The MVP Sensor Planning System for Robotic Vision Tasks

Konstantinos A. Tarabanis, Roger Y. Tsai, and Peter K. Allen, *Member, IEEE*

Abstract—**The MVP model-based sensor planning system for robotic vision is presented. MVP automatically synthesizes desirable camera views of a scene based on geometric models of the environment, optical models of the vision sensors, and models of the task to be achieved. The generic task of feature detectability has been chosen since it is applicable to many robot-controlled vision systems. For such a task, features of interest in the environment are required to simultaneously be visible, inside the field of view, in focus, and magnified as required. In companion papers we analytically characterize the domain of admissible camera locations, orientations, and optical settings for which each of the above feature detectability requirements is satisfied separately. In this paper, we present a technique that poses the vision sensor planning problem in an optimization setting and determines viewpoints that satisfy all previous requirements simultaneously and with a margin. In addition, we present experimental results of this technique when applied to a robotic vision system that consists of a camera mounted on a robot manipulator in a hand–eye configuration. The camera is positioned and the lens is focused according to the results generated by MVP. Camera views taken from the computed viewpoints verify that all feature detectability constraints are indeed satisfied.**

I. Introduction

A. The Problem

Automation in practice is presently dominated by special-purpose machines that perform predetermined functions in prespecified and tightly controlled environments. Since these systems are clearly inflexible and generally cost a great deal, considerable interest has been drawn to flexible sensor-based automation systems that are able to carry out functions in a more flexible working environment and at lower cost. Such systems are equipped with various types of sensors in order to interact in an intelligent and flexible manner with the environment. However, despite this added intelligence, many functions still require substantial human involvement. For instance, determining the appropriate *configuration* of the sensors, *programming* the sensor-based system itself or, in general, determining *sensor parameter values* that accomplish the task at hand to a prespecified degree of satisfaction, remain very human-intensive operations that considerably increase the development time, cost, and complexity of such automation systems. Sensor planning attempts to automate functions such as these in order to produce more flexible and autonomous sensor-based systems.

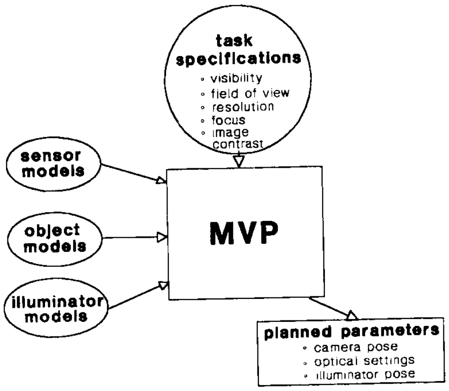


Fig. 1. MVP sensor planning system.

In particular, we are investigating the problem of *model-based* and *task-driven sensor planning* (see Fig. 1). That is, by making use of

1) Models of the environment (e.g., CAD models of parts, calibration models of the sensors, illumination models) and
2) The task requirements (e.g., feature resolution of 3 mils),

planning algorithms are developed that automatically determine sensor parameters values, such as sensor locations and sensor settings, satisfying the requirements of the task at hand.

In currently employed robot controlled vision systems that observe known environments (e.g., visual inspection systems), appropriate sensor parameters values are determined by often laborious and time-consuming techniques. Generally, a trial-and-error approach involving human interaction is taken. Sensor locations and settings are chosen and then tested in order to verify whether they meet the requirements of the task at hand. This procedure results in parameter values that are valid for only a specific setup and that can potentially become unsatisfactory when errors (e.g., robot inaccuracy) alter the environment. Such procedures constitute a major bottleneck when installing such systems, resulting in a design cost which often exceeds that of the equipment. As a result, these applications are costly and have limited intelligence and flexibility.

While sensor parameter values are currently determined manually, information is often available which can be used to

Manuscript received February 17, 1992. This paper was presented in part at the 1991 IEEE International Conference on Robotics and Automation, Sacramento, CA.

K. Tarabanis and P. K. Allen are with the Department of Computer Science, Columbia University, New York, NY 10027 USA.

R. Y. Tsai is with the IBM T. J. Watson Research Center, Yorktown Heights, NY 10598 USA.

IEEE Log Number 9400865.

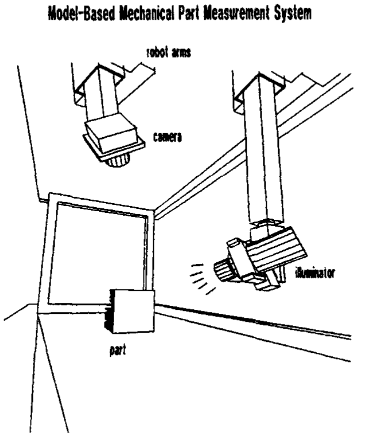


Fig. 2. Example application of vision planning techniques to a robotic mechanical part measurement system.

automatically determine many sensor parameters. For instance, the required geometric and physical information of objects can be extracted from the CAD/CAM models of these parts which are often available in today's manufacturing environment. In addition to object models, camera and illumination models embodying their physical and geometric properties can provide the planning system with the required sensor and illumination characteristics. The planning algorithms can use this model information and augment it with knowledge regarding the functions the system is to perform. In this way, the sensor-based system will be able to reason about task related events, one of which is its own configuration.

B. Motivation

In this section we see that a sensor planning component would prove useful in several areas of automation.

The importance of such a component for newer generation *automated visual inspection systems* has been recognized by several researchers [16], [30], [36]. In [30], a system is defined to automatically generate dimensional measurement applications from CAD models of parts. In [16], a concept-based inspection system is outlined in which a model-base inspection system would be augmented with a generic set of principles in order to determine appropriate and flexible behavior in new situations. In [36], the need is underlined for machine vision development tools that will assist machine vision designers and will allow less specialized labor to be employed (see Fig. 2). The tool proposed would automatically generate the configuration of a machine vision system by selecting sensors, illumination, optics, and image processing algorithms.

Vision sensor planning is also useful for *robot-controlled vision systems* in which cameras and light sources are mounted on robot manipulators (e.g. [1], [35]). In order for these systems to perform their task reliably (e.g., remote assembly/disassembly or manipulation of objects in space or in hazardous environments), selection of the proper vision sensor parameter values is critical. The developed planning techniques can be used in such systems in order to automatically position and orient the cameras and light sources, as well as to control the camera optics (e.g., controlling the zoom, focus, and aperture settings of programmable zoom lenses).

Similarly, sensor planning techniques are applicable to areas such as the *automatic synthesis of vision programs* from task specifications and model information [7], [10], [15]. For example, to a certain extent, a vision program to inspect an object can be automatically generated based on sensor planning techniques that determine appropriate camera and illuminator poses, optical settings, and image processing algorithms.

It is important to note that such planning techniques developed for vision sensors can also prove useful in other areas of automation, such as robotic grasp planning, automated machining, and dimensional inspection of mechanical parts [2], [22], [23]. For instance, accessibility of a surface in order to machine or probe it by tactile means is equivalent to visibility of this surface in the orthographic projection model. Planning in these latter domains of automation determines parameters such as workpiece and probe orientation, machine selection, and cutter or probe type selection.

C. An Overview of Our Approach: In this paper, we describe a vision planning system named machine vision planner (MVP) (see Fig. 1). MVP automatically determines vision sensor parameter values that satisfy certain feature detectability task requirements [24]–[28], [32], [33]. MVP includes methods that take as input the object geometry information from a CAD database, as well as models of the camera and the lens, and determine camera poses and settings for which features of interest of polyhedral objects are

1) Visible (occlusion-free positions of the sensor),
2) Resolvable by the sensor to a given specification,
3) In focus, and
4) Contained entirely in the sensor field of view.

Customarily, a viewpoint is considered to be the viewer location alone. However, it is useful to define the concept of a viewpoint in a broader sense that includes not only viewer orientation, but also the optical settings associated with the viewpoint at hand. These settings are also observer attributes affecting the resulting image of the scene. Examples of such attributes include the focus and aperture settings of a regular lens, the focal length settings of a zoom lens, as well as the diopter values for glasses of human observers. The parameters that are planned in MVP include the three positional degrees of freedom of the sensor—$\boldsymbol{r}_V(x_V, y_V, z_V)$—and the two orientational degrees of freedom, pan and tilt angles, described by a unit vector $\boldsymbol{\nu}$ along the viewing direction. The third orientational degree of freedom corresponding to rotation around the optical axis (i.e., the swing angle) is not considered, since the image plane is taken to be circular with a diameter equal to the minimum of the two image plane dimensions. In addition to the previous geometric parameters, three optical parameters are also considered, namely, the back nodal point to image plane distance d, the focal length f, and the entrance pupil diameter a of the lens. Thus, planning in MVP is done in eight-dimensional imaging space [21] and a point in this space is defined as a *generalized viewpoint* $\boldsymbol{V}(\boldsymbol{r}_V, \boldsymbol{\nu}; d, f, a)$.

