordinates. For that purpose, the center of each circle was detected first. The center point is defined as:

$$Ctr_x = K^{-1} \cdot \sum_{k=1}^K [Xe_k - Xs_k] \cdot 2^{-1}; Ctr_y = G^{-1} \cdot \sum_{g=1}^G [Ye_g - Ys_g] \cdot 2^{-1} \tag{5}$$

where K and G = the total numbers of pixel rows and columns in the object, respectively, Xe = the x coordinate point for the left most pixel in row k , Xs = the x coordinate point for the right most pixel in row k , Ye = the y coordinate point for a bottom pixel in column g , and Ys = the y coordinate point for a top pixel in column g .

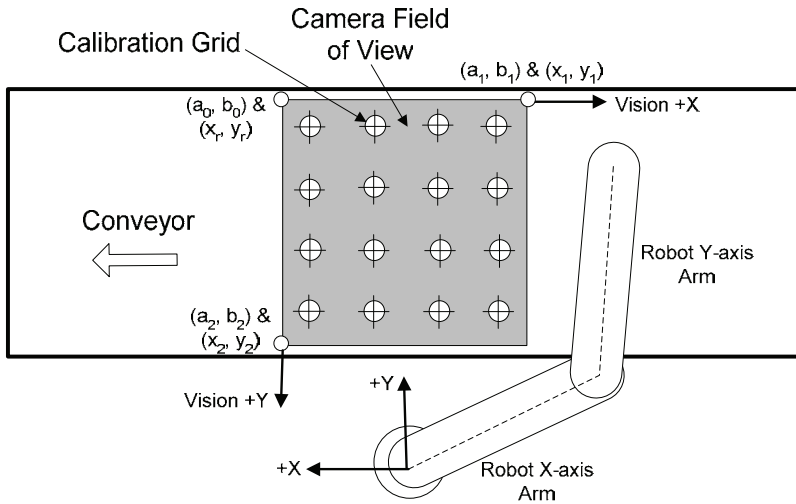


Fig. 4. Schematic of vision calibration grid

The scale factors consider the robot Cartesian coordinates at every intersection of the grid lines. Any point detected within the image plane will be scaled with respect to the increment in the grid from the origin. Let $P(a_i, b_i)$ be the center point of the moving object detected, then the following equations translate it into:

$$\begin{aligned}
 x_i &= x_r + \sum_{n=1}^{p-1} S_{x,n} \cdot |a_n - a_{n-1}| + S_{x,p} [a_i - a_{p-1}] + \varepsilon_x; \\
 y_i &= y_r + \sum_{m=1}^{q-1} S_{y,m} \cdot |b_m - b_{m-1}| + S_{y,q} [b_i - b_{q-1}] + \varepsilon_y,
 \end{aligned}
 \tag{6}$$

where n = the number of columns, m = the number of rows, p and q = the number of grids from the origin where $P(a_i, b_i)$ is located, and ε_x & ε_y = imprecision involved in scaling. In order to capture the moving objects on a conveyor, a series of images is taken at a fixed rate and the time interval between each frame is calculated. The algorithms in the ACI automatically detect the center of moving object and translate that into robot coordinates. The speed of the object is defined as:

$$v_p(mm/s) = \|\mathbf{u} - \mathbf{v}\| \cdot t_f^{-1} = \left[(Ctr_{x,f} - Ctr_{x,f-1})^2 + (Ctr_{y,f} - Ctr_{y,f-1})^2 \right]^{1/2} \cdot t_f^{-1}
 \tag{7}$$

where $\mathbf{u} = (Ctr_{x,f}, Ctr_{y,f})$, the center point of the object at frame no. f , $\mathbf{v} = (Ctr_{x,f-1}, Ctr_{y,f-1})$, the center point of the object at frame no. $f-1$, and t_f = the time taken for a part to travel from frame no. $f-1$ to frame no. f . The time includes not only the part travel between consecutive image frames, but also the lapse for part detection, A/D conversion, image processing, mapping, and data transfer from the camera to a PC. The detection of the moving object, image processing, and calculation of center point are done within the vision system, and the only data transmitted out of the vision system over the networks are the x and y pixel

coordinates. Tests showed no delays in receiving the data over the Internet. The request is sent to the DVT camera by using the syntax: "*object.Connect remoteHost, remotePort.*" The remoteHost is the IP address assigned to the DVT camera and the default remote port is 3246. The DVT DataLink Control is used for the passing of the data from the DVT Smart Image Sensor to the Java application. The data is stored in the string and which is transferred synchronously to the application. DataLink is a built-in tool used to send data out of the system and even receive a limited number of commands from another device. This tool is product-specific, that is, every product has its own DataLink that can be configured depending on the inspection and the SoftSensors being used. DataLink consists of a number of ASCII strings that are created based on information from the SoftSensors: "*<ControlName>.Connect2 (strIPAddress as String, iPort as Integer).*" This is the syntax to connect to the DVT DataLink control.

