An Illumination Planner for Lambertian Polyhedral Objects

Fredric Solomon and Katsushi Ikeuchi

CMU-RI-TR-94-26

(

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213-3890

October 4, 1994

Contents

List of Figures

Abstract

The measurement of shape is a basic object inspection task. We use a noncontact method to determine shape called photometric stereo. The method uses three light sources which sequentially illuminate the object under inspection and a video camera for taking intensity images of the object. A significant problem with using photometric stereo is determining where to place the three light sources and the video camera. In order to solve this problem, we have developed an illumination planner that determines how to position the three light sources and the video camera around the object. The planner determines how to position light sources around an object so that we illuminate a specified set of faces in an efficient manner, and so that we obtain an accurate measurement. We predict the uncertainty in our measurements due to sensor noise by performing a statistical simulation in our planner. This gives us the capability to determine when a measured shape differs in a statistically significant way from what we expect. From a high level, our planner has three major inputs: the CAD model of the object to be inspected, a noise model for our sensor, and a reflectance model for the object to be inspected. We have experimentally verified that the plans generated by the planner are valid and accurate. In most cases, the uncertainty predictions made by the planner were accurate to within 10%.

1. Introduction

The measurement of shape is a basic object inspection task. This measurement is performed in many modern manufacturing environments. However, the measurements are usually made manually. Manual inspection is monotonous, is very labor intensive, and is subject to human error. Some companies have turned to computer vision techniques.

Computer vision research has produced a number of basic techniques for measuring the shape of an object: stereo vision, range finders, and photometric techniques. Photometric techniques, unlike stereo, determine shape without needing to solve the correspondence problem. Photometric systems, unlike range finders, are cheap and are able to measure shape at a wide range of resolutions. For these reasons, we use photometric methods to determine object shape. Photometric techniques use physically based reflectance models [14] to transform image brightness into shape. Image brightness depends on lighting geometry, imaging geometry, and surface shape. If we can control imaging geometry and illumination geometry, we can use a reflectance model in conjunction with measured image brightness to determine surface shape.

Imaging geometry has been explored by a number of researchers. However, very few researchers have investigated how to illuminate an object that they are trying to inspect. In photometric stereo, the position of the light source affects what parts of the object are illuminated and the accuracy with which you can recover the object's shape. The problem is to determine the best place to put light sources in order to inspect a given object. If we arrange light sources in the shape of a tessellated sphere (say a N-frequency icosahedron), the optimal light source positions are not obvious, and the number of potential light source combinations is too large for a human operator to consider.

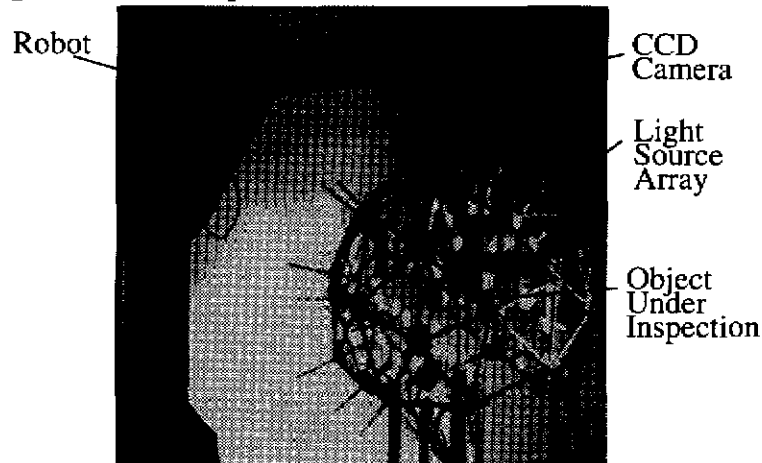Fig. 1. Experimental Setup

We investigate the illumination problem from two perspectives. First, we determine how to position light sources around an object so that we illuminate a specified set of faces in an efficient manner. In order to solve this problem, we have to determine light source visibility and we have to find some method of efficiently "covering" the specified set of object faces.

Secondly, we determine how to position the light sources so that they give us an accurate measurement of shape. There are two basic types of errors in photometric measurements of lambertian objects: random errors (noise) and fixed errors. Random errors are due to the variance of the camera and digitizer. These are the errors that we try to predict with our planner. Fixed errors include: errors in light source direction, errors in light source radiance, errors in the photometric function. Fixed errors can be accounted for by a careful calibration procedure.

Noise causes uncertainty in our shape measurement. The amount of uncertainty that this noise will create in our measurement will depend on light source positions. By using a noise model of the CCD, a CAD model of the object we are inspecting, and the lambertian reflectance model we can determine how much uncertainty we can expect in our shape measurement for a given light source configuration. We can find an optimal light source configuration, one that produces a minimum amount of measurement uncertainty.

Our ability to predict the uncertainty in our measurements gives us the capability to determine when a shape differs in a statistically significant amount from what we expect. This is an essential requirement for an inspection system because it allows defects to be reliably identified.

The environments we will study are structured. We will assume that we know what we are looking at (i.e.: We have a CAD model of the object that we want to inspect.) and that we know the pose of the object. This gives us a tremendous amount of information. We can plan our light placement to sequentially illuminate the entire surface of the object (cover the object) we are trying to inspect, and we can use our a priori geometric knowledge of the object to optimize our inspection parameters. This is a realistic scenario for many structured environments including manufacturing environments, nuclear plant maintenance, and space station missions.

The objects we look at are convex. So, we will be ignoring interreflection. While this may seem like a big restriction, we do not feel that the remaining problem is trivial. We assume orthographic projection and parallel incident light for our work.

Our work allows surface shape measurements to be made. The intent is that this information can be used for inspection. However, we do not become involved with setting up specific inspection criteria. Each inspection task requires unique inspection criteria, which must be determined on a case by case basis.