In MVP, a *synthesis* approach is taken to be problem. In this approach, the task constraints are characterized analytically. Using concepts from geometry and optics, each task requirement is modeled by an equivalent analytical relationship which in turn is satisfied in a domain of admissible values in the space of parameters to be planned. Generally speaking, for each constraint, the admissible domain for sensor placement and settings is a region in eight-dimensional imaging space bounded by the hypersurfaces that are characterized by these analytical relationships.

These component admissible domains obtained for each task requirement separately are then combined in order to find parameter values that satisfy all constraints simultaneously. For this purpose the problem has been posed in an optimization setting in which a globally admissible eight-dimensional viewpoint is sought that is central to the admissible domain, that is, far from the bounding hypersurfaces described by the constraint equations. Such a generalized viewpoint is considered desirable since it is robust in the event of inaccuracy of either sensor placement or setting. The analytical relationship for each task constraint provide the constraints of the optimization, while the objective function is chosen so as to characterize the distance between the viewpoint and these bounding hypersurfaces.

Once a central generalized viewpoint is determined from the optimization, it is then realized in the actual sensor setup. In order to achieve these planned sensor parameter values, a mapping is established between the planned parameters (e.g., camera pose and optical settings) and the parameters that can be controlled (e.g., end effector pose, zoom, and focus settings). This mapping between the two parameter spaces is determined in the case of a sensor setup that consists of a camera in a hand–eye arrangement equipped with a lens that has zoom, focus, and aperture control. The optical system is modeled by a general thick lens model [9]. For such a lens model, the optical or perspective center of the lens defined in the pinhole model is replaced by the front and back nodal points. These are points on the optical axis whose properties are such that any ray passing through the front nodal point emerges from the back nodal point in the direction parallel to that of the original ray. Other characteristic properties of an optical system [12] that are employed include the aperture stop and the entrance and exit pupils. The aperture is the stop of the optical system that limits the angle of an axial cone of rays passing through the optical system. While the images of the aperture stop with respect to all lens elements preceding and following it are, respectively, the entrance and exit pupils. By maintaining such a generality regarding the optical model, the results are applicable to general lens systems such as a zoom lens and also it is possible to quantify the effect of any assumptions that may be made later. The resulting camera-lens-robot embody knowledge of the geometric relationships between the manipulator and the sensor and of the optical relationships among the various lens parameters.

The MVP sensor planning and sensor modeling techniques have been used to automatically synthesize camera views for a robotic vision system. This system consists of a camera mounted on a robot manipulator in a hand–eye configuration. The camera is positioned and the lens is focused according to the results generated by MVP. Camera views taken from the computed viewpoints verify that all feature detectability constraints are indeed satisfied.

Compared with other vision sensor planning systems [3], [4], [17], [19], [37], the MVP system is notable in that it takes a synthesis rather than a generate-and-test approach, thus giving rise to a powerful characterization of the problem. In addition, the MVP system provides an optimization framework in which constraints can easily be incorporated and combined. For example, while MVP has concentrated on the camera planning part of the problem and thus now lacks an illumination planning component, it would be relatively straightforward to include in such a framework. Compared with other approaches to the camera planning problem, MVP studies the broadest range of feature detectability constraints. Furthermore, the MVP system stands out for its generality of handling general viewing configurations and scene geometries (e.g., nonorthogonal viewing, general convex, and concave polyhedral objects) and determining a complete set of imaging sensor parameters as described by its concept of a generalized viewpoint. Another advantage of the MVP system is that it addresses the associated sensor modeling problems which are typically ignored or treated naively in other systems.

II. The Constraint Satisfying Viewpoint Loci

In this section, we briefly present the loci of admissible generalized viewpoints that satisfy each of the feature detectability constraints of visibility, resolution, focus, and field of view. Details and derivations of the following can be found in [24] and [26].

A. The Feature Visibility Constraint

The domain of admissible sensor locations is first limited to regions in three-dimensional space from where the features to be observed are *visible* (i.e., not occluded). We have developed techniques that compute the locus of all viewpoints from which features on known polyhedral objects can be viewed in their entirety without being occluded by anything in the environment (see Fig. 9). Convex and concave polyhedra with or without holes and the viewing model of perspective projection are employed in this work. Initially, properties of the occlusion-free and occluded loci of viewpoints are determined. Based on these properties, two methods to construct these loci together with their complexity analysis are presented. In the first method, a boundary representation of the occlusion-free locus is obtained. In the second method, the locus of occluded viewpoints is expressed in terms of a constructive solid geometry (CSG) representation that consists of a union of component solids. The features to be observed can be of any polyhedral type: a point, line segment, or face (including concave faces). The details of both techniques are described in [24].

Of these two methods, the second can be readily implemented in a solid modeling environment and thus is discussed here briefly. In this technique, the boundaries of the polyhedral objects in the environment are considered to be comprised of one or more polygonal faces, and the occluded region of a feature due to the environment as a whole (more precisely the union of the occluded region and the objects in the environment) is equal to the union of the component occluded regions generated by the individual object faces. In this method, the problem of computing the occluded region between a convex occluding polygon and a convex feature polygon is employed as a component step. A concave or multiply connected object face or feature is decomposed (i.e., partitioned) into convex polygons. As an example of convex partitioning, it is known that any simply or multiply connected polygonal domain can be triangulated [18]. The occluded region is the union of the component occluded regions generated between all pairs of convex polygons, one taken from the convex polygons of the object face and one from the convex polygons of the feature. Computing the occluded region in the case of a convex occluding polygon and a convex feature is relatively straightforward [4] and a linear time algorithm for this is presented in [26].

B. The Feature Resolution Constraint

Pixel resolution is used to indicate the approximate size of the smallest scene feature which can be seen by the vision system. In many machine vision tasks it is required that a particular unit feature size on an object appear as a minimum number of picture elements on a sensor. This feature resolution constraint can be satisfied by properly selecting the image sensor (e.g., pixel size), as well as by carefully planning its placement and settings. The objective of sensor planning for the feature resolution constraint is to determine the sensor parameters that achieve this resolution. The type of features that are considered for the resolution constraint are line segments, thus including feature edges, linear features of interest (e.g., width between two edges) or generally features that can be approximated by a finite set of line segments (e.g., a circular patch approximated by a finite set of diameters). In [24] we have obtained the locus of camera poses and lens optical settings for which chosen linear features are magnified to meet a given resolution specification, for instance, feature AB has an image of length that is at least equal to w. This locus of resolution satisfying generalized viewpoints is described in vector form by the following formula:

$$g_1\colon \frac{\|[(\boldsymbol{r}_A - \boldsymbol{r}_V) \times \boldsymbol{u}_1] \times \boldsymbol{\nu}\|}{[(\boldsymbol{r}_A - \boldsymbol{r}_V) \cdot \boldsymbol{\nu}][(\boldsymbol{r}_B - \boldsymbol{r}_V) \cdot \boldsymbol{\nu}]} - \frac{w}{dl} \geq 0 \qquad (1)$$

where $\boldsymbol{r}_V$, $\boldsymbol{r}_A$, $\boldsymbol{r}_B$ are the position vector of the front nodal point V of the lens and the vertices, A and B of the linear feature AB, with respect to the object world coordinate system, d is the distance from the back nodal point of the lens to the image plane, $\boldsymbol{\nu}$ is the unit vector along the optical axis in the viewing direction, $\boldsymbol{u}_l$ is the unit vector along the linear feature AB, l is the length of the linear feature AB, f is the intrinsic focal length of the lens, that is, the focal length of the lens for an object at infinity, and w is the required length of the image of the feature AB.

C. The Focus Constraint

When planning camera placement and lens settings so that all features of interest in a scene are in focus simultaneously, this corresponds to determining the locus of generalized viewpoints for which the feature points that are farthest and nearest with respect to each viewing direction lie within the range described by the depth of field. The feature vertex pair that is farthest and closest from the front nodal point of the lens, can be computed when a viewing direction is given. Thus, the region that satisfies the depth-of-field constraint for general polyhedral object features is the band between the planes perpendicular to the viewing direction, and at distances D_1 and D_2 from the farthest and closest feature vertices, respectively. Consequently, the functional relationship that describes the locus of generalized viewpoints satisfying the depth-of-field constraint is given in vector form by the following formulas:

$$g_{2a}\colon (\boldsymbol{r}_c - \boldsymbol{r}_V) \cdot \boldsymbol{\nu} - D_2 \geq 0 \qquad (2)$$

$$g_{2b}\colon D_1 - (\boldsymbol{r}_f - \boldsymbol{r}_V) \cdot \boldsymbol{\nu} \geq 0 \qquad (3)$$

where $\boldsymbol{r}_V$ is the position vector of the front nodal point V of the lens, $\boldsymbol{\nu}$ is the unit vector along the optical axis in the viewing direction. D_1 is the far limit of the depth of field, D_2 is the near limit of the depth of field, $\boldsymbol{r}_f$ is the position vector of the farthest feature vertex from the front nodal point of the lens along the viewing direction, and $\boldsymbol{r}_c$ is the position vector of the closest feature vertex from the front nodal point of the lens along the viewing direction.

