Optimized Sensor Placement for Active Visual Inspection

Christopher C. Yang*
Department of Systems Engineering and Engineering Management
The Chinese University of Hong Kong
Shatin, Hong Kong SAR
e-mail: yang@se.cuhk.edu.hk

Frank W. Ciarallo
Department of System and Industrial Engineering
The University of Arizona
Tucson, AZ 85721

Received February 14, 2000; revised August 1, 2000; accepted September 12, 2000

This article presents an optimized sensor planning system for active visual inspection of three-dimensional manufacturing computer-aided design (CAD) models. Quantization errors and displacement errors are inevitable in active visual inspection. To obtain high accuracy for dimensioning the entities of three-dimensional CAD models, minimization of these errors is essential. Spatial *quantization errors* result in digitization. The errors are serious when the size of the pixel is significant compared to the allowable tolerance in the object dimension on the image. In placing the active sensor to perform inspection, displacement of the sensors in orientation and location is common. The difference between observed dimensions obtained by the displaced sensor and the actual dimensions is defined as *displacement errors*. The density functions of quantization errors and displacement errors depend on camera resolution and camera locations and orientations. The sensor constraints, such as resolution, focus, field-of-view, and visibility constraints, restrict sensor placement. To obtain a satisfactory view of the targeted entities of the CAD models, these constraints have to be satisfied. In this article, we focus on the edge line segments as the inspected

* To whom all correspondence should be addressed.

entities. We use a genetic algorithm to minimize the probabilistic magnitude of the errors subject to the sensor constraints. Since the objective functions and constraint functions are both complicated and nonlinear, traditional nonlinear programming may not be efficient and it may trap at a local minimum. Using crossover operations, mutation operations, and the stochastic selection in the genetic algorithm, trapping can be avoided. Experiments are conducted and the performance of the genetic algorithm is presented. Given the CAD model and the entities to be inspected, the active visual inspection planning system obtains the sensor setting that maximizes the probabilities of a required accuracy for each entity.

1. INTRODUCTION

1.1. Sensor Planning

In sensor planning, the objective is determining the appropriate sensor settings automatically when the requirements of the task are given. In most of the current robot controlled vision systems, substantial human involvement using laborious and time-consuming techniques is still required. A typical trial-and-error approach is commonly used, where a sensor setting is selected, tested, and modified if the requirement is not satisfied. Such a process is not cost-effective and lacks flexibility. An automated vision sensor planning system, which uses the available prior knowledge to generate a sensor setting satisfying the task requirement, is desired.

Recently, several automated vision sensor planning approaches have been proposed. These approaches can be divided into three categories, (1) generate-and-test approach, (2) synthesis approach, and (3) expert system approach.

The *generate-and-test* approach involves two steps. In the first step, a geodesic dome is created around a target object and the spherical surface is tesselated. In the second step, sensor settings are generated on the tesselated spherical surface and evaluated according to the task criteria. The final sensor setting is the one that has the best evaluation. Sakane et al.[1–3] developed the HEAVEN system. The prior chosen distance based on the resolution constraint is set as the radius of the sphere. By projecting a ray from the target point of the object to the center of each facet of the spherical dome, the visibility is determined. Each occlusion-free region is then ranked by a distance method criterion. Shadow casting is also considered in camera placement, where the reliability criterion and the detectability criterion are investigated. Sakane, Niepold, and co-workers[4,5] also developed the vision illumination object system. In addition to the planning of camera placement, the planning of the point light source is also investigated. The goodness of the camera and illuminator settings is evaluated by comparison of the expected scene and the image feature attributes obtained, such as edge visibility and contrast. Yi et al.[6] developed the illumination control expert system. Edge visibility, evaluated based on aspect graph representation, is considered in the camera placement planning. Sensor viewpoint is accessed based on the ratio of the unoccluded portion of the edge to the actual edge length. The generate-and-test approach works well for point features; however, it has limitations when extended to other features such as lines and planes. The computation cost is high and only approximation for each viewpoint in the center of each facet is used. Therefore, it is not cost-effective. Besides, field-of-view and focus constraints are not considered.

The *synthesis* approach models the sensor constraints as analytic functions and treats the sensor planning as a constraint satisfaction problem. The drawback of the synthesis approach is the high dimension of the analytic functions. The generalized viewpoint includes 3 degrees of freedom in camera location, 2 degrees of freedom in camera orientation, and the optical parameters of focal length, aperture, and focus. Given the eight-dimensional function spaces and the nonlinear constraint functions, the efficiency in obtaining a solution is low. Cowan and Kovesi[7,8] reduced the eight-dimensional spaces into three-dimensional spaces by intersecting admissible domains of individual constraints. However, some of the parameters are determined separately or not planned. Tarabanis et al.[9–12] developed the machine vision planner. Given the eight-dimensional function spaces, Tarabanis et al. defined the optimization function as a weighted sum of the constraint functions. The weights are selected based on the constraint magnitude.

The *expert system* approach suggests possible lighting and viewing settings based on the rules and the information collected from users by asking a series of questions. Batchelor[13,14] developed the lighting advisor, which asks users questions such as

an object's reflectance characteristic and type of features. Rules are then fired to control the illumination of the inspection cell. Other expert systems were also developed by Kitamura et al.[15] and Novini.[16]

1.2. Active Vision Errors

Errors are inherent in active vision, and they affect the accuracy of inspection. For example, the potential errors are quantization errors, displacement errors, illumination errors, parallax, and sensor motions. Illumination errors, parallax, and sensor motions can be minimized to a negligible level by careful design and control of the environment. However, quantization errors and displacement errors are inevitable.

In image quantization, the quantized sample indicates itself as a part of the object image if and only if the segment covers more than half of it. Such distortion produces the spatial quantization error. Kamgar-Parsi and Kamgar-Parsi[17] developed mathematical tools for computing the average error. Blostein and Huang[18] analyzed the effect of image plane quantization on the three-dimensional point position error. Ho[19] expressed the digitizing error for various geometrical features in terms of the dimensionless perimeter of the object. Griffin and Villalobos[20] discussed an approach to integrate the errors inherent in the visual inspection to determine the sensor capability for inspecting a specified part dimension using binary images.

The displacement of the active head produces distortion of the projected object on the image. The difference between the observed dimensions and the actual dimensions is the displacement error in active visual inspection. Su and Lee[21] presented a methodology for manipulating and propagating spatial uncertainties in a robotic assembly system. Menq and Bohm[22] and Renders et al.[23] presented a framework to characterize the distribution of the position errors of robot manipulators over the work space and study their statistical properties. Chen and Chao[24] identified and parameterized the sources that contribute to the position error and estimated the values of the parameters. Veitschegger and Wu[25] presented a calibration algorithm to find the values of the independent kinematic errors. Bryston[26] developed a methodology for the measurement and characterization of the static distortion of the position data from a three-dimensional tracker. Smith and Chessman[27] presented a method for explicitly representing and manipulating the uncertainty associated with the transformation between coordinate frames representing the relative locations of objects. Yang et al.[28–30] examined both the quantization errors and displacement errors in active vision inspection and derived the density functions of the total error.

1.3. Sensor Planning for Active Visual Inspection

Beside the active vision errors, the sensor settings (or sensor configurations) are affected by the sensor constraints, such as resolution constraints, focus constraints, field-of-view constraints, and visibility constraints. All of the sensor settings that are to be used as part of an inspection plan must satisfy these constraints. Sensor planning is defined as an automatic process to generate appropriate sensor settings based on prior knowledge, such as the required geometric and physical information of objects extracted from computer-aided design (CAD)/computer-aided manufacturing (CAM) models. Cowan and Kovesi,[7,32,33] Tarabanis et al.,[9,10,11,34] and Yang and Marefat[35] examined these constraints and proposed algorithms to obtain potential sensor settings.