The speed of the robot as provided by the manufacturer ranges from integer value 1 to 100 as a percentage of the maximum speed (4000 mm/s). The ACI calculates the speed of the moving objects, then adjusts the robot speed. Once a part is detected, a future coordinate point where the part to be picked up, is determined by the ACI. This information is automatically transmitted to the robot controller, and the robot moves to pick up at the designated location. Therefore, the robot travel time to reach the future coordinate must coincide with the time taken by the part to reach the same coordinate. The reach time t_r (ms) is defined in the form of:

$$t_r = \left[\sum_{f=2}^h \|\mathbf{u} - \mathbf{v}\| \right] \cdot \left[\|\mathbf{u} - \mathbf{v}\| \cdot t_f^{-1} \right]^{-1} = \left[(x_i - x_r)^2 + (y_i - y_r)^2 \right]^{1/2} \cdot v_r^{-1} \quad (8)$$

where f = the frame number, indicating the first frame from which the vision system detects the center of moving object, h = the frame number at a pick up location, x_i & y_i = the coordinate of the pick up location, and v_r = the robot speed (mm/s). The discrepancy between the vision generated pick up location and the actual robot coordinate was measured using the following equation:

$$Error(mm) = \left[(Ctr_{x,f} - x_i)^2 + (Ctr_{y,f} - y_i)^2 \right]^{1/2} \quad (9)$$

5. Empirical Verification

To simulate the industrial applications, two different sets of moving speed (20 & 30 mm/s) were used for conveyor, while varying the part heights (65, 55, & 42 mm). To facilitate the testing of robot positioning accuracy, round objects were CNC machined out of aluminum, and its diameter was made intentionally identical to that of a vacuum adaptor. Parts surface was painted matt black in order to accentuate the contrast against the background. Parts were randomly placed on the conveyor and the ACI was initiated to track and guide the robot. The first 12 parts are 65-mm high, then 55 mm, followed by 42-mm parts. Each 36 data points were tested under two speed settings. The height or the z-axis of the object is calculated with the DVT 540 camera, which is kept parallel to the z-axis and perpendicular

to the x-y plane. For x-y coordinates, the DVT 540C camera is positioned above the conveyor, parallel to the robot x and y axes. To facilitate the part pick-up, the robot z-coordinate was set intentionally lower (by 1-mm) than the detected part height from the camera. The vacuum adaptor is made out of compliant rubber material, compressible to provide a better seal between the adaptor and the part surface. Figures 5 and 6 illustrate the errors in x & y plane as well as along the z-direction. Figure 5 shows a slight increase in the magnitude of errors as the part height decreases, while in Figure 6, the error is more evident due to height variation. Such circumstance can be speculated for a number of reasons: (1) possible errors while calibrating image coordinates with robot coordinates; (2) pronounced lens distortional effect; and (3) potential network delay in data transfer.

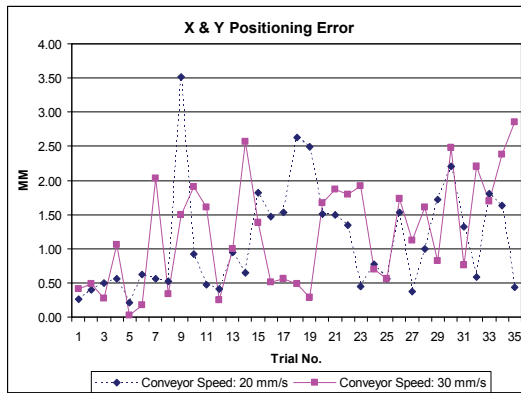


Fig. 5. Errors in X & Y-coordinates (First Trial)

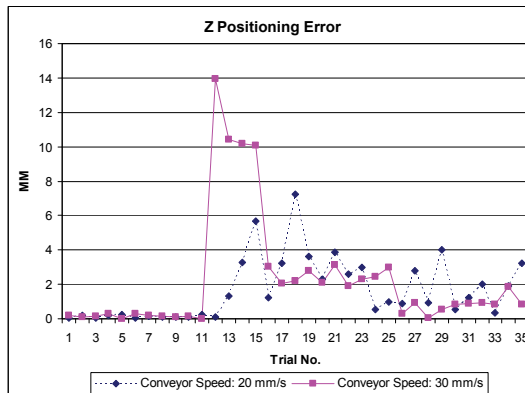


Fig. 6. Errors in Z-coordinates (First Trial)

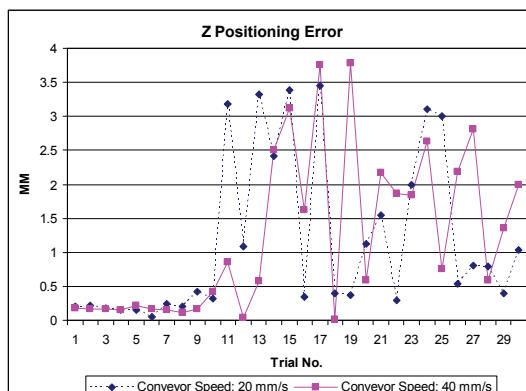


Fig. 7. Errors in Z-coordinates (Second Trial)