The far and near limits of the depth of field are given by the following relationships [13], [24]:

$$D_1 = \frac{afd}{a(d-f) - cf} \qquad (4)$$

$$D_2 = \frac{afd}{a(d-f) + cf} \qquad (5)$$

where c is the limiting blur circle diameter that is considered acceptable, a is the diameter of the entrance pupil of the lens system, d is the distance from the back nodal point of the lens to the image plane, and f is the intrinsic focal length of the lens.

D. The Field-of-View Constraint

For the visibility constraint posed in Section II-A it was implicitly assumed that there were no field-of-view limitations, that is, the sensor was assumed to have an 180° field-of-view angle and therefore orientation of the sensor was immaterial provided that the features to be observed were in the half-space associated with the front of the camera. For a CCD camera, the field of view is generally limited by the minimum dimension $I_{\min}$ of the active sensor area in the image plane. Any observed feature must project onto this sensor area; otherwise, it will either be totally outside the field of view or will be truncated. It has been shown in [24] that the relationship describing the field of view satisfying locus of generalized viewpoints is given in vector form by the following formula:

$$g_3\colon (\boldsymbol{r}_K - \boldsymbol{r}_V) \cdot \boldsymbol{\nu} - \|\boldsymbol{r}_K - \boldsymbol{r}_V\| \cos\left(\frac{\alpha}{2}\right) \geq 0 \qquad (6)$$

where $\boldsymbol{r}_V$ is the position vector of the front nodal point V of the lens, $\boldsymbol{\nu}$ is the unit vector along the optical axis in the viewing direction, α is the field-of-view angle and is given by $\alpha = 2\tan^{-1}(I_{\min}/2d)$, $I_{\min}$ is the minimum dimension, width or height, of the active area of the sensor plane, d is the distance from the back nodal point of the lens to the image plane, and $\boldsymbol{r}_K$ is the position vector given by the relationship $\boldsymbol{r}_K = \boldsymbol{r}_C - R_0\boldsymbol{\nu}$, where $R_0 = R_f/(\sin\alpha/2)$ and $\boldsymbol{r}_C$ is the position vector of the center of a sphere circumscribing the object features and R_f is the radius of this sphere.

III. Integration of Constraints

A. Introduction

The approach taken to this point is bottom up. Each task constraint has been modeled in detail and independently of the other constraints, and thus an understanding has been developed regarding the effect of each on the general planning objective. At this point, these component admissible domains obtained for each task requirement separately are to be combined in order to find parameter values that satisfy all constraints simultaneously. For this purpose, the problem is posed in an *optimization* setting in which a globally admissible generalized viewpoint is sought that is central to the admissible domain. Such a viewpoint will, on the one hand, satisfy all the imposed constraints, but also, by favoring points far from the admissible domain boundaries, will do so with margin. Such a generalized viewpoint is desirable, since it is robust in the event of inaccuracy (e.g., due to sensor noise) of either camera placement or lens setting.

B. Admissibility of a Viewpoint

Constraint merging is formulated as a constrained optimization problem. The constraints of the optimization consist of the feature detectability requirements that must be satisfied. As discussed in the previous sections, these requirements are described by the visibility region and the analytical relationships given in (1)–(6). Consequently, in order to determine admissibility of a generalized viewpoint with respect to the resolution, field-of-view and depth-of-field requirements, it is sufficient to evaluate the following relationships:

$$g_1\colon \frac{d\|[(\boldsymbol{r}_A - \boldsymbol{r}_V) \times \boldsymbol{l}] \times \boldsymbol{\nu}\|}{[(\boldsymbol{r}_A - \boldsymbol{r}_V)\cdot][(\boldsymbol{r}_B - \boldsymbol{r}_V)\cdot \boldsymbol{\nu}]} - w \geq 0 \tag{7}$$

$$g_{2a}\colon (\boldsymbol{r}_c - \boldsymbol{r}_V)\cdot\boldsymbol{\nu} - D_2 \geq 0 \tag{8}$$

$$g_{2b}\colon D_1 - (\boldsymbol{r}_f - \boldsymbol{r}_V)\cdot\boldsymbol{\nu} \geq 0 \tag{9}$$

$$g_3\colon \frac{(\boldsymbol{r}_K - \boldsymbol{r}_V)\cdot\boldsymbol{\nu}}{\|\boldsymbol{r}_K - \boldsymbol{r}_V\|} - \cos\left(\frac{\alpha}{2}\right) \geq 0. \tag{10}$$

Compared to the resolution, field-of-view, and depth-of-field requirements where admissibility is determined by a function evaluation, visibility satisfaction, on the other hand, is determined by a point-in-polyhedron type classification, where the point is the location of the viewpoint under consideration and the polyhedron is the visibility region that has been constructed. Such a classification can be performed by shooting a ray from the point under consideration, computing the intersection of this ray with the faces of the polyhedron, determining whether the intersection point is inside each face, and keeping count of such face crossings.

An additional optimization constraint g_5 expresses the unit vector condition for the viewing vector $\boldsymbol{\nu}$ that appears in relationships g_1, g_{2a}, g_{2b}, and g_3:

$$g_5\colon \|\boldsymbol{\nu}\|^2 - 1 = 0. \tag{11}$$

This constraint g_5 is an equality, whereas g_1, g_{2a}, g_{2b}, and g_3 are all inequalities.

There is a g_1 equation for each edge feature that is to be resolved, while for the depth-of-field and field-of-view constraints, there is a single g_{2a}, g_{2b} and g_3 relationship for all features. For the depth-of-field constraint, auxiliary geometric queries are implemented in order to determine $\boldsymbol{r}_f$ and $\vec{r}_c$ for (8) and (9), where $\boldsymbol{r}_f$ and $\boldsymbol{r}_c$ are the farthest and closest feature points for the particular viewing direction that the optimization scheme is investigating at any given stage. A similar geometric query is implemented for the resolution constraint as described by relationship (7), in order to find for a given viewing direction the farthest minifeature on the given feature edge with length equal to the minimum size that is to be resolved.

C. Optimality of a Viewpoint

While the constraints address the admissibility of the computed solution, the optimization function on the other hand is constructed so as to characterize the quality of the computed solution, according to some metric. The metric used currently is robustness. That is, a solution is considered preferable, if it satisfies the constraints with a greater degree of tolerance (i.e., more comfortably). As a result, the measures used to assess the goodness of a solution with respect to the resolution, field-of-view, and depth-of-field constraints are the values of the constraint relationships g_i, $i = 1, 2a, 2b, 3$ themselves. This is appropriate since a large positive value of g_i indicates that the constraint is satisfied comfortably, a small positive value indicates marginal satisfaction, while inadmissible solutions give rise to negative values. More specifically, the value of the resolution constraint in the form given by relationship (7), expresses the difference between the actual and required lengths of the image of the minimum feature size. The values of the focus constraint in the form given by relationships (8) and (9), express the distances of the closest and farthest feature points from the near and far limits of the depth of field. The value of the field-of-view constraint in the form given by relationship (10), expresses the angular distance of the viewpoint from the boundary of the field-of-view locus. Specifically, it is the difference between the cosines of the field-of-view half angle, and the angle between the axis of the cone defining the field-of-view locus, and the vector from the apex of the field-of-view cone locus to the viewpoint.

It should be noted that a metric other than robustness may also be appropriate to assess the quality of a computed viewpoint. For example, in the previous discussion, viewpoints are considered more desirable with respect to the focus constraint if the features are located far from the depth-of-field limits.

This, however, does not necessarily lead to a more focused (i.e., less blurred) image. The reason for this is that the case where the features are situated close to the position of perfect focus, may be less robust in that the features will be closer to one of the two limits of the depth of field. This is especially true since the location of perfect focus is not symmetrically located with respect to the near and far limits of the depth of field.

Similar to the optical constraints, a measure of robustness of the computed viewpoint with respect to the visibility constraint, also needs to be formulated. For this purpose, the distance from the viewpoint to the polyhedron describing the visibility region is appropriate since such a metric corresponds to the maximum tolerance in camera placement that still avoids occlusion of the feature.

The distance d_0 from a point O to a polyhedron is defined as the minimum distance from the point to any of the faces of the polyhedron. The distance from the point to a face of the polyhedron is given by

1) d_Π, the distance from the point to the plane Π of the face, if the projection O' of the point O onto Π lies inside the face,
2) $(d_\Pi^2 + d_{e-o'}^2)^{1/2}$, if O' lies outside the face, where $d_{e-o'}$ is the minimum distance from O' to any of the edges of this face.

In turn for $d_{e-o'}$, the distance from a point to an edge is similarly given by

1) The distance from the point to the line of the edge, if the projection of the point onto this line, lies inside the edge
2) The distance to the closest of the two vertices of the edge, if the projection lies outside the edge.