To develop a robust active visual inspection planning system, the desire is to minimize the active vision errors (maximize inspection accuracy) subject to the sensor constraints. In this article, we propose using the genetic algorithm to perform the optimization. Genetic algorithms have shown excellent performance in optimization with complicated and nonlinear objective and constraint functions. Compared to traditional nonlinear programming,[36] it is more efficient and trapping at a local minimum is avoided.

Figure 1 depicts the active visual inspection planning system. The structure of the CAD model, the inspection entities identified $\{e_1, e_2, \ldots, e_n\}$, and the minimum required inspection accuracy are the input of the system. The output of the planning process is the planned sensor settings for the set of inspected entities. For example, sensor setting 1

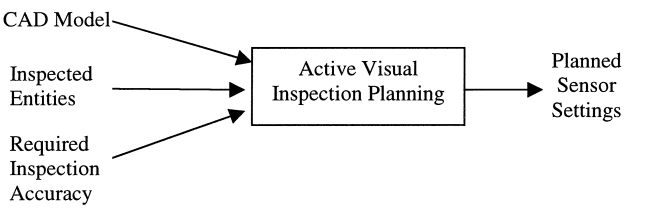

Figure 1. Active visual inspection planning system.

(ss1) is planned for inspecting $\{e_1, e_2\}$, sensor setting 2 (ss2) is planned for inspecting $\{e_1, e_4, e_5\}$, and so on. Each sensor setting is represented by its 6 degrees of freedom in orientation and location.

2. ACTIVE VISION ERRORS

Without careful control, quantization errors and displacement errors can produce significant measurement errors in active vision inspection. For example, a sensor placed very close to an inspected edge segment has high resolution on the image plane; however, the displacement errors from this setting may be too large.

2.1. Quantization Errors

Spatial quantization errors affect the accuracy of inspection seriously, when the size of the pixel is significant compared to the allowable tolerance in object dimension on the image. In digitization, a quantized sample indicates itself as part of the object image if and only if the edge segment covers more than half of the pixel. Using traditional edge detection, a point in the image can only be located up to 1 pixel of accuracy. Figure 2 shows a line segment on a two-dimensional array of pixels.

The density functions of the horizontal and vertical quantization errors, ε_{q_x} and ε_{q_y} are derived as[29,30]:

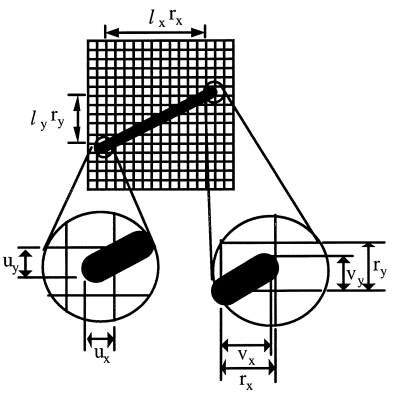

Figure 2. (a) Probability density function of the horizontal quantization errors. (b) Probability density function of the vertical quantization errors.

For the x direction,

$$f_{\varepsilon_{q_x}}(\varepsilon_{q_x}) = \begin{cases} -\dfrac{1}{r_x^2}\varepsilon_{q_x} + \dfrac{1}{r_x} & 0 \le \varepsilon_{q_x} \le r_x \\ \dfrac{1}{r_x^2}\varepsilon_{q_x} + \dfrac{1}{r_x} & -r_x \le \varepsilon_{q_x} \le 0 \\ 0 & \text{else} \end{cases} \tag{1}$$

where r_x is the horizontal pixel size.

For the y direction,

$$f_{\varepsilon_{q_y}}(\varepsilon_{q_y}) = \begin{cases} -\dfrac{1}{r_y^2}\varepsilon_{q_y} + \dfrac{1}{r_y} & 0 \le \varepsilon_{q_y} \le r_y \\ \dfrac{1}{r_y^2}\varepsilon_{q_y} + \dfrac{1}{r_y} & -r_y \le \varepsilon_{q_y} \le 0 \\ 0 & \text{else} \end{cases} \tag{2}$$

where r_y is the vertical pixel size.

Figure 3(a) is the probability density function of the quantization errors where the maximum and minimum errors are $+r_x$ and $-r_x$. Figure 3(b) is the probability density function of the quantization errors where the maximum and minimum errors are $+r_y$ and $-r_y$.

Using a geometric approximation, the density function of the two-dimensional quantization errors (ε_q) can be derived from Eqs. (1) and (2) as follows:

$$f_{\tilde{\varepsilon}_q}(\tilde{\varepsilon}_q) = \frac{1}{|\cos\gamma|} \times \int_{-\infty}^{\infty} f_{\varepsilon_{qx}}\left(\tan\gamma\left(\frac{1}{\sin\gamma}\tilde{\varepsilon}_d - \tau\right)\right) f_{\varepsilon_{qy}}(\tau)\, d\tau, \tag{3}$$

where γ is the angle between the line segment and the horizontal axis of the image plane.

Figure 4 shows the probability density function of the two-dimensional quantization errors where γ is 40° or 50° and r_x equals r_y.

2.2. Displacement Errors

In active visual inspection, we place the sensors by the active head; there are usually errors in the final position and orientation. If the sensor location and orientation are different from the planned sensor setting (i.e., there is sensor displacement), the same edge segments may be observable, but the dimension measured will be inaccurate. The difference

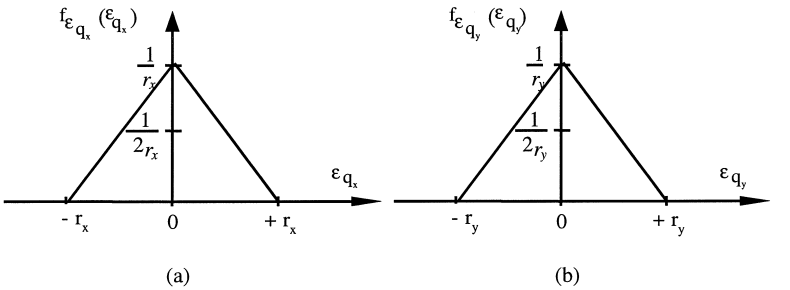

Figure 3. A line on a two-dimensional array of pixels. The horizontal resolution is r_x and the vertical resolution is r_y.

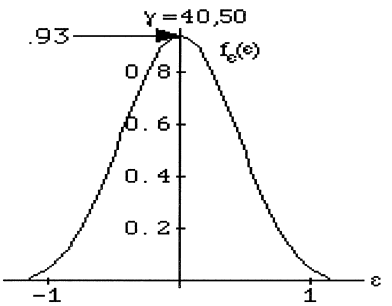

Figure 4. Probability density of the two-dimensional quantization error for $\gamma = 40°$ or $50°$.

between the observed dimensions and the actual dimensions is defined as *displacement errors*. Figure 5 gives an illustration of the displacement errors.

The displacement errors are derived based on the perspective transformation.[28,31] Given that the coordinate of an object point in the world coordinate system is $[x_w, y_w, z_w]^T$, the corresponding projected coordinate on the image plane is $[u, v]^T$. As illustrated in Figure 6, $[x_{w1}, y_{w1}, z_{w1}]^T$ is projected onto $[u_1, v_1]^T$ on the image plane. Similarly, $[x_{w2}, y_{w2}, z_{w2}]^T$ to $[x_{w7}, y_{w7}, z_{w7}]^T$ is projected onto $[u_2, v_2]^T$ to $[u_7, v_7]^T$ on the image plane.