To test the speculation, the second set of experiments was conducted. This time, the conveyor speed was increased to 40 mm/sec, in order to observe any speed related fluctuations. The errors in x and y directions show the similar pattern as the first trial. The z directional errors also display a very similar pattern, however, all data points fall below the 4-mm error line (Fig. 7). This indicates that the system acts more stable, compared to the first trial. By observing the two graphs along z direction, it can be stated that the lens distortion effect is more serious in the 540 camera. Other factor might be contributed to the initial location of parts on a conveyor. Based on where the parts are positioned, it can be further from the 540 camera or quite close to the camera lens. Such situation will obviously affect the calibration, and resulted in the larger errors.

6. Experiment on Remote Quality Control

For quality control tasks, the computer vision system is initially trained to learn the profile of the object. The training is done using FrameWork software, which is the firmware of the vision system. Pattern matching techniques are then applied by the firmware to detect the presence of the similar objects under different orientations. Once trained, the camera makes live measurement on the objects passing on the conveyor. Once the object is detected, measurements are made automatically by the vision system. Each inspection and measurement task is defined through soft-sensors (i.e., user defined image processing and analysis algorithms). Figure 8 shows 5 soft-sensors for detecting the object, and extracting its length, width, radii and a center-to-center distance. The 'WorkPiece' is an ObjectFinder soft-sensor, defined by the rectangular box as shown in Fig. 8, which is trained to detect the shape and position of the product. At the core of this system is the Java-based software that integrates all the hardware components and provides the user with an unified view of the system in terms of quality control operation, information exchange (text and video), decision functions, and robot operations. The software is implemented as three separate entities: scripts on the hardware (e.g., vision sensor, robot, and web-cams), the application server and the applet-based client interface. Though the vision camera can be set to detect and measure the objects using the firmware, the task of formatting the measurements and communicating it to the server is done through the built-in Java-based scripting tool. Two

scripts are defined for this purpose. Inter script communication between these two script happens by setting and clearing status flags in the common memory registers. The first script, called the inspection script, is executed after each snapshot.

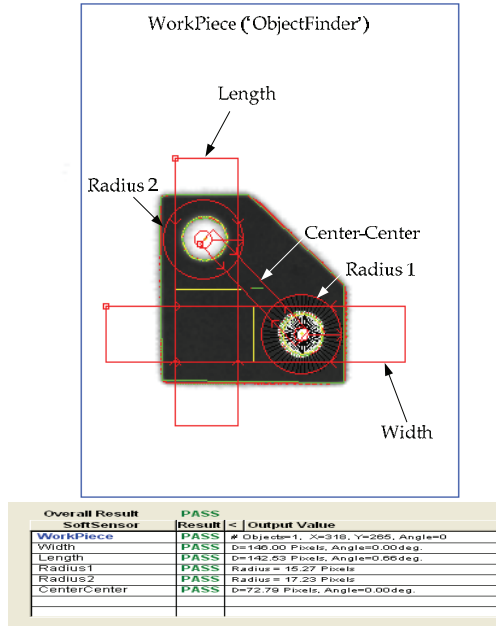


Fig. 8. Configuration of vision soft-sensors for quality measurement

The inspection script checks if an object has been detected after each snapshot and informs the main script of the outcome. If an object is detected, the background script informs this to the robot, which stops the conveyor momentarily. Once an acknowledgement is received from the application server that the belt is halted, the main script sets the 'ready for measurements' flag. The inspection script reads this flag after the next snapshot and makes the necessary measurements. It passes them on to the main script by setting the 'measurement ready' flag and writing the measured values to pre-known register locations. The main script reads these values, parses them to the required format and transmits it to the application server. Once an acknowledgement is received from the application server, it goes back on the detection mode and waits for the next object. The second script, called the main script, runs constantly in the background. It is responsible for establishing and maintaining communications with the application server. It uses the standard Java style *Socket()* function to open a TCP connection to the application server on a dedicated port. Once the connection is established, it exchanges information on this connection with the application server through the *send()* and *recv()* functions. The main program starts by waiting for a TCP connection on a predetermined port from a browser through the *ServerSocket()* class object. Since the end user has to activate the inspection process, this is deemed as the appropriate entry point into the program. Once a connection is accepted through the *accept()* method, an object of *BrowserSession* class, which is a class designed to

handle all the browser communication is initialized with this new connection. This newly initialized object is then used as a parameter to initialize an object of the *Operations* class. The *Operations* class is designed to handle the communications with the computer vision camera and the robot. The *Operations* object implements the Java *Runnable* system class – therefore being able to run as a thread. A thread is a mini-program within a program that can execute in parallel with other threads. The reason behind assigning the browser communications to a separate class and the robot-camera operations to another class should be evident from the way the system is organized. The browser forms the client part of the operation, while the robot and camera constitute the server side of the system, hence this division. Once the initialization of the objects is done, a new thread is started with the newly initialized *Operations* object and the thread is started. This effectively establishes the required connections between the entities in the system and starts operation. The *BrowserSession* class encapsulates all operations pertaining to the communications with the browser. The class provides a *WriteMessage()* method through which information is sent to the browser. The main workhorse of this system is the *Operations* class. It is initialized with an instant of the *BrowserSession* class as an input, which effectively gives it access to the connection with the browser. It then utilizes the *TelnetWrapper* class for communicating with the Telnet daemon on the robot. This class emulates a Telnet client in Java for communications with the robot. An instance of the *Operations* class, once run as a thread, starts off with opening a connection and initializing the robot.