The quality of the solution with respect to the visibility constraint will be expressed by the signed distance d_V, where the distance is taken to be positive if the viewpoint lies inside, and negative if the viewpoint lies outside the visibility region. Thus

$$g_4 = +d_V, \qquad \text{if } V \in V_R \tag{12}$$

$$g_4 = -d_V, \qquad \text{if } V \notin V_R \tag{13}$$

where d_V is the minimum distance from the viewpoint to the polyhedron describing the visibility region V_R.

Alternatively, an approximation of the distance d_V can be employed by drawing several rays from the viewpoint under consideration and determining the points of intersection between these rays and the boundary of the visibility region polyhedron. The distance measure will then be

$$g_4 = k \min_l d_{r_i} \tag{14}$$

where k is 1, -1 and 0 depending on whether the point is inside, outside or on the visibility region, respectively, r_i is a ray emanating from the viewpoint V, d_i is the length along the ray between the viewpoint and the boundary of the visibility polyhedron.

D. The Nonlinearly Constrained Optimization Problem

The optimization function is taken to be a weighted sum of the above component criteria, each of which characterizes the quality of the solution with respect to each associated requirement separately. Thus the optimization function is written as follows:

$$h = \max\left(\alpha_1 g_1 + \alpha_{2a} g_{2a} + \alpha_{2b} g_{2b} + \alpha_{3b} g_3 + \alpha_4 g_4\right) \tag{15}$$

subject to

$$g_i \geq 0, \qquad i = 1,\, 2a,\, 2b,\, 3,\, 4 \tag{16}$$

and

$$g_5 = 0. \tag{17}$$

The weights α_i in the above relationship are currently chosen so that the contribution of each constraint to the objective function is of the same order of magnitude at the start of the optimization. In this way, we avoid having a subset of the constraints dictate the optimization. Such a choice of weights may obviously have problems. For example, in the neighborhood of the local optimal and with the above choice of weights, the contribution of each constraint to the objective function may be of different orders of magnitude at the end of the optimization, thus leading to a biased optimization result. In general, determination of the appropriate weights in such a combining optimization function is a problem of its own [6] and many optimization routines (e.g., NCONF in IMSL [11]) include their own internal scaling procedures. In future work we shall investigate other weight settings that will depend on the particulars of the problem at hand.

Given the above formulation, the optimization starts with an initial point in the domain of possible generalized viewpoints and then generates a generalized viewpoint that is globally admissible and locally optimal as described by the optimization function. In other words, all constraints are satisfied with the largest margin in a neighborhood of the initial point.

This constrained optimization formulation constitutes a simple integration scheme based on the assumption that multiple and coupled objectives can be combined in an additive sense into a single global objective. Such a formulation has inherent problems with conflicting objectives which do exist in this sensor planning problem. For example, smaller values of the sensor parameter d increase the field of view on the one hand, but give rise to shorter (i.e., less magnified) scene features. What is truly desired is to have an image that is very much in focus, magnified, centered in the field of view, and far from being occluded. Thus, the problem may also be posed as one of multiple objective optimization with each task constraint constituting an individual objective. For simplicity the problem has currently been formulated as one of global objective optimization. The result of the optimization will be locally optimal only with respect to the chosen optimization function. This result will generally be different than the optimal generated by a multiple objective optimization approach. In order to address this discrepancy, it may be possible to determine values of the weights (i.e., weight calibration) for which the two solutions coincide in certain cases.

Fig. 3. Experimental setup.

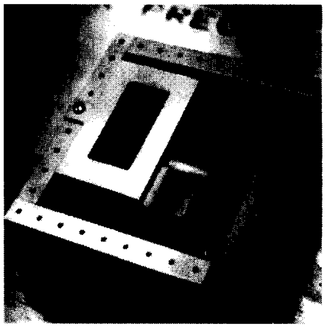


Fig. 4. The object.

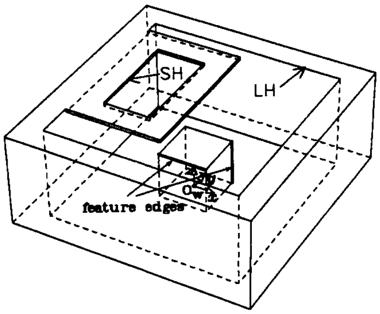


Fig. 5. CAD model of the object.

Techniques that combine the admissible domains of individual constraints in order to determine globally admissible solutions need to be investigated further. Optimization used in MVP provides one such framework. However, further work is needed in order to study convergence properties, appropriate weight values and the dependence of the final solution on the quality of the initial guess. Furthermore, techniques that determine admissible regions rather than an optimal point should also be investigated. For example, a description of interval-based techniques for solving large systems of nonlinear constraints is given in [8].

IV. Experiments

As part of the MVP system, we have implemented the machine vision planning algorithms that were discussed in the previous sections. In the experiments, we demonstrate the effectiveness of this approach using a robot vision system that plans its pose and the lens settings of its camera according to these techniques.

In this section, we present how these techniques are applied to the robotic setup in order to determine a globally admissible generalized viewpoint (i.e., camera pose and optical settings). We then describe placement and setting of the camera and demonstrate satisfaction of all feature detectability constraints.

A. Setup Description

The experimental setup is shown in Fig. 3. A Javelin CCD camera is fastened to the last joint of an IBM clean room robot (CRR). The CRR has two manipulators, each with seven joints, which consist of three linear joints (x, y, z), three rotary joints (roll, pitch, and yaw) and the gripper joint. The manipulator can place and orient the camera according to the parameter values determined by the planning system.

The Javelin CCD camera has 480 vertical and 384 horizontal sensor elements. The spacing between sensor elements in the horizontal and vertical directions are, respectively, 23 and 13.5 μm. The ratio of the sensor element spacing in the horizontal direction to the picture element spacing after sampling by the image acquisition hardware was found to be 0.70642 from calibration [14]. This horizontal scale factor relates the sensor element spacing to the pixel spacing in the frame buffer.

Several lenses covering a range of focal lengths are available to be mounted on the CCD camera as needed. In addition, since the back nodal point to image plane distance d is a parameter that is planned, a set of lens extension tubes are also available in the event that the value of this planned parameter lies outside the working range specified by the closeup focusing capability of the lens itself. A computer controlled zoom lens with programmable zoom, focus, and iris functions has also been used in [32], but not in a general manner. The reason for this is that currently the operating range of the sensor has not been incorporated in MVP as a constraint of the optimization. If the sensor operating range is ignored, the computed sensor parameter values are typically not simultaneously achievable by the sensor at hand. For example, the optical parameters f, d, and a, often cannot simultaneously assume the planned values. As part of the MVP sensor modeling work discussed in [25], general purpose techniques to characterize the operating range of a general lens have been presented. At this point, the resulting relationships that characterize the set of achievable sensor parameter values need to be included as additional constraints in the constrained optimization problem.

The object used in the camera placement experiments is shown in Fig. 4, and a CAD model of it is shown in Fig. 5. The features to be observed are the two edges of the enclosed cube shown in Fig. 5. This object is assembled from smaller primitive objectives (i.e., cubes, parallelpipeds, etc.) so that it can be reconfigured to test a variety of object geometry arrangements.

B. Sensor Planning Results

The domain of admissible camera locations is initially limited to the region in three-dimensional space from where the edge features to be observed are visible. The visibility algorithm discussed in Section II-A has been implemented in AML/X, an object-oriented programming language intended for use in design and manufacturing applications. The programs are run in the tiered geometric modeling system (TGMS) environment [5]. TGMS provides an object-oriented programming interface to an IBM in-house solid modeling system, geometric design processor (GDP) [34], as well as many geometry classes and methods. In this framework, the occluding objects, the feature, as well as the visibility and occluded regions themselves are represented as solids, and any operations on them (e.g., convex hull, Boolean set operations) are conveniently developed.

The visibility region for each edge feature is computed separately. These two regions are then intersected in order to determine the region in space from where both edges are simultaneously visible. First, the component occluded regions are computed between each feature edge and the inside vertical faces as well as the bottom face of the overhang. The other faces do not need to be considered since they are backfacing[1] and it has been shown in [26] that the occluded regions of backfacing faces are subsumed by the occluded regions of faces that are not backfacing. Since the features are convex and the inside vertical faces are also convex, their respective occluded regions can be easily obtained. On the other hand, since the bottom face of the overhang is multiply connected, that is, it consists of two loops, this face is triangulated. The occluded regions between the edge feature and each triangle are unioned, resulting in the occluded region between the bottom face of the overhang and the edge feature. The component occluded regions between the edge feature on the one hand, and the inside vertical faces and the bottom face of the overhang on the other, are unioned, resulting in the total occluded region for each edge. The complement of the occluded region is the visibility region for each edge, which are shown in Figs. 6 and 7. For this particular case it is simpler to conceptualize the visibility region for each edge as the union of the visibility regions associated with the two loops LH and SH, which correspond to viewing the target through the small hole SH and the large hole LH of the object (see Fig. 5). In Fig. 8 it can be noted that the visibility region for the vertical edge diminishes quickly in the vicinity of the narrow slit of the larger hole LH. The two edge visibility regions are then intersected resulting in the region shown in Fig. 9, that is, the space from where both edges are simultaneously visible.