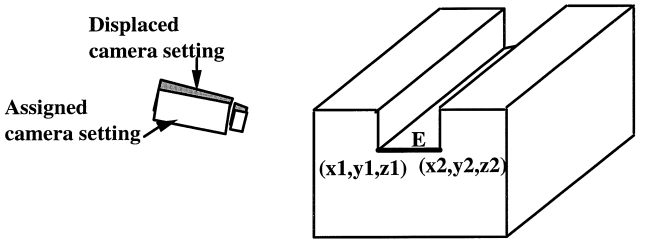

Figure 5. An edge segment, E, which is the width of the slot on the component, with end points, $(x1, y1, z1)$ and $(x2, y2, z2)$, is dimensioned by the camera with the assigned setting as shown. However, due to displacement errors, the displaced camera setting could undesirably affect the measurement.

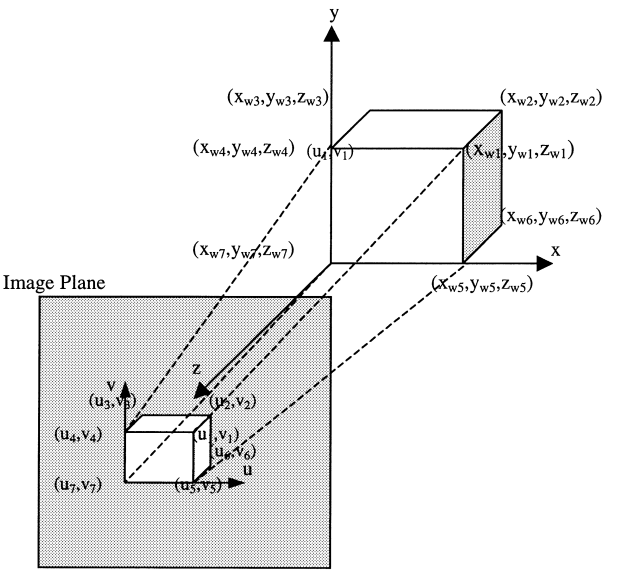

Figure 6. Perspective projection: an object point (x_{wi}, y_{wi}, z_{wi}) is projected onto the image plane as (u_i, v_i) where $i = 1, 2, 3, \ldots, 7$.

$$u = \frac{fC1}{f - C3} \tag{4}$$

$$v = \frac{fC2}{f - C3} \tag{5}$$

where $C1 = r_{11}x_w + r_{12}y_w + r_{13}z_w + t_x$, $C2 = r_{21}x_w + r_{22}y_w + r_{23}z_w + t_y$, $C3 = r_{31}x_w + r_{32}y_w + r_{33}z_w + t_z$, r_{11}, r_{12}, r_{13}, r_{21}, r_{22}, r_{23}, r_{31}, r_{32} and r_{33} are the parameters of the rotational matrix between the world coordinate and the image coordinate systems, and t_x, t_y, and t_z are the parameters of the translational matrix between the world coordinate and the image coordinate systems.

Given the translational errors, d_x, d_y, and d_z, and the rotational errors, δ_x, δ_y, and δ_z, the projected image coordinate will be displaced to $[u', v']^T$:

$$u' = \frac{f(C1 + C3\,\delta y - C2\,\delta z + dx)}{f - (C3 + C2\,\delta x - C1\,\delta y + dz)} \tag{6}$$

$$v' = \frac{f(C2 + C1\,\delta z - C3\,\delta x + dy)}{f - (C3 + C2\,\delta x - C1\,\delta y + dz)} \tag{7}$$

The displacement errors of the projected image coordinate in horizontal and vertical directions, ε_{du} and ε_{dv}, are $u' - u$ and $v' - v$, respectively,

$$\varepsilon_{du} = \frac{fC1C2\,\delta x + f((f-C3)C3 - C1^2)\,\delta y + fC2(C3 - f)\,\delta z + f(f-C3)\,dx + fC1\,dz}{-C2(f-C3)\,\delta x + C1(f-C3)\,\delta y + (f-C3)\,dz + (f-C3)^2} \tag{8}$$

$$\varepsilon_{du} = \frac{f(C2^2 - (f-C3)C3)\,\delta x - fC1C2\,\delta y + fC1(f-C3)\,\delta z + f(f-C3)\,dy + fC2\,dz}{-C2(f-C3)\,\delta x + C1(f-C3)\,\delta y + (f-C3)\,dz + (f-C3)^2}. \tag{9}$$

Assuming that the translation errors and rotational errors are normal distributed with zero means, the density functions of the horizontal and vertical displacement errors of a point (u, v) on the image are:

For the x direction,

$$f_{\varepsilon_{d_u}}(\varepsilon_{d_u}) = \frac{\sigma_\zeta \sigma_\chi \sqrt{1 - r_{\zeta\chi}^2}}{\pi g_1(\varepsilon_{d_u})} \times \exp\left(-\frac{u_\chi^2}{2g_1(\varepsilon_{d_u})} \left(\varepsilon_u^2 + \frac{g_2(\varepsilon_{d_u})^2}{\sigma_\chi^2} \right) \right) + \frac{g_2(\varepsilon_{du})\mu_\chi \sigma_\zeta}{\sqrt{2\pi}\, g_1(\varepsilon_{du})^{3/2}} \exp\left(-\frac{\mu_\chi^2 \varepsilon_{du}^2}{2g_1(\varepsilon_{du})} \right) \times \operatorname{erf}\left(\frac{g_2(\varepsilon_{du})\mu_\chi}{\sigma_\xi \sqrt{2\left(1 - r_{\zeta\chi}^2\right) g_1(\varepsilon_{du})}} \right), \tag{10}$$

and for the y direction,

$$f_{\varepsilon_{d_v}}(\varepsilon_{d_v}) = \frac{\sigma_\xi \sigma_\chi \sqrt{1 - r_{\xi\chi}^2}}{\pi h_1(\varepsilon_{d_v})} \times \exp\left(-\frac{\mu_\chi^2}{2h_1(\varepsilon_{d_v})} \left(\varepsilon_{d_v}^2 + \frac{h_2(\varepsilon_{d_v})^2}{\sigma_\chi^2} \right) \right) + \frac{h_2(\varepsilon_{dv})\mu_\chi \sigma_\xi}{\sqrt{2\pi}\, h_1(\varepsilon_{dv})^{3/2}} \exp\left(-\frac{\mu_\chi^2 \varepsilon_{dv}^2}{2h_1(\varepsilon_{dv})} \right) \times \operatorname{erf}\left(\frac{h_2(\varepsilon_{dv})\mu_\chi}{\sigma_\chi \sqrt{2\left(1 - r_{\xi\chi}^2\right) g_1(\varepsilon_{dv})}} \right), \tag{11}$$

where $g_1(\varepsilon_{d_u}) = \sigma_\zeta^2 - 2r_{\zeta\chi}\sigma_\zeta\sigma_\chi\varepsilon_{d_u} + \sigma_\chi^2\varepsilon_{d_u}^2$, $g_2(\varepsilon_{d_u}) = \sigma_\zeta - r_{\zeta\chi}\sigma_\chi\varepsilon_{d_u}$, $h_1(\varepsilon_{d_v}) = \sigma_\xi^2 - 2r_{\xi\chi}\sigma_\xi\sigma_\chi\varepsilon_{d_v} + \sigma_\chi^2\varepsilon_{d_v}^2$, $h_2(\varepsilon_{d_v}) = \sigma_\xi - r_{\xi\chi}\varepsilon_{d_v}$, ζ and ξ are the numerators of the horizontal and vertical displacement errors, and χ is the denominator of the horizontal and vertical displacement errors.