From a high level, our problem has three major inputs: the CAD model of the object we are trying to inspect, the noise model of our sensor (the noise model for our CCD camera), and

the reflectance model of the object we are trying to inspect. These models are used to generate an illumination plan.
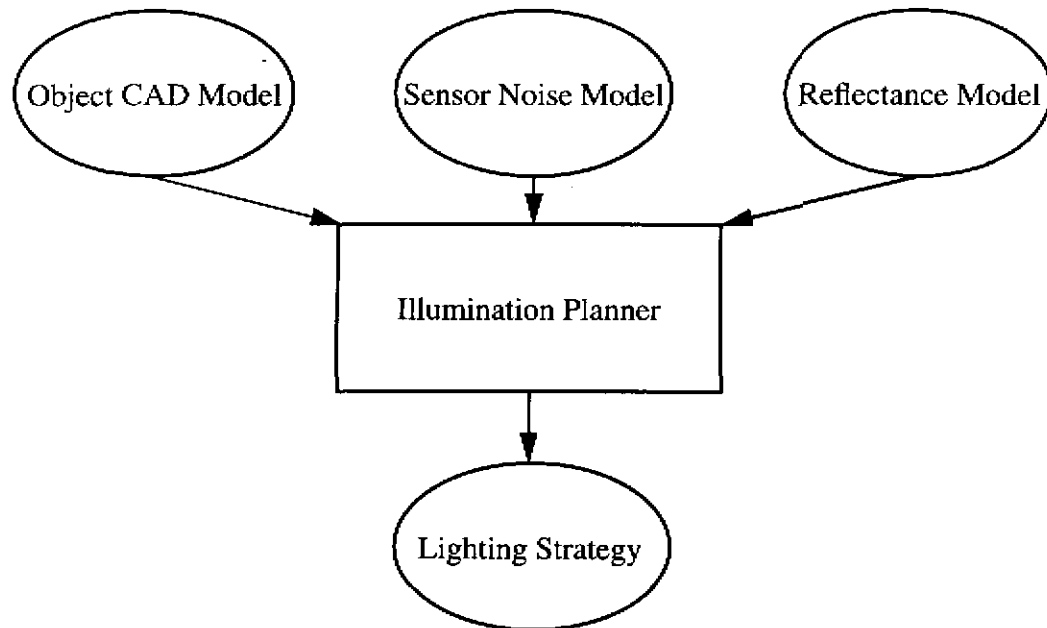

Fig. 2. Illumination Planning Data Flow

2. Previous Work

We summarize related CAD based inspection work. We cover work by Cowan, Tarabanis and Tsai, Sakane, Yi, and the SPIE.

2.1. Cowan

Cowan developed the synthetic approach to determining a camera's viewpoint. In [1], given a camera and lens, Cowan develops methods for determining 3-D camera locations that satisfy the following requirements: focus, field of view, visibility, view angle, and prohibited regions. For each requirement he builds a 3-D region that satisfies the requirement. Then, he intersects the regions to find camera locations that satisfy all the requirements.

The visibility constraint is the most relevant constraint to our work. Cowan determines visibility using the concept of "separating support planes". Given a convex polygon "S" and an occluding convex polygon "O", a separating support plane divides space into two halves. One half space contains S and not O. The other half space contains O and not S. The half planes are constructing by rotating a plane about each edge of S. The plane is oriented so it is between O and S, overlapping S. Then, the plane is then rotated about each edge of S until it hits a vertex or an edge of O. This is repeated for each edge of S, and for each edge of O. The union of all the half spaces containing S forms the set of viewpoints where O does not occlude S. The procedure can be extended to convex polyhedral objects and obstacles.
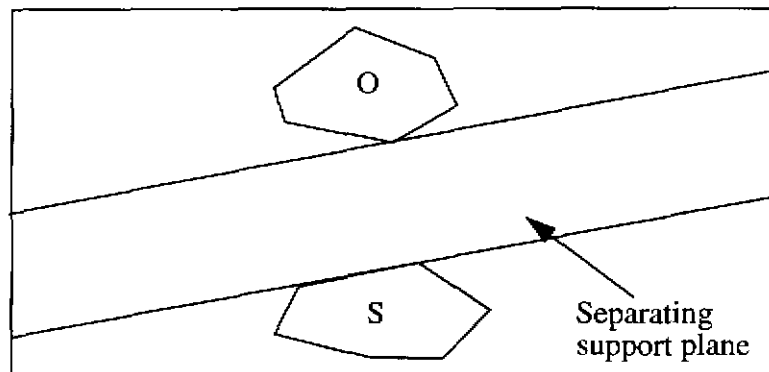
Fig. 3. Cowan's Separating Support Plane

In [2], Cowan discusses positioning point light sources for scene illumination. First, given a viewpoint region, Cowan finds the minimum and maximum camera apertures that bound the viewpoint region. Then, Cowan relates image irradiance to scene radiance and scene radiance to lambertian scene irradiance. Assuming a camera with a certain dynamic range, and a light source with a certain flux, he determines the minimum and maximum distance from the surface that the light source can be positioned.

In [3] Cowan reviews his previous work and extends it by considering the edge contrast of convex lambertian surfaces. He determines the possible source orientations and distances, with respect to a given convex corner (consisting of two lambertian surfaces), that produce a required intensity difference (contrast) between the two surfaces.

2.2. Tsai and Tarabanis

In [4], Tsai and Tarabanis develop a method for determining the visibility regions for general concave/convex polyhedral targets occluded by general concave/convex polyhedral obstacles. Their method, at its heart, is similar to Cowan's method. They have added on a convex decomposition algorithm, and have improved the computational efficiency of Cowan's original algorithm.

In [5], Tarabanis, Tsai, and Allen use the methods of [4] to search for visible regions where a feature can be viewed. Then, they determine the admissible camera locations where a feature can be resolved to a desired resolution. Locations are determined for orthogonal and general viewing directions.

In [6], Tarabanis, Tsai, and Allen expand on their previous work. They want to determine camera pose and optical settings so that a polyhedral object is visible, in the field of view, in focus, and resolved to sufficient resolution (four constraints). They pose the problem as an optimization problem in 8 dimensional space. The camera orientation and position contain five degrees of freedom. The distance from the back nodal point to image plane, lens focal length, and aperture are the three other degrees of freedom. Visibility is determined using the method described in [4]. The other constraints are posed as inequalities, and are merged into one optimization function with arbitrary weights assigned to each constraint. The function is then optimized using the visibility region boundaries as optimization constraints.

Tsai and Tarabanis and Cowan use a synthesis approach to sensor location. The sensor locations that satisfy a task are determined directly using optimization techniques. Sakane and Yi use a generate and test approach. Sensor locations are generated. Then, they are evaluated against a criteria function.

2.3. Sakane

In [9], Sakane describes a system that determines camera placement on a geodesic dome, that is placed around a target's point of interest. The camera is placed so that the point of interest is not occluded by other objects in the environment or by the manipulator holding the camera. Sakane determines occlusion free regions by doing depth buffering from the center of the dome to each facet. The minimum distance is stored at each facet. If the minimum distance is less than the radius of the dome, the viewpoint is occluded. The radius of the geodesic dome is chosen to get sufficient target resolution.

In [10, 11], Sakane describes positioning light sources for a photometric stereo system. He tries to optimize the extracted surface normal and the surface coverage for lambertian surfaces. Sakane uses the method of determining occlusion free viewpoints described in [9] to find candidate positions for the light sources. These positions provide shadow free illumination.