Viewpoints chosen from this visibility region must also satisfy the other constraints in order to be globally admissible. For this experiment the resolution specification is taken to be 1 frame buffer pixel spacing for a minifeature size of $l = 0.1$ in. The pixel spacing mentioned in Section II-B refers to the distance between pixels on the sensor plane. An equivalent specification in terms of frame buffer pixels can be obtained using the scale factor that relates the sensor element spacing of the CCD array to the pixel element spacing of the frame buffer [14]. Furthermore, this spacing between sensor elements varies along different directions on the sensor plane. The maximum value corresponds to the diagonal direction of the sensor plane. This direction yields the worst case for the pixel resolution requirement since the image of the minimum feature length needs to be magnified the most in order to cover two diagonal pixels. We compute the sensor plane spacing in the diagonal direction corresponding up to 1 frame buffer pixel spacing as follows:

$$w = \sqrt{(23 \times .70642)^2 + 13.5^2} = 21.12\mu\text{m} \qquad (18)$$

where the spacing between sensor elements in the horizontal and vertical directions are, respectively, 23 and 13.5 μm and the ratio of the sensor element spacing in the horizontal direction to the picture element spacing after sampling by the image acquisition hardware is found to be 0.70642 from

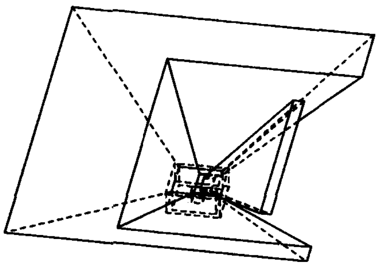


Fig. 6. Visibility region of the horizontal edge.

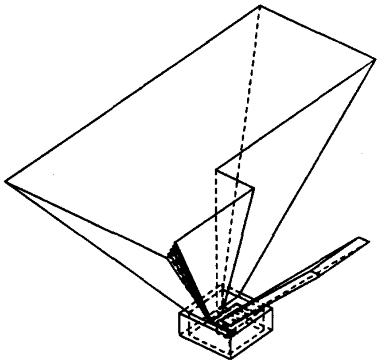


Fig. 7. Visibility region of the vertical edge.

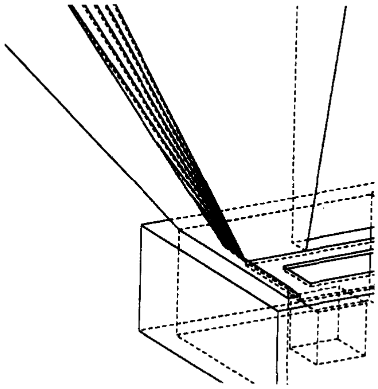


Fig. 8. Closeup of the visibility region of the vertical edge.

[1] Faces with an interior half-space that includes the feature in its entirety are named backfacing faces.

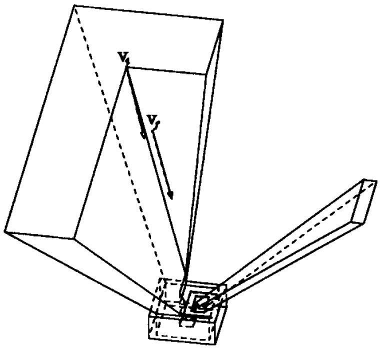


Fig. 9. Visibility region for both edges.

calibration [14]. For the field-of-view constraint, the minimum sensor plane dimension is $I_{\min} = \min(479 \times 13.5, 383 \times 21) = 479 \times 13.5 \simeq 6.5$ mm where 480 and 384 are the number of sensor elements in the vertical and horizontal directions, respectively, for the Javelin CCD camera at hand. The diameter of the circle of confusion for the depth-of-field constraint is taken to be the minimum of the horizontal and vertical sensor element spacings, that is $c = 13.5\ \mu$m.

With this information, the objective function h and the optimization constraints given by g_i, $i = 1, 2a, 2b, 3, 4, 5$, are constructed. From the two alternatives discussed in Section III-C regarding the constraint g_4, the exact signed distance d_V and not its approximation has been employed in the experiments since it did not adversely affect the speed of computation. In the experiments that follow, values of the entrance pupil diameter a and the intrinsic focal length f are chosen a priori ($f = 12.5$ mm and $a = f/16$). Thus values for the remaining parameters x_V, y_V, z_V, $\boldsymbol{v}$, and d are computed. The reason for setting the value of the focal length beforehand is that the available lenses cover a discrete set of focal length values. As a result, the focal length parameter cannot be handled in the optimization as a continuous variable assuming any value in a given range, like the other parameters. On the other hand, the reason for presetting the entrance pupil diameter is that it affects both focusing and image brightness. Since the later is not incorporated as a sensor planning constraint, if the entrance pupil diameter is planned, the resulting images are often either too dim or saturated. As a result, since the sensor and illumination planning problems are coupled, it is important to include an illumination planning component in this work.

The optimization is performed using the IMSL [11] nonlinear constrained optimization routine NCONF. This routine uses a successive quadratic programming method to solve the general nonlinear programming problem [20]. When using NCONF, adjustment of the initial guess point or the weights was needed in cases of premature program termination due to errors associated with attaining the maximum number of iterations, not satisfying the optimality condition due to roundoff errors, or not finding a search direction while the current iterate is still infeasible (the last indicates badly scaled problem functions [20]). It should be noted that NCONF assumes that all functions of the problem are assumed to be continuously differentiable. Strictly speaking this assumption is not satisfied by all the constraint relationships of our problem. For example, in the case of the focus constraint, as the viewing direction varies, different feature points become the farthest and closest points along this viewing direction, thus leading to a discontinuity in the associated constraint.

In the first experiment, the visibility constraint is not taken into account in the optimization function, that is, $\alpha_4 = 0$. However, the point-in-polyhedron classification is incorporated. The viewpoint determined by the optimization is classified with respect to the visibility region and thus satisfaction of the visibility constraint is checked. If the viewpoint lies inside or on the visibility region, then it is the viewpoint of choice, since it is a globally admissible generalized viewpoint that is locally optimal with respect to the resolution, depth-of-field and field-of-view constraints. However, if the viewpoint lies outside the visibility region, then the intermediate points that are generated by the optimization are checked for global admissibility, that is, satisfaction of all the constraints including visibility. From among these, the one with the largest value of the objective function is chosen. The values of the other weights in the optimization function are taken to be $\alpha_1 = 10^3$, $\alpha_{2a} = \alpha_{2b} = 10^{-2}$, and $\alpha_3 = 10^{-1}$, where distances are expressed in millimeters, and the constraints are written in the form given by relationships (1)–(6).

Both the initial and the computed camera viewpoints are listed in Table I, and are shown in Fig. 9 as points V_i and V_f, respectively, along with their associated viewing vectors. It can be seen from Fig. 9 that the initial guess viewpoint for the optimization is chosen to lie on an edge of the visibility region with a viewing vector in the direction from the viewpoint to the center of the sphere that circumscribes the features to be observed (see Section II-D). The viewpoint determined by the optimization V_f is classified with respect to the visibility region and is determined to lie inside the visibility region. Thus this viewpoint is chosen since it is both globally admissible and locally optimal with respect to the resolution, depth-of-field, and field-of-view constraints.

In the following experiments, the visibility constraint is taken into account in the optimization function, with the value of the associated weight $\alpha_4 = 1$. The values of the other weights in the optimization function are the same as in the previous experiment: $\alpha_1 = 10^2$, $\alpha_{2a} = \alpha_{2b} = 10^{-2}$, and $\alpha_3 = 1$, when distances are expressed in millimeters and the constraints are written in the form given by relationships (1)–(6).