Figure 7 shows the probability density functions of the displacement errors in horizontal or vertical directions, $f_{\varepsilon_{d_u}}(\varepsilon_{d_u})$ or $f_{\varepsilon_{d_v}}(\varepsilon_{d_v})$, given that $\mu_\chi = 1.0$, 5.0, 10.0, or 20.0, $\sigma_\zeta = \sigma_\xi = \sigma_\chi = 2.0$, and $r_{\zeta\chi} = r_{\xi\chi} = 0.5$. The density functions of horizontal and vertical displacement errors for a line segment are then derived as:

For the x direction,

$$f_{\varepsilon_{d_x}}(\varepsilon_{d_x}) = \int_{-\infty}^{\infty} f_{\varepsilon_{d_{u_1}}}(\varepsilon_{d_x} + \tau) f_{\varepsilon_{d_{u_2}}}(\tau)\, d\tau. \tag{12}$$

For the y direction,

$$f_{\varepsilon_{d_y}}(\varepsilon_{d_y}) = \int_{-\infty}^{\infty} f_{\varepsilon_{d_{v_1}}}(\varepsilon_{d_y} + \tau) f_{\varepsilon_{d_{v_2}}}(\tau)\, d\tau. \tag{13}$$

The density functions of two-dimensional displacement errors are derived similar to the quantization errors using a geometric approximation:

$$f_{\tilde{\varepsilon}_d}(\tilde{\varepsilon}_d) = \frac{1}{|\cos\gamma|} \times \int_{-\infty}^{\infty} f_{\varepsilon_{d_x}}\left(\tan\gamma \left(\frac{1}{\sin\gamma} \tilde{\varepsilon}_d - \tau \right) \right) f_{\varepsilon_{d_y}}(\tau)\, d\tau. \tag{14}$$

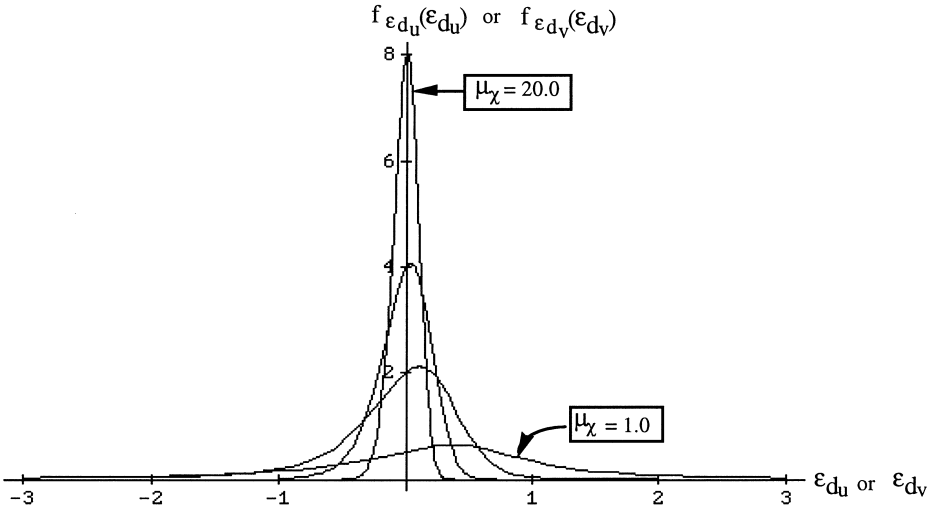

Figure 7. The probability density functions of the displacement errors due to displacement of the end-effectors in horizontal or vertical directions, $f_{\varepsilon_{d_u}}(\varepsilon_{d_u})$ or $f_{\varepsilon_{d_v}}(\varepsilon_{d_v})$, where $\mu_\chi = 1.0$, 5.0, 10.0, or 20.0, $\sigma_\zeta = \sigma_\xi = \sigma_\chi = 2.0$, and $r_{\xi\chi} = r_{\xi\chi} = 0.5$.

2.3. Integrating Quantization and Displacement Errors

Given the density functions of the quantization errors and displacement errors, we integrate both to obtain the total error in dimensional inspection using active vision. The total inspection error, ε_i, is the sum or the quantization errors and the displacement errors. Its density function is

$$f_\varepsilon(\varepsilon) = \int_{-\infty}^{\infty} f_{\varepsilon_q}(\varepsilon_i - \tau) f_{\varepsilon_d}(\tau)\, d\tau. \qquad (15)$$

3. SENSOR CONSTRAINTS

Visual inspection has become popular because of advances in computer and imaging technologies; however, we need careful sensor placement to make sure all the sensor constraints are satisfied. In this section, we discuss resolution constraints, focus constraints, field-of-view constraints, and visibility constraints.

3.1. Resolution Constraints

To obtain the desired accuracy in the measured dimensions of a line segment on the image, a minimum resolution is needed.[11,37] A minimum resolution of the line segment on image constrains a maximum distance between the corresponding edge segment of the object and the sensor. We can compute the maximum distance in terms of the focal length and the distance between the image plane and the lens. Using this formulation, we can obtain a locus of points (a viewing sphere) that give the maximum bound of the viewing locations,

$$\frac{\left\| \left[\left(\vec{r}_1 - \vec{r}_v \right) \times \vec{u} \right] \times \vec{v} \right\|}{\left[\left(\vec{r}_1 - \vec{r}_v \right) \cdot \vec{v} \right] \left[\left(\vec{r}_2 - \vec{r}_v \right) \cdot \vec{v} \right]} - \frac{L_{\min}}{dL_w} \geq 0 \qquad (16)$$

where $\vec{r}_1$ and $\vec{r}_2$ are the position vectors of the vertices of the line segment, $\vec{r}_v$ is the position vector of the front nodal point of the lens, $\vec{u}$ is the unit vector along the line segment, $\vec{v}$ is the unit vector along the optical axis in the viewing direction, L_w is the length of the line segment, $L_{\min}$ is the minimum required length of the line segment on the image, and d is the distance from the back nodal point of the lens to the image plane.

3.2. Focus Constraints

To keep the image in focus, we need a lower bound and a upper bound for the distance between the sensor and the inspected edge segment.[11,37] For any lens setting of the sensors, there is a *focus distance*, at which a point is perfectly focused on the image plane. Any points within a range of the focus distance are in focus if the diameter of the blur circle of each point is less than the minimum pixel dimen-

sion. Based on the focus distance, we can find the maximum and minimum distances, D_{max} and D_{min}, such that any points within these distances are in focus. The difference between the maximum and minimum distance is the *depth of field*,

$$\left(\vec{r}_c - \vec{r}_v\right)\cdot\vec{v} - D_{min} \geq 0 \tag{17}$$

$$D_{max} - \left(\vec{r}_f - \vec{r}_v\right)\cdot\vec{v} \geq 0 \tag{18}$$

where $\vec{r}_c$ is the position vector of the closest feature vertex from the front nodal point of the lens along the viewing direction, $\vec{r}_f$ is the position vector of the farthest feature vertex from the front nodal point of the lens along the viewing direction, $\vec{r}_v$ is the position vector of the front nodal point of the lens, $\vec{v}$ is the unit vector along the optical axis in the viewing direction, D_{min} is the near limit of the depth of field, and D_{max} is the farthest limit of the depth of field.