The next step is to establish connection with the camera, through a TCP/IP socket on a pre-assigned port. Once connected, it continuously waits for messages from the camera that include measurements, object position and other status messages. It processes this incoming measurement information to decide on the quality of the product. Once a decision is made, it instructs the robot to perform the necessary action on the inspected object. Figure 3 shows a screen capture of the web-based end-user interface. The applet does not execute any task upon loading. The user initiates the inspection procedure by clicking on the 'Start Inspection' button. Once clicked, it attempts to establish a connection with the application server. Successful establishment of connection also starts the inspection process at the other end as explained earlier. There are several fields in the applet that gives a live analysis of the ongoing inspection process such as a number of objects inspected, a number of compliant objects, current dimension and cumulative measurement history. The browser interface also includes two web camera views and the computer vision sensor's inspection view that show the robotic operation. The web cameras are equipped with an embedded web server, which capture a stream of images over the Internet using the HTTP protocol. The system is tested using the sample objects, of which geometry is shown in Fig. 9.

Six key dimensions: length (L), width (W), diameters of the two circles (D_1 , D_2), horizontal center-center distance between the circles (CC_L), and vertical center-center distance between the circles (CC_W) are shown in Fig. 9. These pieces are machined with a ± 0.25 mm tolerance limit. Pieces whose dimensions lay outside this range are rejected. Twenty five pieces are made, with a few purposely machined out of tolerance limits on each dimension. Several pieces from each type are mixed up and fed through the conveyor belt for inspection. The specifications for the different test pieces used for testing are shown in Table 1. Although this object does not pose any serious measurement challenge, it presents a

moderate complexity for the requirement of our work. Since we concentrate mostly on 2-D measurement, the work-piece thickness is limited to less than 1 mm. Otherwise, a higher thickness would result in problems due to shadows, unless a telecentric lens is used. Shadows would trick the camera to see the object as being elongated on some sides and lead to wrong measurements under the current lighting configuration.

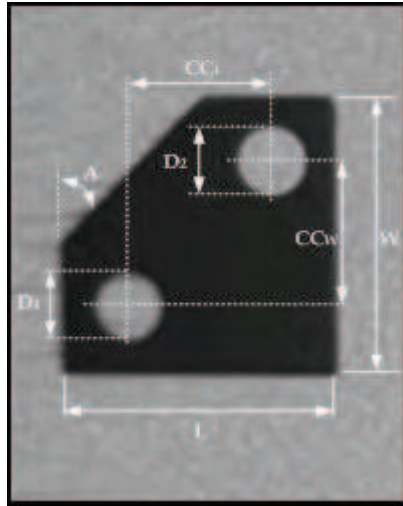


Fig. 9. Test piece geometry

Type	L	W	D ₁	D ₂	CC _L	CC _W	A(angle)	Remarks
1	50	50	12	12	26	26	45	Correct Dimensions
2	52	50	12	12	26	26	45	Wrong Length
3	50	52	12	12	26	26	45	Wrong Width
4	50	50	14	12	26	26	45	Wrong Radius
5	50	50	12	14	26	26	45	Wrong Radius
6	50	50	12	12	28	26	45	Wrong Center-to-Center Horizontal Distance
7	50	50	12	12	26	28	45	Wrong Center-to-Center Vertical Distance
8	50	50	12	12	26	26	47	Wrong Slope

Table 1. Workpiece specifications (all units are in mm except the angular dimensions (degrees))

The camera extracts the first 4 features and the center-to-center distance between the circles. The center-to-center distance is measured instead of the last two quantities (i.e., CC_W and CC_L) since they can be jointly quantified by the center-to-center distance, CC_D. However,

this requires us to develop a tolerance limit, based on which object has to be classified. The tolerance limit for CC_D is determined using the root sum of squares (RSS) approach. CC_D , CC_W , and CC_L have a simple geometrical relationship given by:

$$CC_D = [CC_W^2 + CC_L^2]^{1/2} \quad (10)$$

Since this is a non linear relationship, the non linear RSS extension is employed. The variance of CC_D is defined as:

$$Var(CC_D) = \left[\frac{\partial(CC_D)}{\partial(CC_W)} \right]^2 \cdot Var(CC_W) + \left[\frac{\partial(CC_D)}{\partial(CC_L)} \right]^2 \cdot Var(CC_L) \quad (11)$$

where ∂ = the partial derivative operator, and $Var()$ = the variance of the dimensions. For six sigma quality control, the standard deviation would correspond to the full tolerance limit, hence the following relationship holds:

$$Var(CC_X) = Tol^2(CC_X) \quad (12)$$

where X corresponds to either D,W or L, and Tol() = corresponding tolerance limits. Hence Equation (12) can be written as:

$$Tol(CC_D) = \left[\left(\frac{\partial(CC_D)}{\partial(CC_W)} \right)^2 \cdot Tol^2(CC_W) + \left(\frac{\partial(CC_D)}{\partial(CC_L)} \right)^2 \cdot Tol^2(CC_L) \right]^{1/2} \quad (13)$$