Sakane proposes a metric of surface orientation reliability that relies on the condition number of the source matrix to estimate the error in the surface normal vector (By definition, |N|

= 1.):

$$\frac{|\delta N|}{|N|} \le cond\,(S) \cdot \frac{|\delta I|}{|I|}$$

The second metric Sakane uses is the size of the intersection region of the three light sources and the camera on the gaussian sphere. Each light source will illuminate a hemisphere of the Gaussian sphere, and the camera will be able to observe a hemisphere of the gaussian sphere. By intersecting the four hemispheres, Sakane determines the amount of the gaussian sphere that is detectable.

Sakane combines the detectability metric and surface orientation reliability metric into a single criterion function. The weights of the two metrics are arbitrary. In [11], Sakane uses a movable camera. This allows a further degree of freedom in optimizing the detectability metric. In [12], Sakane considers lambertian edge contrast as a metric. Visibility seems to be determined using the methods described in [9].

2.4. Yi

In [13], Yi considers edge contrast for specular lobe objects, using a polarized light version of the Torrance Sparrow model. Yi forms a discrete spherical viewing space, with points positioned so that the arc length between viewing points is approximately equal. At discrete points along an edge Yi calculates the intensity difference. Then, he finds the "contrast distribution function" metric, which tells how much of the edge has a contrast above a certain threshold. The second metric is sensor visibility of the edge, which he defines as the ratio of the unoccluded portion of the edge to the entire edge length. Given a set of required edges, Yi searches for light source and sensor position which maximizes the two criteria functions. Yi does not describe how he combines the two criteria functions. The results presented in the paper are for a synthetic cylinder and cube. No real results are presented.

2.5. SPIE - Society of Photo-Optical Instrumentation Engineers

The SPIE (Society of Photo-Optical Instrumentation Engineers) has a large literature [23 - 25] on machine vision. Their approach is to solve real industrial problems. They have realized the importance of illumination for detecting defects in surfaces. However, their approach is best described as expert experience. For each inspection task, they formulate an illumination strategy, based on their own trial and error experience [23]. They have even developed expert systems, which give illumination strategies based on task specifications [24]. For example, one application is to view bright metal surfaces without causing glinting. Their solution is to use diffuse illumination. While their techniques are useful, they tend to be adhoc.

2.6. Summary/Conclusions

Cowan addressed the issues of visibility, imaging parameters, irradiance, and lambertian edge contrast. Tsai and Tarabanis addressed the issues of visibility and imaging parameters. Sakane addressed the issue of illumination in order to determine the shape of a lambertian object. Yi addressed the issue of finding specular lobe edge contrast.

The problem of finding edge contrast is very different than recovering surface shape using photometric methods. In order to generate edge contrast, one only needs to achieve a high intensity difference between the two surfaces that form the edge. This is very different from using intensity to find shape. Sakane's work is the closest to our research. There are at least three major differences between Sakane's approach and ours. First, we propose a new metric for finding orientation error. This is developed in section 3. Secondly, Sakane solves the problem of positioning lights to avoid objects in the environment or the manipulator holding the camera. We are positioning lights so that we illuminate a specified set of object faces in an efficient manner. Thirdly, we incorporate an accurate sensor model in our solution.

3. 2-D Lambertian Illumination

Lambertian illumination is important for two reasons. First, the illumination of lambertian surfaces is a basic problem. Lambertian surfaces are one of the three basic surface types. (Lambertian, specular lobe, and specular spike describe the basic surface types.) Solving the lambertian illumination problem will provide a foundation for approaching other, more difficult problems.

The 2-D problem presented here is simplified. However, it will provide insight into the solution space of the lambertian illumination problem. The solution space of the 3-D lambertian illumination problem, which we present later in this paper, is much more complex. It is very hard to visualize because of its high dimensionality.

3.1. Introduction

For our initial investigations, we have chosen a simplified version of the 2-D lambertian illumination problem. The problem does not consider surfaces, but considers discrete normals in a 2-D space. We concentrate on visibility, minimum coverage, and the accuracy of surface normal recovery. Our discrete normals exhibit lambertian reflectance characteristics. Our sensor noise model is a fixed amount of sensor noise. (In 3-D we will use a geometric modeler and a planning system to reason about surfaces. We will also use an accurate sensor noise model.)

Our approach is to identify the visibility range for each normal. After we have the visibility range for each normal, we break the viewing circle into visibility regions. Within each visibility region, certain normals are viewable (This idea is similar to an aspect.) We then try to find combinations of visibility regions that provide an exact cover of the normals we are trying to view. Since we are working in 2-D lambertian space, we need two light sources to recover the normals within each visibility region. After we have our minimum covers, we determine the most reliable lighting positions for recovering the orientation of the surface normals within each visibility region.

3.2. 2D Visibility Regions

Given a set of normals (n1, n2, n3, ...), we first find the visible range for each normal. Since these are discrete normals, we do not need to consider occlusion. So, the visible range of each normal is simply +/- 90° with respect to the normal's orientation. Once we have calculated the visibility range for each normal, we divide the 2-D viewing circle into visibility regions. The regions are formed by sorting the visibility ranges of all the normals into a continuous list. The list is then converted into intervals. Within each interval, certain normals will be visible. For example:

If we have normals at 45°, 90°, 135°, and 180° we have the following visibility

ranges:

normal designation	normal	visibility range designation	Visibility Range	
A	45°	VA	315.0°	135.0°
B	90.0°	VB	0.0°	180.0°
C	135°	VC	45.0°	225.0°
D	180°	VD	90.0°	270.0°

2D Visibility range/region diagram:
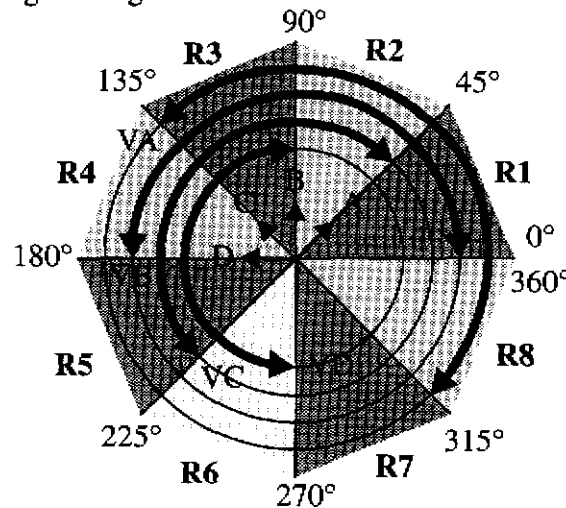
Fig. 4. Visibility Range/Region Diagram

The associated visibility regions are:

visibility region		region designation	normals visible			
0.0°	45.0°	R1		45.0°	90.0°	
45.0°	90.0°	R2		45.0°	90.0°	135.0°
90.0°	135.0°	R3	45.0°	90.0°	135.0°	180.0°
135.0°	180.0°	R4		90.0°	135.0°	180.0°
180.0°	225.0°	R5			135.0°	180.0°
225.0°	270.0°	R6			180.0°	
270.0°	315.0°	R7			none	
315.0°	360.0°	R8			45.0°	