3.3. Field-of-View Constraints

The sensor's field of view provides a minimum distance between the sensor and the inspected edge segment. Any part of the inspected edge segment that projects completely or partially outside of the sensor area is not within the sensor's field of view. For a camera with a rectangular sensor, the field-of-view boundary is a tetrahedron, where its apex is the front nodal point of the lens. The field-of-view angle, α, depends on the minimum dimension of the image plane and the distance between the image plane and the lens center,

$$\frac{\vec{r}_s - \dfrac{R_s}{\sin(\alpha/2)}\vec{v} - \vec{r}_v}{\left\|\vec{r}_s - \dfrac{R_s}{\sin(\alpha/2)}\vec{v} - \vec{r}_v\right\|\cos(\alpha/2)} \geq 0 \tag{19}$$

$$\frac{\alpha}{2} = \tan^{-1}\left(\frac{I_{min}}{2d}\right) \tag{20}$$

where $\vec{r}_s$ is the position vector of the center of a sphere circumscribing the inspected features, $\vec{r}_s$ is the position vector of the front nodal point of the lens, $\vec{v}$ is the unit vector along the optical axis in the viewing direction, R_s is the radius of the sphere circumscribing the inspected features, α is the field-of-view angle, d is the distance from the back nodal point of the lens to the image plane, and I_{min} is the minimum dimension of the image plane.

3.4. Visibility Constraints

Cowan and Kovesi[7] and Tarabanis et al.[11,12] used the structure of an object and the entities to be observed to construct a region such that all entities are visible. However, this methodology is not as efficient as the aspect graph methodology.[34] An aspect graph is a graphic representation of all the characteristic views of an object. Each node of the graph corresponds to a distinct characteristic view of an object. Given a set of entities, we can obtain all the viewing domains that are capable of observing these entities. If a different set of entities is to be inspected, the corresponding set of viewing domains can be found based on the aspect graph without reconstructing all of the viewing domain boundaries.[38]

The complexity of constructing the complete aspect graph is high,[34] and, therefore, it is not favored in most of the current sensor planning systems. However, the complete aspect graph is not needed in determining the sensor setting for inspected entities of the complete object. In this article, we propose to use the entity-based aspect graph to obtain the visibility regions for the inspected entities only. The entity-based aspect graph is constructed based on the entity of interest, E, where E is the inspected entities in the active visual inspection planning system. In the entity-based aspect graph, only the distinct characteristic views of the inspected entities are constructed. The characteristic views of any other entities apart from the inspected entities will no longer be constructed. As a result, the complexity will be reduced significantly. For example, Figure 8 shows an object with a step feature; the entities of interest are $e1$, $e2$, $e3$, $e4$, $e5$, $e6$, and $e7$, as labeled. The constructed entity-based aspect graph has 20 nodes compared to 71 nodes in the complete aspect graph of the complete object.[39] The algorithm of constructing the entity-based aspect graph is as follows:

Algorithm for Constructing the Entity-Based Aspect Graph (EAG) with Entity of Interests= E

Input: The boundary representation of the object (vertices, edges, and faces).

Output: A EAG with a set of viewing domains, V, a set of lists of observable entities, O, one list for each element of V, and a set of adjacent pairs of the viewing domain, A, EAG (E, V, O, A).

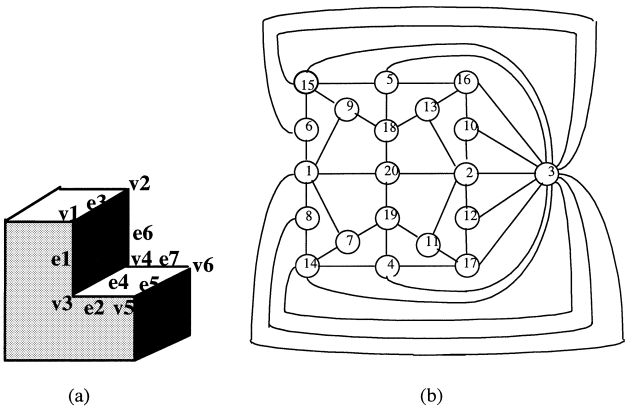

Figure 8. (a) An object with a step feature: the entity of interests are $e1$, $e2$, $e3$, $e4$, $e5$, $e6$, and $e7$. (b) The entity-based aspect graph for the entities of interests as shown in (a).

1. Find all the partition planes and surfaces (*using type A, type B, and type C partition planes and surfaces described by Freeman*[39]) and eliminate those generated by entities that are not elements of $E, \{p_1, p_2, \ldots, p_n\}$.
2. Construct all possible n-tuples, $L = \{\{p_{1^+}, p_{2^+}, \ldots, p_{n^+}\}, \{p_{1^-}, p_{2^+}, \ldots, p_{n^+}\}, \ldots, \{p_{1^-}, p_{2^-}, \ldots, p_{n^-}\}\}$ where p_{i^+}, p_{i^-} are inequalities denoting the different half-spaces of the partition plane p_i.
3. For each n-tuple,
   Determine the feasibility of the three-dimensional region described by it. (*If a point cannot belong the the corresponding set of n inequalities, the region is infeasible.*)
   If such an n-tuple is feasible,
      Determine the exact boundaries of such a three-dimensional region. (*If all the intersections of an element in the n-tuple* (*a half-space*) *with the other half-spaces in the n-tuple are not on the boundary of the region represented by the n-tuple, then that element* (*half-space*) *can be removed from the n-tuple.*)
      Save the reduced tuple in V.
      Save the corresponding list of observable entities in O. (*The list of observable entities is the union of entities observable from each* p_{i^+} *in the n-tuple.*)
4. For each pair of elements in V,
   If these elements share the same boundary in the three-dimensional space,
   The corresponding regions are adjacent.
   Save the pair in A.

4. OPTIMIZATION BY THE GENETIC ALGORITHM

To obtain inspections with high accuracy, we determine the optimized sensor settings such that the active vision errors are minimized and the sensor constraints are satisfied. The objective of the optimization is to maximize the accuracy of inspection. In other words, we maximize the probability that the errors are within an acceptable tolerance. The constraints of the optimization are the sensor constraints as described in Section 3.

In this work, we use a genetic algorithm to perform the optimization. In our experience, when using traditional nonlinear programming, such as gradient search, penalty approach, etc., trapping at the local minimum occurs occasionally. Moreover, its performance is not consistent. On the other hand, the genetic algorithm performs more consistently. Because of the random generation process (mutation operation) in the genetic algorithm, trapping at a local minimum does not occur.

4.1. Genetic Algorithm

Initialization of Population

A chromosome represents a sensor setting. The genes in a chromosome represent the locations and orientations of sensor settings, which are binary numbers. A chromosome has six sets of genes; three of them correspond to the three parameters of locations, and the other three of them correspond to the three parameters of orientations. The higher number of genes (M) in a chromosome, the higher the precision of the parameters of translations and orientations. Initially, N numbers of chromosomes are generated randomly. For each chromosome, C_i, we compute the total probability P_i that the active vision errors are over the required accuracy for all inspected entities, A ($i = 1, 2, \ldots, N$).

$$P_j = P\left(|\varepsilon| \le L_j(1 - A)\right) \tag{21}$$

where L_j is the project length of entity j.

$$P_i = \text{is the fitness of chromosome I} = \sum_j P_j. \tag{22}$$

Reproduction

Reproduction is the selection of a new population. A chromosome that has a higher P_i has a better chance of being selected. Each chromosome occupies a certain number of slots on a roulette wheel directly proportional to its P_i. Spinning the roulette wheel N times, we select N chromosomes for the new population. According to the genetic inheritance, the best chromosomes get more copies, the average stay even, and the worst die off.