Differentiation Equation 11 with respect to CC_L and CC_W yields:

$$\frac{\partial(CC_D)}{\partial(CC_L)} = -CC_L \cdot [CC_W^2 + CC_L^2]^{-0.5}; \quad \frac{\partial(CC_D)}{\partial(CC_W)} = -CC_W \cdot [CC_W^2 + CC_L^2]^{-0.5} \quad (14)$$

When substituted in (13) results in:

$$Tol(CC_D) = \left[\frac{CC_W^2}{CC_W^2 + CC_L^2} \cdot Tol^2(CC_W) + \frac{CC_L^2}{CC_W^2 + CC_L^2} \cdot Tol^2(CC_L) \right]^{1/2} \quad (15)$$

We substitute the following numerical values from the design specifications: $CC_W = 26$ mm, $CC_L = 26$ mm, and ± 0.25 mm for Tol(CC_W) and Tol(CC_L), then test for a tolerance limit on five dimensions (L, W, D₁, D₂ and CC_D) for quality compliance. The corresponding process control chart during a trial run is given in Fig.10. The CL line corresponds to the mean value, which is calculated to be 36.8 mm. Therefore, the upper control limit (UCL) and lower control limit (LCL) lines correspond to 0.25 mm above and below the mean value. The 4 offsets outside the limits represent the non-compliant objects (Types 6 & 7) that are intentionally mixed up with the compliant ones. Otherwise, it can be seen that the center to center distance measurements follow the expected statistical behavior with a 0.25 mm tolerance limit.

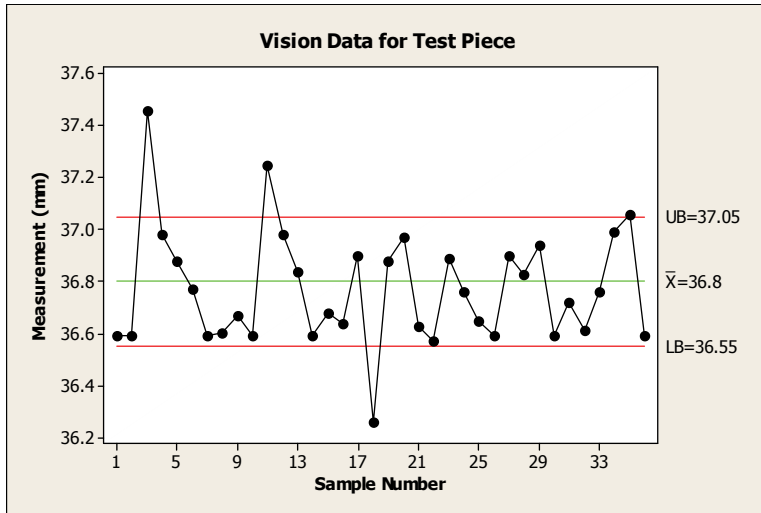


Fig. 10. Statistical process control chart for CC_p

7. Conclusions

This work successfully demonstrates a concept of EQM through the implementation of web-based quality control. A computer vision system measures the dimensions of products on the conveyor belt and reports it to an application server, which activates an appropriate action for the robot. Various image processing and analysis algorithms have been integrated with the ACI for remote quality inspection. Operators can remotely adjust the inspection routine in the case of process changes, such as lighting conditions, part variations, and quality criteria. In the event of changes in moving speed of conveyor or the part spacing, the inspection speed and the inspection region should be adjusted accordingly. The changes are necessary due to varying exposure time of passing parts on a conveyor and the time should be sufficient for the vision system to complete the intended quality control functions. This conforms to the changing environment to ensure the seamless transition between different production settings. One of the most problematic troubles associated with the vision system is the ambient lighting. Sometimes, a precise control of lighting is difficult and consequently, inspection routines may become ineffective. To counter such problem, the parameter settings, such as delay after trigger, exposure time, digitizing time, use of integrated illumination, use of anti-blooming filter, reflectance calibration, field of view balance, product sensor gain, and image area to acquire can be reset remotely. This capability provides less delay in production due to subtle or unexpected changes, which has a great potential, since engineers can access and control the equipment anytime, from anywhere.