3.3. 2D Exact Covers

After we have the visibility regions, we find combinations of visibility regions that form exact covers of our set of normals (An exact cover contains all of the normals, non-redundantly.) Since the number of regions is small, we use exhaustive search to find the exact covers. If "n" is the number of visibility regions, the complexity of this search is $O(2^n)$. (In the next section of the paper, we develop a heuristic for finding exact covers.) For the previous

example, the exact covers are:

Cover number	visibility regions included
1	R3
2	R1, R5
3	R2, R6
4	R4, R8

3.4. 2D Orientation Error

An error in light source illumination will cause an error in surface normal recovery. This can be seen from the lambertian equation:

$$\begin{bmatrix} S1x & S1y \\ S2x & S2y \end{bmatrix}^{-1} \begin{bmatrix} I1 \\ I2 \end{bmatrix} = \begin{bmatrix} Nx \\ Ny \end{bmatrix} \qquad (1)$$

(S1x and S1y are the x and y components of the unit vector to light source number one. Nx and Ny are the x and y components of the unit surface normal. I1 is the measured intensity at N due to light source number one.)

An error in either I1 or I2 will cause an error in Nx and Ny:

$$\begin{bmatrix} S1x & S1y \\ S2x & S2y \end{bmatrix}^{-1} \begin{bmatrix} I1 + \delta I1 \\ I2 + \delta I2 \end{bmatrix} = \begin{bmatrix} Nx + \delta Nx \\ Ny + \delta Ny \end{bmatrix} \qquad (2)$$

In matrix notation, we can write:

$$S^{-1}(I + \delta I) = N + \delta N$$

For example, if we have a normal at 90°, S1 at 100°, S2 at 80°, and $\delta I1$ and $\delta I2$ independently range from -0.1 to 0.1, we will get the following noisy normal distribution in Nx-Ny vector space (The nominal normal is the point Nx = 0.0, Ny = 1.0.):
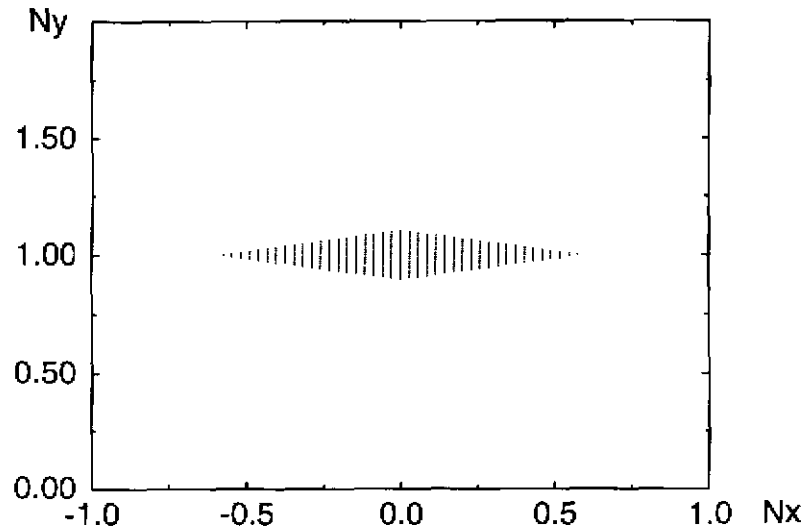

Fig. 5. Noisy Normal Distribution: N at 90°, S1 at 100°, S2 at 80°.

We normalize the resulting values of Nx and Ny because the surface normal is by definition a unit vector. The noisy, normalized, normal is:

$$N_{noise} = \frac{Nx + \delta Nx}{|N + \delta N|}, \frac{Ny + \delta Ny}{|N + \delta N|}$$

$$N_{noise} = Nx_{niose}, Ny_{noise}$$

When we do this, we perform a non-linear transformation from Nx-Ny vector space to the unit normal circle. The angle that the noisy normal distribution projects to, on the unit normal circle is the angular orientation error. This is shown below:
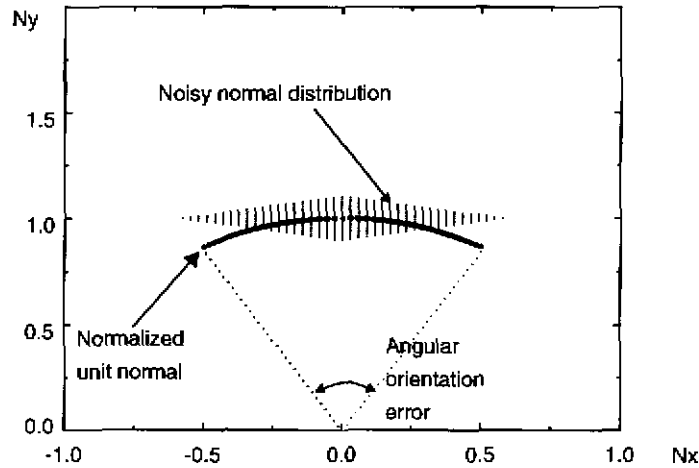
Fig. 6. Normalized Noisy Normal Distribution: N at 90°, S1 at 100°, S2 at 80°.

There may be little correlation between the magnitude of the normal vector's error in Nx-Ny vector space and the resulting angular orientation error. This point can be emphasized by looking at a second example. If we have a normal at 90°, S1 at 10°, S2 at 170°, and $\delta I1$ and $\delta I2$ independently range from -0.1 to 0.1, we will get the following normal distribution in Nx-Ny vector space (The nominal normal is the point Nx = 0.0, Ny = 1.0.):
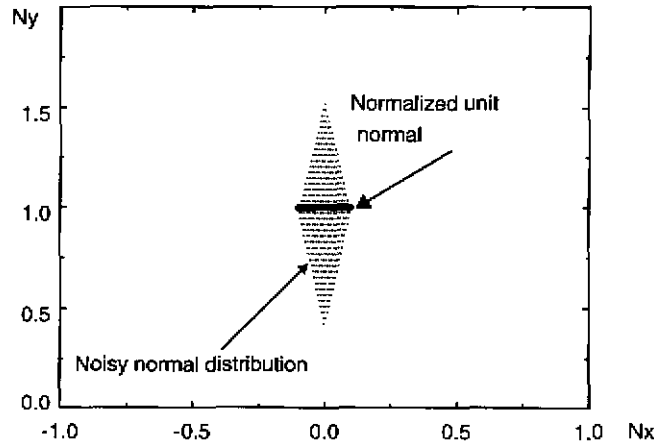

Fig. 7. Normalized Noisy Normal Distribution: N at 90°, S1 at 10°, S2 at 170°.