Crossover and Mutation

There are two recombination operations, crossover and mutation. The crossover operates between a pair of chromosome and the mutation operates on a single chromosome.

Crossover. The probability of crossover, P_c, gives us the expected number $P_c N$ of chromosomes that should undergo the crossover operation. For each chromosome, we generate a random number, X, between 0 and 1. If X is less than P_c, the chromosome is selected for crossover. For each pair of selected chromosomes, we generate a random number, Y, between 0 and $M-1$. Y indicates the position of the crossing point. The coupled chromosomes exchange genes at the crossover point. If the crossover chromosome does not satisfy the sensor constraints, it does not survive (it is eliminated).

Mutation: Mutation is performed on a bit-by-bit basis. The probability of mutation, P_m, gives us the expected number of mutated bits $P_m MN$. Every bit in all chromosomes of the whole population has an equal chance to undergo mutation. For each chromosome and for each bit within the chromosome, we generate a random number, Z, between 0 and 1. If Z is less than P_m, we mutate the bit. Similar to the crossover operation, if the mutated chromosome does not satisfy the sensor constraints, it does not survive (it is eliminated).

Convergence

After reproduction, crossover, and mutation, the new population is ready for the next generation. The evolution of the solution continues in this way, repeating these steps until the system converges. This occurs when the total of P_i for the whole population decreases less than a small value, δ, for a few generations.

5. EXPERIMENT

We conduct two experiments to investigate the performance of the genetic algorithm in optimizing the sensor placements. Two objects as shown in Figures 9(a) and 10(a) are used in our experiments. Parameters that are utilized in the computation of the probability density function of displacement errors and quantization errors are listed below:

Robot manipulator parameters for displacement error are

$$\sigma_x^2 = \sigma_y^2 = \sigma_z^2 = 0.03 \text{ mm}^2$$
$$\sigma_{\delta x}^2 = \sigma_{\delta y}^2 = \sigma_{\delta z}^2 = 0.003 \text{ rad}^2.$$

Sensor placements are

$$r_x = 0.01 \text{ mm}$$
$$r_y = 0.013 \text{ mm}.$$

To determine the visibility constraint, the entity-based aspect graphs for the objects in Figures 9(a) and 10(a) are developed and shown in Figures 9(b) and 10(c), respectively, where the entities are labeled in Figure 9(a) and 10(b), respectively. There are totally eight nodes and nine nodes on the entity-based aspect graphs, where one of them on each does not have any visibility entity. The corresponding entities for each node on Figures 9(b) and 10(c) are given in Tables I and Table II.

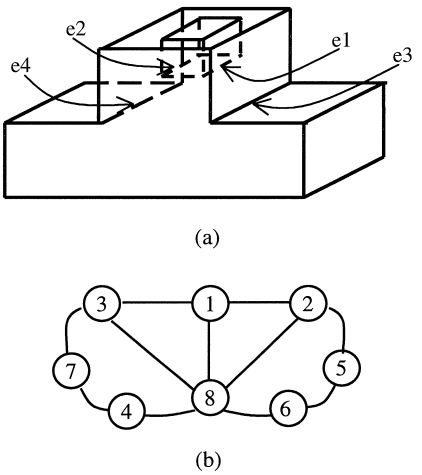

Figure 9. (a) An object with four edges as entities of interest. (b) An entity-based aspect graph for the object in Figure 9(a).

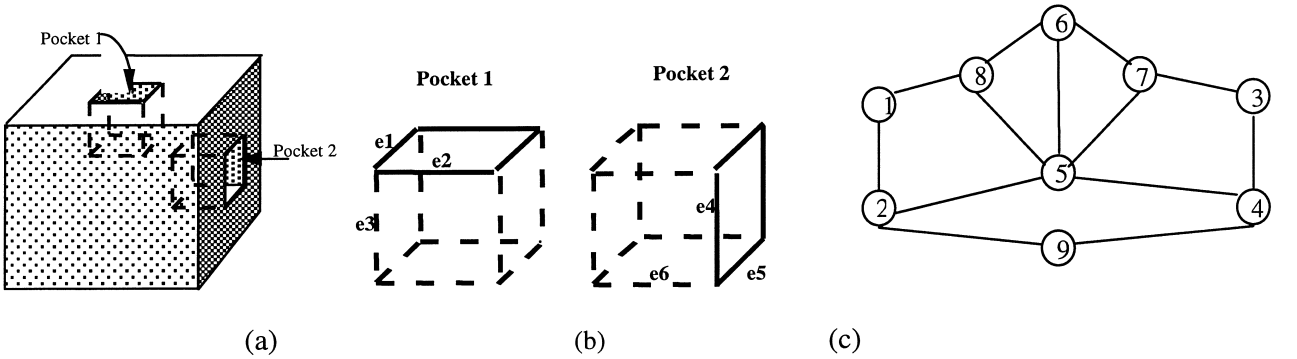

Figure 10. (a) An object with six edges as entities of interest, three from each pocket on two adjacent sides of the object. (b) Entities $e1$, $e2$, $e3$, $e4$, $e5$, and $e6$, labeled on the pockets. (c) Entity-based aspect graph for the object in Figure 10(a).

The parameters used in the genetic algorithm, such as crossover rate, mutation rate, and population, are important for the optimization result. In experiment 1, we shall investigate the effect of these parameters.

Table I. Visible entities in the nodes of the entity-based aspect graph in Figure 9(b) for the object in Figure 9(a).

Node	Visibility
1	$e1$, $e2$
2	$e2$
3	$e1$
4	$e2$, $e3$
5	$e3$
6	$e1$, $e4$
7	$e4$
8	None

Table II. Visible entities in the nodes of the entity-based aspect graph in Figure 10(c) for the object in Figure 10(a).

Node	Visibility
1	$e1$, $e2$, $e3$
2	$e1$, $e2$
3	$e4$, $e5$, $e6$
4	$e4$, $e5$
5	$e1$, $e2$, $e4$, $e5$
6	$e1$, $e2$, $e3$, $e4$, $e5$, $e6$
7	$e1$, $e2$, $e4$, $e5$, $e6$
8	$e1$, $e2$, $e3$, $e4$, $e5$
9	None

As shown in the entity-based aspect graphs, in some characteristic views, more than one entity can be observed. However, inspecting multiple entities with the use of only one sensor setting may not obtain the minimum errors for all entities. The optimized sensor setting only minimizes the global errors for all entities. These effects will be discussed in experiment 2.

Experiment 1

The values of the crossover rate (P_c), mutation rate (P_m), and population (N) in the genetic algorithm affect the result of the optimization and the speed of the processing significantly. In this experiment, we investigate how the combinations of these parameters affect the probability of a required accuracy and the time needed to obtain the result.

We perform the genetic algorithm on each of the entities labeled in Figures 9(a) and 10(b). Different combinations of P_c, P_m, and N are selected. For each combination, we perform the genetic algorithm 5 times on each of 10 entities. The probabilities to obtain 98% or above accuracy, the number of generations, and the processing time for each optimization are recorded. The average of 50 optimizations for each combination are presented in Table III.

The result in Table III shows that the probability of 98% or above accuracy decreases as the mutation rate increases and the crossover rate decreases. The number of generations and the processing time increase as the mutation rate increases from 0.1 to 0.5 and the crossover rate decreases from 0.9 to 0.5 when the populations are 20 and 50. When the population is 100, the number of generations and the processing time increase as the mutation rate

Table III. Result of optimized sensor settings by genetic algorithm.