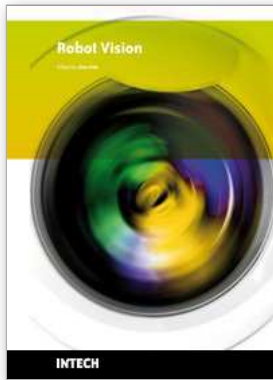
8. References

- Abderrahim, M. & Whittaker, A. R. (2000). "Kinematic model identification of industrial manipulators," *Journal of Robotics and Computer-Integrated Manufacturing*, Vol.16, pp. 1-8.
- Albada, G.D.V., Lagerberg, J.M. & Visser, A. (1994). "Eye in hand robot calibration," *Journal of Industrial Robot*, Vol. 21, No. 6, pp. 14-17.
- Amavasai, B. P., Caparrelli, F. & Selvan, A. (2005). "Machine vision methods for autonomous micro-robotic systems," *Kybernetes*, Vol. 34 No. 9/10, pp. 1421-1439.
- Andreff, N., Renaud, P., Martinet, P. & Pierrot, F. (2004). "Vision-based kinematic calibration of an H4 parallel mechanism: practical accuracies," *Journal of Industrial Robot*, Vol. 31, No. 3, pp. 273-283.
- Bennis, F., Castagliola, P. & Pino, L. (2005). "Statistical Analysis of Geometrical Tolerances: A Case Study," *Journal of Quality Engineering*, Vol. 17, No. 3, pp. 419 - 427.
- Bone, M. G. & Capson, D. (2003). "Vision-guided fixtureless assembly of automotive components," *Robotics and Computer Integrated Manufacturing*, Vol. 19, pp. 79-87.
- Bouganis, A. & Shanahan, M. (2007). "A vision-based intelligent system for packing 2-D irregular shapes," *IEEE Transactions on Automation Science and Engineering*, Vol. 4, No. 3, pp. 382-394.
- Bozma, H.I. & Yal-cin, H. (2002). "Visual processing and classification of items on a moving conveyor: a selective perception approach," *Robotics and Computer Integrated Manufacturing*, Vol. 18, pp. 125-133.
- Brady, K & Tarn, T-J. (2000). "Internet based manufacturing technology: Intelligent remote teleoperation," *Proceedings of the 2000 IEEE/RSJ International Conference on Intelligent Robots and Systems*, pp. 843-848.
- Center for Intelligent Maintenance Systems, 2005, URL: <http://wumrc.engin.umich.edu/ims/?page=home>.
- Cheng, Y. & Jafari, M.A. (2008). "Vision-Based Online Process Control in Manufacturing Applications," *IEEE Transactions on Automation Science and Engineering*, Vol. 5, No. 1, pp. 140 - 153.
- Cohen, A. (1997). *Simulation Based Design*, DARPA Plus-Up Workshop on SBD Alpha Release, (DARPA/TTO Program).
- Connolly, C. (2005). "Vision enabled robotics," *Journal of Industrial Robot*, Vol. 32, No. 6, pp. 456-459.
- Connolly, C. (2007). "A new integrated robot vision system from FANUC Robotics," *Journal of Industrial Robot*, Vol. 34, No. 2, pp. 103-106.
- Goldin, D., Venneri, S. & Noor, A. (1998). "New Frontiers in Engineering," *Mechanical Engineering*, Vol. 120, No. 2, pp. 63-69.
- Goldin, D., Venneri, S. & Noor, A. (1999). "Ready For the Future?" *Mechanical Engineering*, Vol. 121, No. 11, pp. 61-70.
- Golnabi, H. & Asadpour, A. (2007). "Design and application of industrial machine vision systems," *Robotics and Computer Integrated Manufacturing*, Vol. 23, pp. 630-637.
- González-Galván, E.J., Cruz-Ramírez, S.R., Seelinger, M.J. & Cervantes-Sánchez, J.J. (2003). "An efficient multi-camera, multi-target scheme for the three-dimensional control of robots using uncalibrated vision," *Journal of Robotics and Computer-integrated manufacturing*, Vol. 19, No. 5, pp. 387-400.