In the two figures, the worst case error between the nominal normal and the noisy normal

distribution is the same. The noisy normal distribution in the two figures is the same except for a 90° rotation. However, normalization causes the angular orientation errors to be very different. The angular orientation error is 29.9° in the first figure and is 5.8° in the second figure.

Sakane [10, 11] proposes a metric of surface orientation reliability that relies on estimating the unnormalized vector error. His method uses the condition number of the source matrix to estimate the error in the surface normal vector (By definition, $|N| = 1$.):

$$\frac{|\delta N|}{|N|} \leq cond(S) \cdot \frac{|\delta I|}{|I|}$$

As we have just seen, there is not necessarily any correlation between the magnitude of δN and the resulting orientation error.

If we look at our two previous examples: a normal at 90°, S1 at 100°, S2 at 80°, and a normal at 90°, S1 at 10°, S2 at 170°, we get the following results from Sakane's method (In both examples, $\delta I1$ and $\delta I2$ independently range from -0.1 to 0.1.):

> For the first example, the condition of the matrix is 5.67 (found using "svdcmp" of "Numerical Recipes in C"). The error is estimated to be $\leq .579$. This agrees with the magnitude of the worst case Nx error in our first plot.

> For the second example the condition of the matrix is also 5.67. The error is estimated to be ≤ 3.266. This is an over estimation, since the plots show that the two examples produce the same maximum δN.

> If we were to let a program search, in 1° degree increments, for the minimum error for the normal at 90° (the visibility range is between 0° and 180°). Sakane's method would yield a minimum at S1 = 91° and S2 = 1°. As can be seen from the plots, the minimum error should occur when S1 is close to 0° and S2 is close to 180°.

Due to these problems with Sakane's method, we developed another method for optimizing light source placement on lambertian surfaces. Our method calculates $N + \delta N$ for a given S, N, and δI. We normalize the resulting $N + \delta N$ vector (We call the normalized $N + \delta N$ vector N_{noise}.). Then, we find the angular orientation error between N and N_{noise}.

$$\theta_{err} = acos(N \bullet N_{noise})$$

For the first example, our method produces an error of 29.9°. For the second example, our method produces an error of 5.8°. These errors agree with the angular errors in our plots. If we let our program search for the minimum error light source positions, for a normal at 90°, within the 0° and 180° visibility range, we find the minimum error happens at S1 = 1° and S2 = 179°. (We are searching in 1° increments.)

We go back to our exact cover example, where we had normals at 45°, 90°, 135°, and 180°. We let $\delta I1$ and $\delta I2$ independently range from -0.01 to 0.01, and assess our ability to accurately determine surface orientation within each visibility range, of each exact cover. Each

visibility range covers certain normals. We move two light sources, in one degree incre-
ments, between the maximum and minimum visibility values of each visibility range. At
each position, we find the total angular error, in degrees, for the normals within the visibility
range. After searching through the whole visibility range, we find the light source placement
with the minimum error.

We get these results (Min_error is the sum of the angular errors, in degrees, for the normals
within the visibility range. The columns 45°, 90°, 135°, and 180° list the angular error, in
degrees, for each of these normals. Min_s1 and Min_s2 are the light source positions that
produced the minimum error.) :

cover	Visibility Range	Min_error	45°	90°	135°	180°	min_s1	min_s2
1	90.0 135.0	4.10	0.60	1.45	1.45	0.60	91.0	134.0
2	0.0 45.0	2.05	1.45	0.60			1.0	44.0
2	180.0 225.0	2.05			0.60	1.45	181.0	224.0
3	45.0 90.00	3.50	1.45	1.45	0.60		46.0	89.0
3	225.0 270.0	0.6				0.6	226.0	269.0
4	135.0 180.0	3.50		0.60	1.45	1.45	136.0	179.0
4	315.0 360.0	0.60	0.60				316.0	359.

A tessellated error surface for cover 1 is show below. The error is greatest when the light
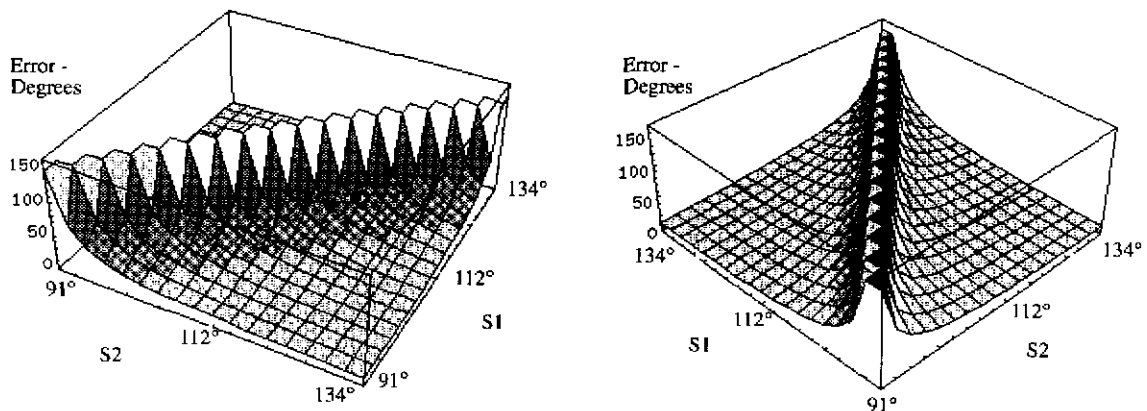sources are close together and is least when the light sources are far apart.

Fig. 8. 2D Angular Orientation Error Surface

4. 3-D Illumination Covers

Finding illumination covers in 3D is similar to finding illumination covers in 2D. However the 3D problem is more complex. Given a 3D convex polyhedral object, and a set of faces on the object that we want to inspect, we want to place light sources around the object in a way that illuminates the set of object faces that we want to inspect, while minimizing the total number of measurements we need to make.

In order to illuminate an object under inspection, we need to be able to divide the viewing sphere into aspects. Each aspect describes topologically connected viewing directions that see the same set of object faces. Eventually, aspects need to be combined in some manner that provides coverage of all the object faces that need to be examined. (We define our inspection task to be inspection of a specified set of object faces. This set may include all object faces, or it may be a proper subset of all object faces.) A combination of aspects that views all specified faces of an object in a non-redundant manner is called an exact cover. There are two types of aspects: aspects that are formed from viewpoints near the object and aspects that are formed by viewpoints far from the object. For now, we are concentrating on aspects that are formed from viewpoints far from the object.

We generate CAD models of objects that we want to inspect using the "Vantage" geometric modeling system [30]. Using these models and the 3D-to-2D scene generation capabilities of Vantage, we can generate orthographic projections from various viewpoints of a tessellated icosahedron. By comparing the area of each visible object face with its foreshortened projected area, we can determine which object faces are completely visible from each viewpoint. If the face is more than 99% visible, we consider the face to be completely visible.