	Crossover rate (P_c)	Mutation rate (P_m)	Population (N)	Probability of 98% or above accuracy	No. of generations	Processing time (s)
1	0.9	0.1	110	0.95	104	135
2	0.9	0.1	100	0.95	105	136
3	0.9	0.1	90	0.95	109	141
4	0.9	0.1	70	0.95	114	148
5	0.9	0.1	50	0.95	132	168
6	0.9	0.1	20	0.95	189	242
7	0.7	0.3	100	0.93	252	328
8	0.7	0.3	50	0.92	288	373
9	0.7	0.3	20	0.92	579	754
10	0.5	0.5	100	0.92	507	660
11	0.5	0.5	50	0.90	522	677
12	0.5	0.5	20	0.89	927	1208
13	0.3	0.7	100	0.80	738	958
14	0.3	0.7	50	0.78	492	642
15	0.3	0.7	20	0.70	366	473
16	0.1	0.9	100	0.50	431	561
17	0.1	0.9	50	0.45	298	389
18	0.1	0.9	20	0.30	135	173

increases from 0.1 to 0.7 and the crossover rate decreases from 0.9 to 0.3. The number of generations and the processing time begin to decrease as the mutation rate continues to increase and the crossover rate continues to decrease for all population. However, the probability of 98% or above accuracy decreases significantly as the number of generations or the processing time begins to decrease.

When the mutation rate increases, the percentage of mutated chromosomes in the population increases. The larger the percentage of mutated chromosomes is, the longer it takes to converge in the genetic algorithm. On the other hand, there is a smaller percentage of chromosomes that are generated by crossover, which means there is a lower chance of new chromosomes being generated by the best chromosomes in the current generation. When the mutation rate increases further, there are too many chromosomes being generated by mutation, converging in fewer generations before any chromosomes obtain a setting that is close to the optimized result.

When the crossover rate is in the range of 0.1 to 0.7 and the mutation rate is in the range of 0.3 to 0.9, the probability of 98% or above accuracy increases as the population increases. The number of generations decreases as the population increases. To obtain the optimized sensor setting in the shortest amount of time, a high crossover rate, a low mutation rate, and a high population are desired. When the crossover rate is 0.9 and the mutation rate is 0.1, the probability of 98% or above does not increase any further when the population increases. The number of generation decreases to 105 when the population increases to 100. When we further increase the population, the number of generations does not change any more significantly.

Experiment 2

The number of entities to be inspected by a sensor setting affects the accuracy of each measurement. As shown in Tables I and II, sensor settings located within some viewing domains have more entities that are visible. Although such sensor settings can be used to inspect more entities, the accuracy obtained for each entity may not be as high as those obtained by sensor settings that have fewer entities that are not visible.

In this experiment, we take four sets of entities as the inputs of the genetic algorithm to obtain the optimal sensor settings. The four sets of entities are $\{e1, e2\}$, $\{e1, e2, e3\}$, $\{e1, e2, e3, e4, e5\}$, and $\{e1, e2, e4, e5\}$. The sensor settings obtained by the genetic algorithm are sensor setting A, sensor setting B, sensor setting C, and sensor setting D, respectively. Image 1, image 2, image 3, and image 4 obtained by sensor setting A to D shown in Figure 11(a)–(d),

respectively. Their edge maps are shown in Figure 11(e)–(h), respectively. The probabilities of obtaining 98% or above accuracy for each entity are reported for each resultant optimal sensor setting in Table IV.

6. CONCLUSION

In this article, we propose an active visual inspection planning system to determine the optimized sensor settings for dimensioning a set of edge segments such that the active vision errors are minimized and the sensor constraints are satisfied. The quantization errors and displacement errors, which are inevitable in active vision inspections, are considered. The combined density functions of these errors are used as objective functions in the optimizations. The resolution constraints, focus constraints, field-of-view constraints, and visibility constraints are used as the constraint functions. We use the genetic algorithm for optimization so that the resulting sensor setting will have the minimum active vision errors and satisfy all sensor constraints. Using such a system, an optimized set of sensor settings is obtained to inspect a set of entities for a CAD model. Each sensor setting is able to inspect one or more entities and the probability of obtaining a minimum accuracy for each entity inspected is guaranteed to be above a threshold. Experiments have been conducted to measure the performance of the genetic algorithm using different parameters of crossover rate, mutation rate, and population. It shows that using a crossover rate of 0.9 and a mutation rate of 0.1 and population of 100 chromosomes produces the highest probability of 98% or above accuracy and takes the least amount of time. Experiments have also been conducted to investigate the effect of the number of entities on the accuracy. It shows that increasing the number of

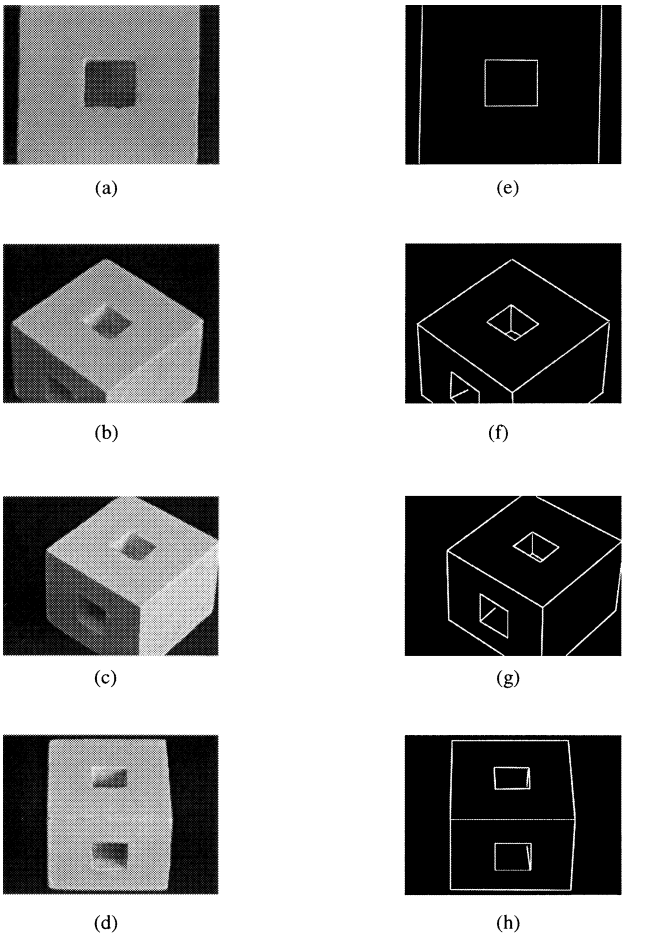

Figure 11. (a) Image 1 obtained by sensor setting A. (b) Image 2 obtained by sensor setting B. (c) Image 3 obtained by sensor setting C. (d) Image 4 obtained by sensor setting D. (e) Edge map of image 1. (f) Edge map of image 2. (g) Edge map of image 3. (h) Edge map of image 4.

Table IV. Probabilities of 98% or above accuracy for $e1$, $e2$, $e3$, $e4$, and $e5$ obtained by sensor setting A, sensor setting B, sensor setting C, and sensor setting D.

	Probability of 98% or above accuracy				
	$e1$	$e2$	$e3$	$e4$	$e5$
Sensor setting A	0.96	0.96			
Sensor setting B	0.94	0.94	0.90		
Sensor setting C	0.92	0.93	0.89	0.92	0.92
Sensor setting D	0.94	0.93		0.93	0.94

entities to be inspected by a sensor setting may result in a decrease in accuracy for some entities.