The two initial viewpoints that are chosen to start the optimization, V_{i1} and V_{i2}, and the corresponding camera viewpoints that are computed by the sensor planning system, V_{f1} and V_{f2}, are listed in Tables II and III, and are shown in Fig. 10 along with their associated viewing vectors. It can be seen from Fig. 10 that the initial guess viewpoint V_{i1} is chosen to lie inside the visibility region with a viewing vector in the $-z$ direction. For V_{i1}, the field-of-view and focus constraints are violated. The initial guess viewpoint V_{i2} is chosen to lie outside the visibility region with a viewing vector again in the $-z$ direction. For V_{i2}, all constraints are violated. The viewpoints V_{f1} and V_{f2} that are determined by the technique

TABLE I
INITIAL AND COMPUTED GENERALIZED VIEWPOINTS V_i, V_f

	x	y	z	v(1)	v(2)	v(3)	f	d	a
V_i	165.1 mm	-279.4 mm	368.3 mm	-0.336	0.569	-0.750	12.5 mm	1.2f	f/16
V_f	107.58 mm	-231.43 mm	258.68 mm	-0.389	0.694	-0.608	12.5 mm	13.07 mm	f/16

TABLE II
INITIAL AND COMPUTED GENERALIZED VIEWPOINTS V_{i1}, V_{f1}

	x	y	z	v(1)	v(2)	v(3)	f	d	a
V_{i1}	228.6 mm	0.0 mm	368.3 mm	0.0	0.0	-1.0	12.5 mm	1.1f	f/16
V_{f1}	197.10 mm	100.58 mm	330.45 mm	-0.513	-0.255	-0.820	12.5 mm	12.84 mm	f/16

TABLE III
INITIAL AND COMPUTED GENERALIZED VIEWPOINTS V_{i2}, V_{f2}

	x	y	z	v(1)	v(2)	v(3)	f	d	a
V_{i2}	2.54 mm	2.54 mm	2.54 mm	0.0	0.0	-1.0	12.5 mm	1.1f	f/16
V_{f2}	112.78 mm	110.49 mm	159.77 mm	-0.444	-0.583	-0.680	12.5 mm	13.25 mm	f/16

can be seen in Fig. 10 to lie inside the visibility region with viewing vectors in the direction of the features.

C. Camera Placement, Lens Setting, and Verification of the Sensor Planning Results

In order to verify that all the constraints are indeed satisfied, camera views are taken from the computed viewpoints. Having determined from the sensor planning results of the previous section, the position vector of the front nodal point of the lens, the optical axis orientation, and the image plane to back nodal point distance, in this section we describe how the camera is placed and its lens is focused accordingly.

The manipulator with the mounted camera is used to place the camera at the chosen viewing positions with respect to the object. The computed camera position is known only with respect to the object world coordinate system. What needs to be determined is the manipulator location that places the camera at the chosen position. As discussed in detail in the Appendix, this manipulator location can be computed from the hand–eye relationship and the pose of the object in the robot world coordinate system. The object pose is determined by performing camera calibration [29] using features from the object itself and knowing their locations from the object model information. Details regarding the camera placement computations can be found in the Appendix.

On the other hand, the image plane to back nodal point distance *d*, which has been computed by the sensor planning system, needs to be realized in the camera-lens setup. From calibration of the lens [29], the image plane to back nodal point distances d can be determined for the $f = 12.5$ mm lens at its various focus settings. The value of this distance for the computed viewpoint V_f is $d = 13.07$ mm. This value lies outside the limits of the focusing capability of the lens, as determined by calibration. As a result a lens extension is used in order to give the necessary lens to image distance d.

Fig. 3 shows the manipulator placed at viewpoint V_f and oriented according to the computed viewing vector. The scene of the object taken from the computed viewpoint V_f is shown in both graphical simulation as well as in the robot vision system in Figs. 11 and 12, respectively. In both scenes, satisfaction of the visibility constraint can be readily verified. However,

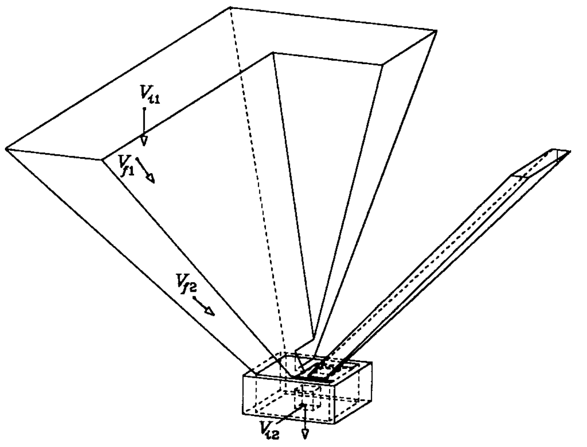


Fig. 10. Initial and the computed viewpoints V_{i1}, V_{i2}, V_{f1}, and V_{f2}.

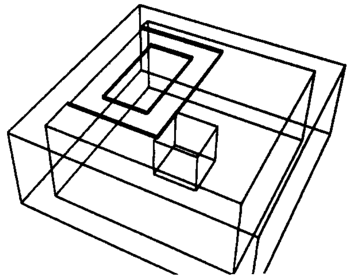


Fig. 11. Simulated view of the scene from the computed viewpoint V_f.

this constraint is satisfied only marginally since the horizontal edge is very close to being occluded by the overhang. This emphasizes the importance of including the contribution of the visibility constraint to the objective function in order to thus avoid viewpoints close to the visibility region boundary. The graphical simulation has not been set to the same field-of-view limits as the real camera; thus satisfaction of field-of-view constraint should be observed in the actual camera image. Furthermore, the resolution requirement is verified by measuring the feature magnification in the image (see Fig. 12). The 1-in horizontal and vertical feature edges are imaged as 45 and 39 frame buffer pixels, thus comfortably satisfying the 1 frame buffer pixel per 0.1-in requirement. Finally, the depth-of-field requirement can also be qualitatively verified in the image. A quantitative measure of focus will also be employed in future work for a more accurate evaluation of the quality of focus.

In a similar manner, the camera viewpoint is placed at points V_{f1} and V_{f2}, and the camera is oriented according to the computed viewing vectors. As seen in Table III, the value of the parameter d for the second computed viewpoint V_{f2} is $d = 13.25$ mm, which also lies outside the limits of the focusing capability of the lens. As a result, a lens extension is again used in order to give the necessary lens to image distance d. The scenes of the object taken from the computed viewpoints V_{f1} and V_{f2} are shown in both graphical simulation and in the robot vision system in Figs. 13–16. In all scenes, satisfaction of the visibility constraint can be readily verified. The graphical simulation has again not been set to the same field-of-view limits as the real camera, thus satisfaction of the field-of-view constraint should be observed in the actual camera images. Furthermore, by measuring the feature magnification in the image, it can be seen that the 1 frame buffer pixel per 0.1-in resolution requirement is comfortably satisfied for both viewpoints. Finally, satisfaction of the depth-of-field requirement can also be qualitatively observed in the image.

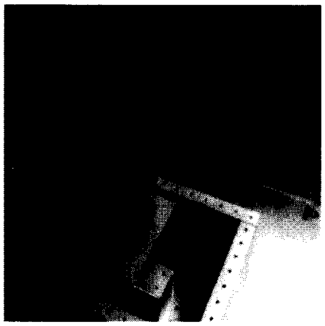


Fig. 12. Camera view of the features from the computed viewpoint V_f.

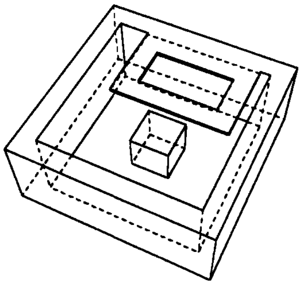


Fig. 13. Simulated view of the scene from the computed viewpoint V_{f1}.

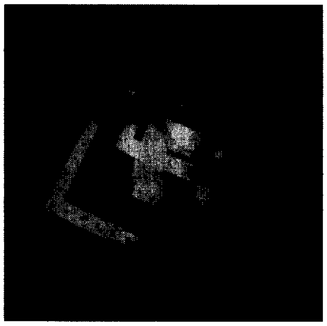


Fig. 14. Camera view of the features from the computed viewpoint V_{f1}.

V. Conclusion

We have presented the MVP system that automatically determines optimal camera poses and lens optical settings so that given visibility, field-of-view, depth-of-field, and resolution requirements are simultaneously satisfied with margin for chosen scene features. The problem is posed in an optimization setting

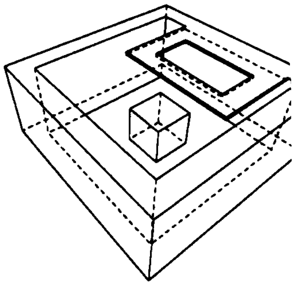


Fig. 15. Simulated view of the scene from the computed viewpoint V_{f2}.

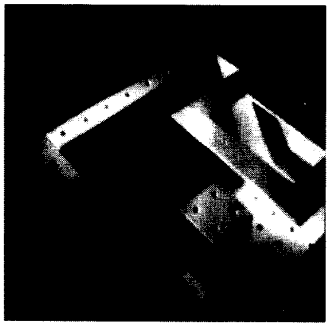


Fig. 16. Camera view of the features from the computed viewpoint V_{f2}.

and a viewpoint is sought that is both globally admissible and central to the admissibility domain. In the experiments, the technique determined viewpoints that satisfy all the above feature detectability requirements when starting from different inadmissible viewpoints.

In future work we shall extend this research to include illumination parameters (e.g., illuminator pose) and investigate how such parameters can also be planned so that features of interest are again robustly detectable in the resulting image.

These planning techniques can help automate robot vision systems that configure and reconfigure themselves in an intelligent manner in order to optimize imaging quality. Furthermore, the results discussed in this paper are useful for automating the vision system design process, as well as for programming the vision system itself.