Once we know which object faces are visible from each icosahedron viewpoint, we merge adjacent icosahedron viewpoints that view the same set of object faces. The resulting merged set of icosahedron viewpoints forms an aspect. Each aspect is a continuous viewing region. A camera or light source placed anywhere within the region will view the same set of object faces.

Given the list of faces visible from each aspect and a list of all the faces on the object that we are inspecting, we can determine combinations of aspects that form exact covers.

4.1. 3D Aspect Generation

Koenderink and van Dorn [15] described aspects and their associated data structure the visual potential (aspect graph). Since then, many researchers [22] have explored aspect graphs, and their application to computer vision tasks. The primary application of aspect graphs to computer vision has been object recognition [16, 17]. Aspect graphs are used to predict potential object appearances. Then, sensor measurements which can recognize an object based upon the set of possible object appearances and sensor measurements can be planned. There are two types of aspect prediction possible exact aspect graphs and approximate aspect graphs. Exact aspect graphs [18-21] use a 3D model of an object to predict where in viewpoint space visual events occur. Once the location of a visual event is known, viewpoint space can be partitioned into regions. Each region will correspond to an aspect.

Exact aspect graphs are mathematically complete. They describe every aspect that an object produces, no matter how small. However, producing an exact aspect graph for general objects is a non-trivial problem which is still an area of active research.

The approximate aspect graph approach [16, 17] uses a tessellated viewing sphere that surrounds the object. Views from each viewpoint on the viewing sphere are generated. Views that are topologically equivalent are merged into aspects. The approximate aspect graph method, unlike the exact aspect graph method, will work on all classes of objects. By using this approach, we will be able to inspect a large class of objects.

The approximate aspect graph approach misses aspects that are close to the object being viewed (usually the viewing sphere is much larger than the object). The method may miss small aspects if the tessellation is not fine enough, and the approximate aspect graph approach is computational intensive.

For our application, aspects that are close to the object are not important. We are trying to perform a macroscopic level inspection of an entire object, not a microscopic inspection of a partial face. Small aspects are undesirable because they are unstable. Computational load is not issue because the inspection plan can be generated off-line.

Aspects are formed by using an 3-frequency (320 face) icosahedron. The icosahedron has a radius equal to 1000 times the largest diameter of the object being inspected. Views are generated using orthographic projection.

4.2. 3D Exact Coverage

The problem of finding all exact covers of a set of faces is equivalent to the set/subset exact cover problem. This problem is known to be NP-complete. In order to decrease the complexity of finding exact covers, we developed a heuristic search approach to the problem Our heuristic measure is: the number of object faces covered by the aspect. Our algorithm is as follows:

1. Select an object face.
2. Find the largest aspect, $Amax_1$, (aspect that sees the most object faces) that contains this object face.
3. Delete any aspects that overlap $Amax_1$, including $Amax_1$, from the aspect-list.
4. $n = 1$.
5. Find the largest remaining aspect in the aspect-list, $Amax_n$.
6. See if the combination of $Amax_1$, $Amax_2$, ... , $Amax_n$ covers all object faces. If so, return the exact cover: $Amax_1$, $Amax_2$, .., $Amax_n$.
7. If not, delete any aspects that overlap $Amax_n$, including $Amax_n$, from the aspect-list.
8. If the aspect-list is empty, signal failure and return.
9. $n = n + 1$.
10. Go to step 5.

In steps 2 and 5, if there are two or more aspects that are the largest aspects, we trace out all the largest aspects in parallel. We can expand our criteria for finding the next aspect to expand in steps 2 and 5 to include the largest aspects, the 2nd largest aspects, ..., the qth largest aspects. As q increases, the algorithm will approach exponential complexity.

The maximal aspect heuristic was chosen because we are trying to minimize the number of sensor measurements needed to cover an object. By choosing aspects that cover as many object faces as possible, we maximize the coverage of each sensor measurement. This should tend to minimize the number of sensor measurements needed.

We do not offer a formal proof that our algorithm will succeed for all objects. (The success of the algorithm is dependent on the tessellation of the viewing sphere since some aspects might be missed if the tessellation is not fine enough.) However, we can say that as our search expands to include the qth largest aspect, the aspect size will in the worst case approach the size of a single face. This is possible in the case of cube, where it is possible to have singular aspects that contain each cube face. Then, an exact cover can be formed by combining these single face aspects. As object complexity increases, the object will eventually approach a tessellated sphere. In this case, two views, 180° apart, provide an exact cover.

In order to test our algorithm against the set of exact covers, we compared our level one search (only the largest aspects), with deeper level searches (largest aspects, the 2nd largest aspects, ..., the qth largest aspects) for the following objects:.
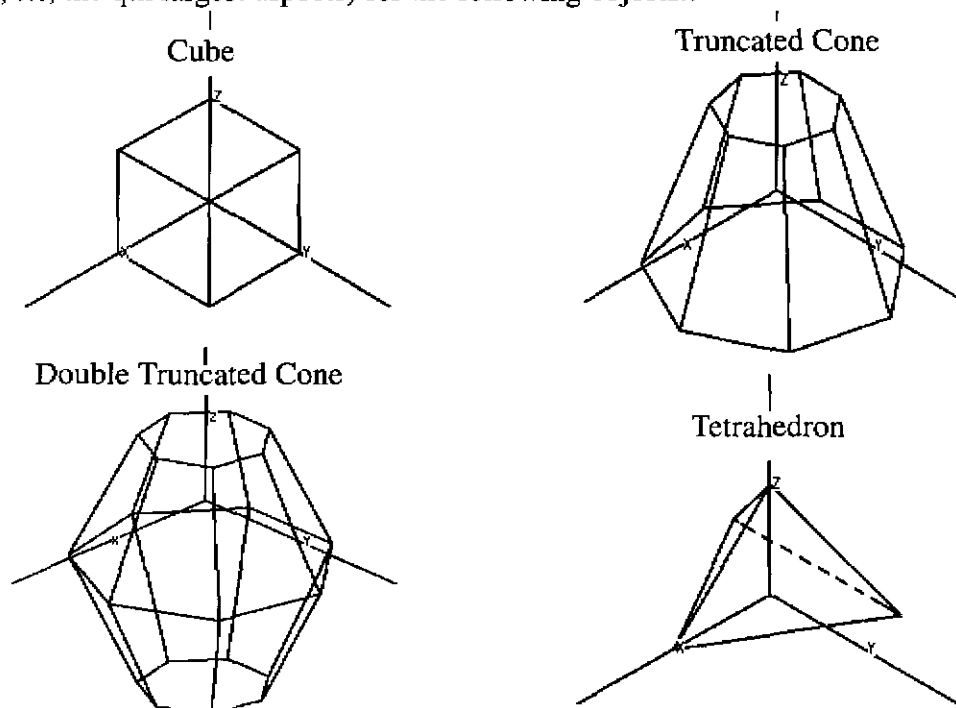

Fig. 9. Convex Polyhedra Used for Exact Cover Tests

The covers in Table 1 include aspects made up of any number of icosahedron viewpoints. For photometric stereo, we would need to place three light sources in an aspect. Therefore,

any aspect would need to be made up of at least three icosahedron viewpoints. The results in Table 2 are for photometric stereo. Only aspects made up of at least 3 icosahedron viewpoints are allowed to be part of an exact cover.