- Gonzalez-Galvana, E.J., Pazos-Flores, F, Skaarb, S.B. & Cardenas-Galindo, A. (2002). "Camera pan/tilt to eliminate the workspace-size/pixel-resolution tradeoff with camera-space manipulation," *Journal of Robotics and Computer Integrated Manufacturing*, Vol. 18, No. 2, pp. 95-104.
- Greenway, B. (2000). "Robot accuracy," *Journal of Industrial Robot*, Vol. 27, No. 4, pp. 257-265.
- Groover, M.P. (2001). *Automation, Production Systems and Computer Integrated Manufacturing*, 2/e, Prentice Hall, Inc., NJ.
- Hidalgo, F. & Brunn, P. (1998). "Robot metrology and calibration systems - a market review," *Journal of Industrial Robot*, Vol. 25, No. 1, pp. 42-47.
- High-Accuracy Positioning System User's Guide, Version 2.0, Adept Technology, Inc., 2004, URL: <http://www.adept.com/>.
- Hosek, M., & Bleigh, T. (2002). "Brooks automation expands control capability of precision industrial robots," *Journal of Industrial Robot*, Vol. 29, No. 4, pp. 334-348.
- Hung, M-H, Tsai, J., Cheng, F-T & Yang, H-C, (2004). "Development of an Ethernet Equipment Integration Framework for Factory Automation," *Journal of Robotics and Computer-Integrated Manufacturing*, Vol. 20, No. 5, pp. 369-383.
- Janocha, H. & Diewald, B. (1995). "ICAROS: over-all-calibration of industrial robots," *Journal of Industrial Robot*, Vol. 22, No. 3, pp. 15-20.
- Jung, B. (2003). "From remote maintenance to MAS-based e-maintenance of an industrial process," *Journal of Intelligent Manufacturing*, Vol. 14, No. 1, pp. 59-82.
- Kang, Y.G., Wang, Z., Li, R. & Jiang, C. (2007). "A fixture design system for networked manufacturing," *International Journal of Computer Integrated Manufacturing*, Vol. 20, No. 2, pp. 143-159.
- Koc, M., Ni, M.J. & Lee, J. (2002). "Introduction of e-manufacturing," *Proceedings of the International Conference on Frontiers in Design and Manufacturing*, Dalian, China.
- Kwon, Y., Wu, T & Ochoa, J. (2004). "SMWA: A CAD-based decision support system for the efficient design of welding," *Journal of Concurrent Engineering: Research and Applications*, Vol. 12, No. 4, pp. 295-304.
- Lal, S.P. & Onwubolu, G.C. (2008). "E-manufacturing system: from design to performance evaluation," *International Journal of Internet Manufacturing and Services*, Vol. 1, No. 4, pp. 323-344.
- Lal, S.P. & Onwubolu, G.C. (2007). "Three tiered web-based manufacturing system-Part 1: System development," *Robotics and Computer-Integrated Manufacturing*, Vol. 23, pp. 138-151.
- Lee, K-M, Li, Q. & Daley, W. (2007). "Effects of classification methods on color-based feature detection with food processing applications," *IEEE Transactions on Automation Science and Engineering*, Vol. 4, No. 1, pp. 40-51.
- Lee, J. (2003). "E-manufacturing—fundamental, tools, and transformation," *Journal of Robotics and Computer-Integrated Manufacturing*, Vol. 19, No. 6, pp. 501-507.
- Lin, G.C.I. & Lu, T-F (1997). "Vision and force/torque sensing for calibration of industrial robots," *Journal of Industrial Robot*, Vol. 24, No. 6, pp. 440-445.
- Ling, D.S.H., Hsu, H-Y, Lin, G.C.I. & Lee, S-H (2005). "Enhanced image-based coordinate measurement using a super-resolution method," *Robotics and Computer-Integrated Manufacturing*, Vol. 21, pp. 579-588.

- Luo, R.C., Su, K.L., Shen, S.H. & Tsai, K.H. (2003). "Networked intelligent robots through the Internet: issues and opportunities," *Proceedings of the IEEE*, Vol. 91, No. 3, pp. 371-382.
- Lu, T-P & Yih, Y. (2001). "An agent-based production control framework for multi-line collaborative manufacturing," *International Journal of Production Research*, Vol. 39, No. 10, pp. 2155-2176.
- Mattone, R., Campagiorni, G. & Galati, F. (2000). "Sorting of items on a moving conveyor belt. Part 1: a technique for detecting and classifying objects," *Journal of Robotics and Computer Integrated Manufacturing*, Vol. 16, pp. 73-80.
- Maurício, J, Motta, S.T., de Carvalho, G.C. & McMaster, R.S. (2001). "Robot calibration using a 3D vision-based measurement system with a single camera," *Journal of Robotics and Computer-integrated manufacturing*, Vol. 17, No. 6, pp. 487-497.
- Meng, Y. & Zhuang, H. (2001). "Self-calibration of camera-equipped robot manipulators," *Journal of Robotics Research*, Vol. 20, No. 11, pp. 909-921.
- Meng, Y. & Zhuang, H. (2007). "Autonomous robot calibration using vision technology," *Journal of Robotics and Computer-Integrated Manufacturing*, Vol. 23, pp. 436-446.
- Pena-Cabrera, M., Lopez-Juarez, I., Rios-Cabrera, R. & Corona-Castuera, J. (2005). "Machine vision approach for robotic assembly," *Assembly Automation*, Vol. 25, No. 3, pp. 204-216.
- Perks, A. (2006). "Advanced vision guided robotics provide "future-proof" flexible automation," *Assembly Automation*, Vol. 26, No. 3, pp. 216-220.
- Review of techniques, Robotics and Automation Research at UWA, December 1998, URL: <http://www.mech.uwa.edu.au/jpt/CalibrationPages/Menu1.htm>.
- Robot calibration, Robotics and Automation Research at UWA, December 1998, URL: <http://www.mech.uwa.edu.au/jpt/calibration.html>.
- Rooks, B. (2003). "Machine tending in the modern age," *International Journal of Industrial Robot*, Vol. 30, No. 4, pp. 313-318.
- SME Manufacturing Engineering - Tomorrow's Factory: Manufacturing Industry Takes First Steps Toward Implementing Collaborative E-Manufacturing Systems, pp. 43-60, Nov. 2001.
- Smith, C.S. & Wright, P.K. (1996). "CyberCut: A World Wide Web based design-to-fabrication tool," *Journal of Manufacturing Systems*, Vol. 15, No. 6, pp. 432-442.
- Stewart, C.V. (1999). "Robust parameter estimation in computer vision," *SIAM Review*, Society for Industrial and Applied Mathematics, Vol. 41, No. 3, pp. 513-537.
- Tsai, M.J., Hwung, J.H., Lu, T-F & Hsu, H-Y, (2006). "Recognition of quadratic surface of revolution using a robotic vision system," *Robotics and Computer-Integrated Manufacturing*, Vol. 22, pp. 134-143.
- Wilson, W.J., Hulls, C.C. & Janabi-Sharifi, F. (2000). "Chapter 13. Robust image processing and position-based visual servoing," appeared in "*Robust Vision for Vision-Based Control of Motion*", edited by Markus Vincze and Gregory D. Hager," IEEE Press, 3 Park Avenue, 17th Floor, New York, NY.
- Wang, L., Orban, P., Cunningham, A. & Lang, S. (2004). "Remote real-time CNC machining for web-based manufacturing," *Robotics and Computer-Integrated Manufacturing*, Vol. 20, pp. 563-571.