REFERENCES

1. S. Sakane, M. Ishii, and M. Kakikura, Occlusion avoidance of visual sensors based on a hand eye action simulator system: HEAVEN, Adv Robotics 2(2) (1987), 149–165.
2. S. Sakane, T. Satio, and M. Kakikura, Automatic planning of light source placement for an active photometric stereo system, Proc IEEE Int Workshop on Intelligent Robotics and Systems, New York, 1990, pp. 559–566.
3. S. Sakane and T. Sato, Automatic planning of light source and camera placement for an active photometric stereo system, Proc 1991 IEEE Int Conf Robotics and Automation, pp. 1080–1087.
4. R. Niepold, S. Sakane, T. Sato, and Y. Shirai, Vision sensor set-up planning for a hand-eye system using environmental model, Proc Society Instrum Control Engineering, Japan, 1988, pp. 1037–1040.
5. S. Sakane, R. Niepold, T. Sato, and Y. Shirai, Illumination setup planning for a hand-eye based on an environmental model, Adv Robotics 6(4) (1992), 461–482.
6. S. Yi, R.M. Haralick, and L.G. Shapiro, Automatic sensor and light source positioning for machine vision, Proc 10th Int Conf Pattern Recognition, 1990, pp. 55–59.
7. C.K. Cowan and P.D. Kovesi, Determining the camera and light source location for a visual task, Proc IEEE Int Conf Robotics and Automation, 1988.
8. C.K. Cowan, Model based synthesis of sensor location, Proc IEEE Int Conf Robotics and Automation, 1988, pp. 900–905.
9. K. Tarabanis, T.Y. Tsai, and P.K. Allen, Automated sensor planning for robotic vision tasks, IEEE Int Conf Robotics and Automation, Sacramento, CA, April, 1991.
10. K. Tarabanis, R.Y. Tsai, and P.K. Allen, Analytic characterization of the feature detectability constraints of resolution, focus, and field-of-view for vision sensor planning, CVGIP: Image Understanding 59(3) (1994), 340–358.
11. K. Tarabanis, R.Y. Tsai, and P.K. Allen, The MVP sensor planning system for robotic vision tasks, IEEE Trans Robotics and Automation, February, 1995, vol. 11, no. 1.
12. R.Y. Tsai and K. Tarabanis, Occlusion-free sensor placement planning, Machine vision for three-dimensional scenes, H. Freeman (Editor), Academic Press, Orlando, FL, 1990.
13. B.G. Batchelor, A Prolog lighting advisor, Proc SPIE Int Symp Intelligent Robots and Computer Vision VIII: System and Application, 1989, vol. 1193, pp. 295–302.
14. B.G. Batchelor, Integrating vision and AI for industrial applications, Proc SPIE Int Symp on Intelligent Robots and Computer Vision VIII: System and Application, 1989, vol. 1193, pp. 168–173.
15. Y. Kitamura, H. Sato, and H. Tamura, An expert system for industrial machine vision, Proc 10th Int Conf Pattern Recognition, 1990, pp. 771–773.
16. A. Novini, Lighting and optics expert system for machine vision, Proc Optics, Illumination, Image Sensing, 1988, pp. 16–25.
17. B. Kamgar-Parsi and B. Kamgar-Parsi, Evaluation of quantization error in computer vision, IEEE Trans Pattern Anal Machine Intell 11(9) (1989).
18. S.D. Blostein and T.S. Huang, Error analysis in stereo determination of 3-D point positions, IEEE Trans Pattern Anal Machine Intell 9(6) (1987).
19. C.S. Ho, Precision of digital vision systems, IEEE Trans Pattern Anal Machine Intell 5(6) (1983).
20. P.M. Griffin and J.R. Villalobos, Process capability of automated visual inspection system, IEEE Trans Syst Man Cybernet 22(3) (1992).
21. S.F. Su and G.C.S. Lee, Manipulation and propagation of uncertainty and verification of application of actions in assembly tasks, IEEE Trans Systems Man Cybernet 22(6) (1992).
22. C.H. Menq and J.H. Borm, Statistical measure and characterization of robot errors, Proc IEEE Int Conf Robotics and Automation, 1988.
23. J.M. Renders, E. Rossignol, M. Becquet, and R. Hanus, Kinematic calibration and geometrical parameter identification for robots, IEEE Trans Robotics and Automation 7(6) (1991).
24. J. Chen and L.M. Chao, Positioning error analysis for robot manipulators with all rotary joints, IEEE Int Conf Robotics and Automation, San Francisco, CA, April, 1986.
25. W.K. Veitschegger and C.H. Wu, A method for calibrating and compensating robot kinematic errors, IEEE Int Conf Robotics and Automation, 1987.
26. S. Bryson, Measurement and calibration of static distortion of position data from 3D trackers, Steroscopic displays and applications III, SPIE 1992, vol. 169.
27. R.C. Smith and P. Chessman, On the representation and estimation of spatial uncertainty, Int J Robotics Res 5(4) (1986).
28. F.W. Ciarallo, C.C. Yang, and M.M. Marefat, Displacement errors in active visual inspection, Proc IEEE Int Conf on Robotics and Automation, Minneapolis, MN, April 22–28, 1996.
29. C.C. Yang and M.M. Marefat, Spatial quantization errors in active vision inspection, Proc IEEE Int Conf Systems, Man and Cybernetics, San Antonio, TX, October 2–5, 1994.
30. C.C. Yang, M.M. Marefat, and F.W. Ciarallo, Analysis of errors in dimensional inspection based on active vision, Proc SPIE Int Symp Intelligent Robots and Computer Vision XIII: 3D Vision, Product Inspection, Active Vision, Cambridge, MA, October 31–November 4, 1994.
31. C.C. Yang, M.M. Marefat, and F.W. Ciarallo, Analysis of errors and planning accuracy on dimensional measurement in active vision inspection, IEEE Trans Robotics and Automation 14(3) (1998), 476–487.
32. C.K. Cowan and P.D. Kovesi, Automatic sensor placement for vision task requirements, IEEE Trans Pattern Analysis and Machine Intelligence, 1988, vol. 10, no. 3.

33. C.K. Cowan, Automatic camera and light source placement using CAD models, Proc IEEE Workshop on Direction in Automated CAD-Based Vision, 1991.
34. K. Tarabanis, R.Y. Tsai, and A. Kaul, Computing occlusion-free viewpoints, IEEE Trans Pattern Analysis and Machine Intelligence, March 1996, vol. 18, no. 3, pp. 279–292.
35. C.C. Yang and M.M. Marefat, Flexible active computer integrated visual inspection, Proc NSF Design and Manufacturing Conference, Cambridge, MA, January 5–7, 1994.
36. K. Crosby, C.C. Yang, M.M. Marefat, and F.W. Ciarallo, Camera settings for dimensional inspection using displacement and quantization errors, Proc IEEE Int Conf Robotics and Automation, Albuquerque, NM, April, 1997.
37. S. Abrams, P.K. Allen, and K. Tarabanis, Computing camera viewpoints in an active robot work cell, Int J Robotics Res 18(3) (1999), 267–285.
38. C.C. Yang, M.M. Marefat, and E. Johnson, Entity-based aspect graphs: making the viewer centered representation more efficient, Pattern Recognition Lett 19(3–4) (1998), 265–277.
39. H. Freeman, The use of characteristic-view classes for 3D object recognition: machine vision for three-dimensional scenes, Academic Press, New York, 1990, pp. 109–163.