VI. Appendix

A. Setting the Camera Pose

While the camera pose is the parameter that is planned, the parameter that can be controlled in the sensor setup is the robot gripper pose. In this section, the objective is to obtain values for the controllable parameter of the robot gripper pose given the desired values for the planned camera pose parameters, $\boldsymbol{r}_V$ and $\boldsymbol{\nu}$, and the relationship between the camera and the robot gripper.

Let us define the following Cartesian coordinate systems:

1) G is the gripper coordinate system which is fixed to the robot gripper;
2) C is the camera coordinate system which is centered at the front nodal point V of the lens, with the z axis coinciding with the optical axis, and the x, y axes parallel to the image X, Y axes;
3) O is the object world coordinate frame. This is an arbitrarily selected coordinate frame relative to which the coordinates of each point on the object are known;
4) W is the robot world coordinate frame. It is fixed in the robot workstation, and as the robot arm moves, the encoder output of all the robot joints determines where the gripper is relative to W.

The robot gripper pose can be expressed as the the transformation matrix H_{gw} from the robot gripper coordinate system G to the robot world coordinate system W. Similarly, the relationship between the robot gripper and the camera can be expressed by the transformation matrix H_{cg} from the camera coordinate system C to the robot gripper coordinate system G.

The desired values for the planned parameters $\boldsymbol{r}_V$ and $\boldsymbol{\nu}$ are expressed with respect to the object coordinate system and determine the desired geometric relationship H_{oc} between the camera and object coordinate systems. In addition, the relationship between the camera and the robot gripper expressed by the transformation matrix H_{cg} can be obtained from existing hand–eye calibration techniques. Thus the gripper pose with respect to the object coordinate system can be obtained from the following relationship:

$$H_{go} = H_{oc}^{-1} H_{cg}^{-1}. \tag{19}$$

What is needed, however, is H_{gw}, the gripper pose with respect to its own world coordinate system. This transformation matrix H_{gw} can be determined using a similar relationship, if in addition to H_{oc} and H_{cg}, the homogeneous transformation H_{wo} expressing the object pose in the robot workspace is also known:

$$H_{gw} = H_{wo}^{-1} H_{oc}^{-1} H_{cg}^{-1}. \tag{20}$$

In the following, we shall obtain H_{gw} by determining each of the transformation matrices on the right-hand side of the above equation.

H_{cg}: The problem of obtaining this relationship has been addressed in robotic hand–eye calibration. The hand–eye calibration technique that is used in the experimental part of this work is that of Tsai and Lenz [31].

H_{oc}: The transformation matrix H_{oc} can be obtained from the planned parameters $\boldsymbol{r}_V = (x_V, y_V, x_V)$ and $\boldsymbol{\nu} = (l_3, m_3, n_3)$ as follows:

$$H_{oc} \equiv \begin{bmatrix} & R_{oc} & & T_{oc} \\ 0 & 0 & 0 & 1 \end{bmatrix} \tag{21}$$

where the rotation matrix R_{oc} and the translation vector T_{oc} are given by

$$R_{oc} = \begin{bmatrix} l_1 & m_1 & k_1 \\ l_2 & m_2 & k_2 \\ l_3 & m_3 & k_3 \end{bmatrix} \tag{22}$$

$$T_{oc} = \begin{bmatrix} T_{oc_x} \\ T_{oc_y} \\ T_{oc_z} \end{bmatrix} = \begin{bmatrix} -(l_1 x_V + m_1 y_V + n_1 z_V) \\ -(l_2 x_V + m_2 y_V + n_2 z_V) \\ -(l_3 x_V + m_3 y_V + n_3 z_V) \end{bmatrix} \tag{23}$$

where (l_1, m_1, n_1), (l_2, m_2, n_2), and (l_3, m_3, n_3) are the direction cosines of the x_C, y_C, z_C axes of the camera coordinate system with respect to the x_O, y_O, z_O axes of the object coordinate system. The viewing vector $\boldsymbol{\nu}$ specifies only the direction cosines of the z_C axis. The extra orientational degree of freedom corresponding to the rotation of the $x_C y_C$ image plane of the camera around the optical axis can be arbitrarily set. For example, it can chosen so that the resulting gripper pose is feasible (i.e., lies within the robot workspace). In the experiments this degree of freedom is specified by taking the x_C axis of the camera coordinate system to be parallel to the $x_O y_O$ plane of the object coordinate system. In this case (l_1, m_1, n_1) and (l_2, m_2, n_2) can be obtained from the following relationships:

$$(l_3, m_3, n_3) \times (0, 0, 1) = \pm(1 - n_3^2)^{1/2}(l_1, m_1, n_1) \quad (24)$$

$$(l_2, m_2, n_2) = (l_3, m_3, n_3) \times (l_1, m_1, n_1). \quad (25)$$

It can be seen from the first of the above relationships that there are two possible solutions for the gripper pose. Each solution is checked whether the resulting gripper pose is attainable for the specific camera positioning gear at hand. With this information, the transformation matrix H_{oc} between the object and camera coordinate systems can be obtained from (21).

H_{wo}: Let us now address the last point in computing H_{gw} from (20); how to determine the position and orientation of the object with respect to the robot world coordinate system as described by the transformation matrix H_{ro}. This problem of object pose determination has received considerable attention in the literature. Since this problem is a research area in itself, a simple approach is taken in the experiments. The relationship H_{wo} is determined from a variant of (20). By observing the object from an arbitrary viewpoint and knowing the gripper and camera poses, as well as the camera-gripper relationship, the object pose H_{wo} can be computed from the following relationship:

$$H_{wo} = H_{oc}^{-1} H_{cg}^{-1} H_{gw}^{-1}. \quad (26)$$

This relationship is equivalent to that given by (20). Note the following in the above equation.

1) H_{cg} is known from hand–eye calibration.

2) H_{gw}, the location of the gripper within the robot workcell, can be obtained from the encoder output of all the robot joints.

3) H_{oc} is obtained by computing the camera pose with respect to the object coordinate system using existing camera calibration techniques (e.g., [29]). While camera calibration is customarily performed using special purpose objects, it is possible to use any known object with identifiable features. The association that is required between image and object features can be obtained in a feature matching step for example. In the experiments, the camera calibration method by Tsai [29] is applied using features (e.g., centers of circles, corners, etc.) of the object rather than a calibration pattern.

Having computed H_{wo}, H_{oc}, and H_{cg}, the manipulator position H_{gw} that achieves a particular camera pose can be computed from (20). The manipulator joint values that achieve this pose can be found simply using the calibration of the robot itself. There are several robot calibration techniques in the literature that can be used to map the gripper pose to joint values, at which point the robot can be commanded to the corresponding location and orientation.

REFERENCES

[1] A. K. Bejczy, W. S. Kim, and S. C. Venema, "The phantom robot: Predictive displays for tele-operation with time delay," in *Proc. 1990 IEEE Int. Conf. Robotics and Automat.*, 1990, pp. 546–551.

[2] L. L. Chin and T. C. Woo, "Computational geometry on the sphere with application to automated machining," Tech. Rept. 89-30, Industrial and Operations Engineering, University of Michigan, Ann Arbor, Aug. 1989.

[3] C. K. Cowan, "Model based synthesis of sensor location," in *Proc. 1988 IEEE Int. Conf. Robotics and Automat.*, 1988, pp. 900–905.

[4] C. K. Cowan and P. D. Kovesi, "Automatic sensor placement from vision task requirements," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 10, no. 3, pp. 407–16, May 1988.

[5] W. C. Dietrich Jr., L. R. Nackman, C. J. Sundaresan, and F. Gracer, "TGMS: An object-oriented system for programming geometry," *Software—Practice and Experience*, vol. 19, no. 10, Oct. 1989.

[6] M. A. Gennert and A. L. Yuille, "Determining the optimal weights in multiple objective function optimization," in *Proc. 2nd Int. Conf. Comput. Vision*, 1988, pp. 87–94.

[7] J. L. Gordillo and A. Lux, "Synthesizing vision programs from robot task specifications," in *Proc. 3rd Int. Symp. Robotics Res.*, 1985, pp. 149–154.

[8] G. D. Hager, "Interval-based techniques for sensor data fusion," Tech. Report, Yale University, New Haven, CT, 1990.

[9] B. K. P. Horn, *Robot Vision*. New York: McGraw-Hill, 1986.

[10] K. Ikeuchi and T. Kanade, "Modeling sensors: Toward automatic generation of object recognition program," *Comput. Vision, Graphics, and Image Processing*, vol. 48, pp. 50–79, 1989.

[11] IMSL Inc., IMSL Math/Library, 1989.

[12] R. Kingslake, *Optical System Design*. Orlando, FL: Academic, 1983.

[13] E. P. Krotkov, *Active Computer Vision by Cooperative Focus and Stereo*. New York: Springer-Verlag, 1989.