Table 1: Exact Covers - All Aspects

Search Depth	cube	tetrahedron	truncated cone	double truncated cone
1	4	12	41	79
2	68	21	45	79
3	68	23	45	79
4		23		

Table 2: Exact Covers for photometric stereo- Aspects of size 3 and greater.

Search Depth	cube	tetrahedron	truncated cone	double truncated cone
1	4	7	13	21
2	4	13	13	21
3	4	14	13	21

The reason that we do not see a jump in the number of aspects for the cube when we increase the search depth from 2 to 3 is that the icosahedron's viewpoints do not line up with the cube's faces. So, we do not have any aspects that see only one object face. We only have aspects that see two or three object faces. The single object face cube aspects are actually accidental aspects. They are not really useful because they occur at one exact viewing direction. They are very unstable. In the case of the tetrahedron, we have a couple of aspects that see only one object face. These aspects are large, containing many icosahedron viewpoints. We also have aspects that see one object face, two object faces, and three object faces. Therefore, we have more jumps in the number of aspects for the tetrahedron than for the cube.

5. 3D Lambertian Illumination

Light source positioning in 3D is similar to light source positioning in 2D. The goal is to find the positions for three light sources that illuminate a given set of object faces while minimizing the "total orientation error" for the object faces. An exact illumination cover for a given set of object faces and a candidate set of light source positions is found using the heuristic developed in the previous section. Then, for each illumination aspect, three light sources are placed in all of the combinations of light source positions that comprise the illumination aspect. The number of combinations is "N choose 3", since we are placing three light sources in the N light source positions that comprise the aspect. The "total orientation error" is evaluated for each combination. The light source positions that produce the minimum total orientation error is returned by the planner.

5.1. 3D Orientation Error

As we developed in the two dimensional case, an error in light source illumination will cause an error in surface normal recovery. This can be seen from the lambertian equation:

$$
\begin{bmatrix} S1x & S1y & S1z \\ S2x & S2y & S2z \\ S3x & S3y & S3z \end{bmatrix}^{-1} \begin{bmatrix} I1 \\ I2 \\ I3 \end{bmatrix} = \begin{bmatrix} Nx \\ Ny \\ Nz \end{bmatrix}
$$

I1, I2, and I3 are the mean, measured, light source intensities. Noise in either I1, I2, or I3 will cause an error in Nx, Ny, and Nz, producing a noisy normal:

$$
\begin{bmatrix} S1x & S1y & S1z \\ S2x & S2y & S2z \\ S3x & S3y & S3z \end{bmatrix}^{-1} \begin{bmatrix} I1 + \delta I1 \\ I2 + \delta I2 \\ I3 + \delta I3 \end{bmatrix} = \begin{bmatrix} Nx + \delta Nx \\ Ny + \delta Ny \\ Nz + \delta Nz \end{bmatrix}
$$

In matrix notation, we can write:

$$
S^{-1}(I + \delta I) = N + \delta N
$$

The noisy, normalized, normal is:

$$
N_{noise} = \frac{Nx + \delta Nx}{|N + \delta N|}, \frac{Ny + \delta Ny}{|N + \delta N|}, \frac{Nz + \delta Nz}{|N + \delta N|}
$$

$$
N_{noise} = Nx_{niose}, Ny_{noise}, Nz_{noise}
$$

In the two dimensional case, the Nx-Ny error area was a quadrilateral. In three dimensions, the Nx-Ny-Nz error volume will be an eight sided polyhedra. In two dimensions, after normalization, the normal was projected onto the unit normal circle. In three dimensions, after normalization, the normal is projected onto the unit normal sphere.

We define the orientation error to be the angle between the nominal unit normal and the

noisy unit normal. The orientation error depends on how we define the δI terms. Initially we defined the worst case error to be:

$$\delta I = \pm 3\sigma_i$$

By letting I1, I2, and I3 independently take on a worst case $+3\sigma_i$ or $-3\sigma_i$ error, we generated 8 potential worst case errors. However, the probability of having I1, I2, and I3 simultaneously have 3σ values is very small (approximately 0.013^3).

A more realistic method of determining the error is to have the planner conduct a simulation using the intensity noise function. If we know, the mean intensity of each light source (This can be determined if the object's pose is known, if the light source directions are known, and if the light source radiances are known.) and the corresponding value of σ_i, we can calculate a noisy surface orientation using the known light source positions. At each iteration, a random intensity for each light source is determined based on the mean light source intensity and σ_i for that light source intensity. These three noisy intensity values are used to determine a noisy surface orientation. We repeat this 1000 times. We calculate the mean, noisy, surface orientation to be the center of mass of the 1000 noisy surface orientations:

$$\bar{N}_{noise} = \frac{\sum\limits_{i=1}^{1000} [Nx_{noise}]_i}{1000}, \frac{\sum\limits_{i=1}^{1000} [Ny_{noise}]_i}{1000}, \frac{\sum\limits_{i=1}^{1000} [Nz_{noise}]_i}{1000}$$

Using each noisy surface orientation and the mean, noisy, surface orientation, we calculate the orientation error.

$$\theta_{err} = acos\,(\bar{N}_{noise} \bullet N_{noise})$$

Using the 1000 values of orientation error that our simulation produced, we then determine the mean surface orientation error, $\bar{\theta}_{err}$, and its standard deviation, $\sigma(\theta_{err})$.

The "total orientation error" for each illumination cover is the sum of the errors for each face illuminated by the cover. Where the error for each face is the mean orientation error + 3 standard deviations. For n faces:

$$\theta_{totalerr} = \sum\limits_{i=1}^{n} [\bar{\theta}_{err}]_i + 3\,[\sigma(\theta_{err})]_i$$

5.1.1. Source Normalization

In any real system, the light sources will have different radiances. This will affect the value of $\sigma(\theta_{err})$. If the three light sources used to determine N do not have the same radiances, the three raw intensity values (I1, I2, and I3) need to be normalized. We normalize to the minimum light source radiance. If we define minimax to be the minimum light source radiance of the three light sources that we are using, then for three measured intensities, our normal-

ization is:

$$i1_{norm_{meas}} = (\frac{minimax}{i1_{max}}) i1_{meas}$$

$$i2_{norm_{meas}} = (\frac{minimax}{i2_{max}}) i2_{meas}$$

$$i3_{norm_{meas}} = (\frac{minimax}{i3_{max}}) i3_{meas}$$

where $i1_{max}$ is the radiance of light source one.