- Wang, L., Shen, W. & Lang, S. (2004). "Wise-ShopFloor: a Web-based and sensor-driven e-Shop Floor," *ASME Journal of Computing and Information Science in Engineering*, Vol. 4, pp. 56-60.
- Wang, L. & Nace, A. (2009). "A sensor-driven approach to Web-based machining," *Journal of Intelligent Manufacturing*, Vol. 20, No. 1, pp. 1-14.
- Xu, Y., Song, R., Korba, L., Wang, L., Shen, W. & Lang, S. (2005). "Distributed device networks with security constraints," *IEEE Transactions on Industrial Informatics*, Vol. 1, No. 4, pp. 217-225.
- Yao, J. (1998). "A New Method for Passive Location Estimation from Image Sequence Using Adaptive Extended Kalman Filter," *Proceedings of ICSP '98*, pp. 1002-1005.
- Young, K. & Pickin, C.G. (2000). "Accuracy assessment of the modern industrial robot," *Journal of Industrial Robot*, Vol. 27, No. 6, pp. 427-436.
- Zhang, M. T. & Goldberg, K. (2005). "Fixture-based industrial robot calibration for silicon wafer handling," *Journal of Industrial Robot*, Vol. 32, No. 1, pp. 43-48.
- Zhang, M., Tao, W., Fisher, W. & Tarn, T.-J. (2006). "An industrial solution to automatic robot calibration and workpiece pose estimation for semiconductor and gene-chip microarray fabrication," *Journal of Industrial Robot*, Vol. 33, No. 2, pp. 88-96.



Robot Vision

Edited by Ales Ude

ISBN 978-953-307-077-3

Hard cover, 614 pages

Publisher InTech

Published online 01, March, 2010

Published in print edition March, 2010

The purpose of robot vision is to enable robots to perceive the external world in order to perform a large range of tasks such as navigation, visual servoing for object tracking and manipulation, object recognition and categorization, surveillance, and higher-level decision-making. Among different perceptual modalities, vision is arguably the most important one. It is therefore an essential building block of a cognitive robot. This book presents a snapshot of the wide variety of work in robot vision that is currently going on in different parts of the world.

How to reference

In order to correctly reference this scholarly work, feel free to copy and paste the following:

Yongjin (James) Kwon, Richard Chiou, Bill Tseng and Teresa Wu (2010). Network-based Vision Guidance of Robot for Remote Quality Control, Robot Vision, Ales Ude (Ed.), ISBN: 978-953-307-077-3, InTech, Available from: <http://www.intechopen.com/books/robot-vision/network-based-vision-guidance-of-robot-for-remote-quality-control>

INTECH
open science | open minds

InTech Europe

University Campus STeP Ri
Slavka Krautzeka 83/A
51000 Rijeka, Croatia
Phone: +385 (51) 770 447
Fax: +385 (51) 686 166
www.intechopen.com

InTech China

Unit 405, Office Block, Hotel Equatorial Shanghai
No.65, Yan An Road (West), Shanghai, 200040, China
中国上海市延安西路65号上海国际贵都大饭店办公楼405单元
Phone: +86-21-62489820
Fax: +86-21-62489821

© 2010 The Author(s). Licensee IntechOpen. This chapter is distributed under the terms of the [Creative Commons Attribution-NonCommercial-ShareAlike-3.0 License](#), which permits use, distribution and reproduction for non-commercial purposes, provided the original is properly cited and derivative works building on this content are distributed under the same license.