[14] R. K. Lenz and R. Y. Tsai, "Techniques for calibration of the scale factor and image center for high accuracy 3-D machine vision metrology," *IEEE Trans. Patt. Anal. Mach. Intell.*, vol. 10, no. 5, pp. 713–20, Sept. 1988.

[15] L. Lieberman, "Model-driven vision for industrial automation," in *Advances in Digital Image Processing*, Stucki, Ed. New York: Plenum Press, 1979, pp. 235–248.

[16] J. L. Mundy, "Industrial machine vision—Is it practical?" in *Machine Vision*, H. Freeman, Ed. Orlando, FL: Academic, 1988.

[17] R. Niepold, S. Sakane, T. Sato, and Y. Shirai, "Vision sensor set-up planning for a hand–eye system using environmental model," in *Proc. Soc. Instrum. Cont. Eng. Japan*, 1987, pp. 1037–1040.

[18] J. O'Rourke, *Art Gallery Theorems and Algorithms*. New York: Oxford University Press, 1987.

[19] S. Sakane and T. Sato, "Automatic planning of light source and camera placement for an active photometric stereo system," in *Proc. 1991 IEEE Int. Conf. Robotics and Automat.*, 1991, pp. 1080–1087.

[20] K. Schittkowski, "NLPQL: A fortran subroutine solving constrained nonlinear programming problems," *Ann. Oper. Res.*, vol. 5, no. 6, pp. 485–500, 1985.

[21] S. A. Shafer, "Automation and calibration for robot vision systems," Rept. CMU-CS-88-147, Carnegie Mellon University, Pittsburgh, PA, May 1988.

[22] A. J. Spyridi and A. G. Requicha, "Accessibility analysis for the automatic inspection of mechanical parts by coordinate measuring machines," in *Proc. 1990 Int. Conf. Robotics and Automat.*, 1990, pp. 1284–1289.

[23] K. Tang, T. C. Woo, and J. Gan, "Maximum intersection of spherical polygons and workpiece orientation for 4- and 5-axis machining," Industrial and Operations Engineering, University of Michigan, Ann Arbor, MI, 1990.

[24] K. Tarabanis and R. Y. Tsai, "Computing viewpoints that satisfy optical constraints," in *Proc. CVPR '91: The Comput. Soc. Conf. on Comput. Vision and Patt. Recognition*, 1991.

[25] K. Tarabanis and R. Y. Tsai, "Calibration of a computer controlled robotic vision sensor with a zoom lens," in *Proc. 1992 IEEE Int. Conf. Robotics and Automat.*, 1992.

[26] K. Tarabanis and Tsai, R. Y., "Computing occlusion-free viewpoints," in *Proc. CVPR '92: The Comput. Soc. Conf. on Comput. Vision and Patt. Recognition,* 1992.

[27] K. Tarabanis, R. Y. Tsai, and S. Abrams, "Planning viewpoints that simultaneously satisfy several feature detectability constraints for robotic vision," presented at *5th Int. Conf. Advanced Robotics (ICAR),* 1991.

[28] K. Tarabanis, R. Y. Tsai, and P. K. Allen, "Automated sensor planning for robotic vision tasks," in *Proc. 1991 IEEE Int. Conf. Robotics and Automat.*, 1991.

[29] R. Y. Tsai, "A versatile camera calibration technique for high accuracy 3D machine vision metrology using off-the-shelf TV camera and lenses," *IEEE J. Robot. Automat.*, vol. 3, Aug. 1987.

[30] R. Y. Tsai and M. A. Lavin, "Three-dimensional mechanical part measurement using a vision/robot system," Rept. RC 10506, IBM T.J. Watson Research Center, Yorktown Heights, NY, May 1984.

[31] R. Y. Tsai and R. K. Lenz, "A new technique for fully autonomous and efficient 3D robotics hand/eye calibration," *IEEE Trans, Robot. Automat.*, vol. 5, pp. 345–58, June 1989.

[32] R. Y. Tsai and K. Tarabanis, "Model-based planning of sensor placement and optical settings," in *Proc. Sensor Fusion II: Human and Mach. Strategies,* 1989.

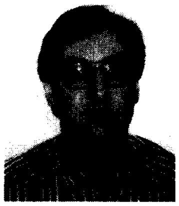
[33] R. Y. Tsai and K. Tarabanis, "Occlusion-free sensor placement planning," in *Machine Vision for Three-Dimensional Scenes*, H. Freeman, Ed. Orlando, FL: Academic, 1990.

[34] M. A. Wesley, T. Lozano-Perez, L. I. Lieberman, M. A. Lavin, and D. D. Grossman, "A geometric modeling system for automated mechanical assembly," *IBM J. Res. Develop.*, vol. 24, no. 1, pp. 64–74, Jan. 1980.

[35] W. J. Wolfe, D. W. Mathis, M. Magee, and W. Hoff, "Task panel sensing with a movable camera," *Proc. SPIE Intell. Robots and Comput. Vision VIII: Syst. and Applicat.,* vol. 1193, 1989.

[36] K. H. Womack, "Front-end foresight," *MVA/SME's Vision Technol. Quart.* p. 4, Jan. 1990.

[37] S. Yi, R. M. Haralick, and L. G. Shapiro, "Automatic sensor and light source positioning for machine vision," in *Proc 10th Int. Conf. Patt. Recognition,* 1990, pp. 55–59.



Konstantinos A. Tarabanis received the B.S. degree in mechanical engineering from the National Technical University of Athens in 1983, the M.S. degree in both mechanical engineering and computer science in 1984 and 1988, respectively, and the Ph.D. degree in computer science in 1991 from Columbia University, New York, NY.

He was employed by the IBM Corporation at East Fishkill during 1984–1985, working on thermomechanical modeling for electronic packaging. From 1988 to 1991 he had a research assignment at the IBM T. J. Watson Research Center, Yorktown Heights, NY, working on sensor planning for machine vision. He is currently a Research Staff Member with the IBM T. J. Watson Research Center. His current research interests include rapid prototyping, CAD-based vision, sensor planning, and geometric modeling.

In recognition of his work in sensor planning, Dr. Tarabanis was the recipient of the Anton Philips Best Paper Award at the 1991 IEEE International Conference on Robotics and Automation. He has served on the program committee for the 1994 CAD-based vision workshop.

Roger Y. Tsai received the M.S. degree from Purdue University and the Ph.D. degree from University of Illinois at Urbana-Champaign, both in electrical engineering, in 1980 and 1981, respectively.

He was employed by Bell-Northern Research/INRS-Telecommunications, Montréal, Canada, for three months during the summer of 1979 as a Visiting Scientist, working on moving image registration and enhancement. During the summer of 1980, he was a Visiting Scientist at the Signal Processing Group, EPFL, Lausanne, Switzerland, for three months working on 3-D time-varying scene analysis. In the summer of 1981, he again visited BNR/NRS, Montréal, for three months, working on image sequence analysis and computer vision. Since 1982, he has been with IBM Research Division, T.J. Watson Research Center, Yorktown Heights, NY, where he has been the Principal Investigator for the projects in the area of statistical process optimization and design of experiments, sensor planning, 3D SEM measurement, etc. His work has been applied extensively to IBM semiconductor and disk manufacturing process optimization, resulting in multimillion-dollar benefits.

He was the recipient of the Best Paper Award for the 1986 IEEE International Conference on Computer Vision and Pattern Recognition (CVPR) and the 1986, 1987, 1988, 1990, and 1991 IBM External Honor Recognition. He served as the Technical Editor of Robotics Vision and Inspection System for the IEEE JOURNAL OF ROBOTICS AND AUTOMATION from 1988 to 1991. He has served as chairman of the Computer Vision committee for the IEEE Society of Robotics and Automation from 1988 to 1991. He served on the program committee for the 1989 IEEE International Conference CVPR, and has served as session chairman for the 1986 IEEE International Conference on CVPR, and the 1987 and 1988 IEEE International Conferences on Robotics and Automation. He has been on the contributing editorial board for *Robotics Review*, MIT Press, from 1988 to 1992. He has served on the advisory committee for Sensor Fusion Conference, SPIE, an annual meeting held at Cambridge, MA, from 1989–1992. He was the coauthor for a paper that was awarded the Antone Philip Award at the 1991 IEEE International Conference on Robotics and Automation.

Peter K. Allen (S'82–M'85) received the A.B. degree from Brown University in mathematics–economics, the M.S. degree in computer science from the University of Oregon, and the Ph.D. degree in computer science from the University of Pennsylvania.

Currently he is an Associate Professor of Computer Science, Columbia University, and Director of the Center for Research in Intelligent Systems. His current research interests include real-time computer vision, hand-eye coordination, and model-based sensor planning.

While he was in graduate school, Dr. Allen received the CBS Foundation Fellowship, Army Research Office fellowship, and the Rubinoff Award for innovative uses of computers. In recognition of his work; he was named a Presidential Young Investigator by the National Science Foundation. Dr. Allen is a member of the Association for Computing Machinery and the AAAI.