The normalization process affects the variance of the measured intensities and therefore affects $\sigma(\theta_{err})$. If the planner has knowledge about light source radiances, then it can take this information into account when it predicts what $\sigma(\theta_{err})$ we can expect.

The planner will predict the mean light source intensities for a face based upon the face's assumed pose, and the known light source radiances:

$$\overline{i1_{plan}} = i1_{max}(S1 \bullet N)$$

$$\overline{i3_{plan}} = i3_{max}(S3 \bullet N)$$

$$\overline{i2_{plan}} = i2_{max}(S2 \bullet N)$$

where **S1** is the direction of light source one, and **N** is the normal of the face.

These mean intensities should be the intensities that irradiate the CCD of our camera. These mean intensities determine the variance of our measurement. So, for each mean intensity, our planner determines the variance (actually the standard deviation) that we expect to measure. We will call these three values $\sigma_{\overline{i1}}$, $\sigma_{\overline{i2}}$, $\sigma_{\overline{i3}}$.

However, the normalization of the intensity values affects these variances. So that:

$$\sigma_{i1_{plan}} = (\frac{minimax}{i1_{max}}) \sigma_{\overline{i1}}$$

$$\sigma_{i2_{plan}} = (\frac{minimax}{i2_{max}}) \sigma_{\overline{i2}}$$

$$\sigma_{i3_{plan}} = (\frac{minimax}{i3_{max}}) \sigma_{\overline{i3}}$$

The planner's normalized predicted intensities are:

$$i1_{normplan} = (\frac{minimax}{i1_{max}}) \overline{i1_{plan}}$$

$$i2_{normplan} = (\frac{minimax}{i2_{max}}) \overline{i2_{plan}}$$

$$i3_{normplan} = (\frac{minimax}{i3_{max}}) \overline{i3_{plan}}$$

So, the intensity distributions used by the planner are:

$$N(i1_{normplan}, \sigma_{i1_{plan}})$$

$$N(i2_{normplan}, \sigma_{i2_{plan}})$$

$$N(i3_{normplan}, \sigma_{i3_{plan}})$$

The planner can then use this distribution in its simulation. The planner uses these distributions to produce 1000 noisy sets of intensity values. Then, it uses these 1000 noisy sets of intensity values to calculate 1000 values of N_{noise}. Using the 1000 values of N_{noise}, the planner can calculate the mean surface orientation error, $\overline{\theta_{err}}$, and its standard deviation, $\sigma(\theta_{err})$.

As can be seen, the error prediction part of the planner is becoming a simulator. Generally, as the planner incorporates more of the factors that influence the measurements, the planner's predictions will become more accurate. It is an open question how accurate a planner should be. This will depend upon the application and the significance of the factors affecting the measurements. The amount of error caused by violating assumptions depends on how severely they are violated, and on how sensitive the measurements are to these violated assumptions.

5.2. Light Intensity Variance

The variance of the camera and digitizer, σ^2_i, is a function of light intensity as shown by Healey [26, 27]. We have taken an empirical approach to determining the value of the function. In general, our measurements are in good agreement with Healey's. However, we do observe deviation from Healey's data for very small values of I, and for values of I which are greater than 175. Healey's data does not go beyond this point.

5.3. Camera Viewpoint Selection

We need to position a camera so that it views all of the faces illuminated by each illumination aspect. Camera visibility is determined in an identical manner to determining light source visibility. The only difference is that the set of potential camera viewpoints may be different from the set of light source directions. Given a set of potential camera viewpoints,

virtual cameras are positioned at each viewpoint. Object face visibility is determined. If the set of illuminated faces is a subset of the set of visible faces, then the viewpoint is added to a list of candidate viewpoints. The viewpoint, from the list of candidates, that views the set of faces to be inspected with the least foreshortening is selected as the camera viewpoint. The idea of this metric is to maximize the area of each inspected face in our image. This should make defects maximally visible. Formally, over the set of possible viewpoints, we try to minimize:

$$max(\frac{1}{\cos\theta_1}, \frac{1}{\cos\theta_2}, ..., \frac{1}{\cos\theta_n})$$

where θ_i measures the angle between the viewpoint and faces 1 ... n.

5.4. Error Sources

There are two basic types of errors in photometric measurements of lambertian objects: random errors and fixed errors. Random errors are due to σ_i. These are the errors that we try to predict with our planner. Fixed errors include: errors in light source direction, errors in light source radiance, errors in the photometric function (deviation from pure lambertian). Fixed errors can be accounted for by a careful calibration procedure. [28]

There is also a third type of error. This is the potential error in object pose. Our planner assumes that the object being viewed is in a certain orientation. If the object is in a different orientation, the planner's predicted surface orientation error will be erroneous. A change in orientation will cause a change in the incident angles between the light sources and object faces. This will cause a change in expected mean light source intensities, σ_i , and thus the mean surface orientation error, $\overline{\theta}_{err}$, and its standard deviation, $\sigma(\theta_{err})$.

The amount of change in mean light source intensities depends on the lambertian photometric function (the cosine function). If the incident angle is near normal, a small change in rotation will not cause a large change in the expected mean light source intensity. If the incident angle is small, a small change in rotation could cause a large change in the expected mean light source intensity.

5.5. Texture and σ_i

Our use of σ_i , assumes that the planar surfaces of our objects have no texture. Any spatial variation of the surface due to texture will cause an increase in the measured value of σ_i , beyond what is caused by the CCD alone. It would be possible to form an aggregate σ^2_i .

$$\sigma^2_{i_{aggregate}} = \sigma^2_{i_{CCD}} + \sigma^2_{i_{texture}}$$

The aggregate would be composed of two terms. One term would be the intensity variation caused by the camera and digitizer. The second term would be the intensity variation due to texture. (These terms are independent.) In this way, surface defects larger than the surface deviation caused by texture could be reliably detected.

6. Implementation

The planner was implemented in Lucid™ Common Lisp. Experiments were carried out using a Puma 560 robot and an array of light sources placed at the vertices of a 1-frequency icosahedron. (For structural reasons, 61 faces actually exist and 36 of the vertices have light sources). A Sony XC-57 camera, with a Nikon AF Nikkor Micro 105mm F2.8 lens, was mounted on the Puma's end effector. The Puma Robot, with the camera mounted on it could only reach 22 of the 61 icosahedron faces (Camera viewpoints were located at the faces of the 1-frequency icosahedron.) because of robot workspace constraints. Images were digitized using an Androx ICS-400 digitizer.
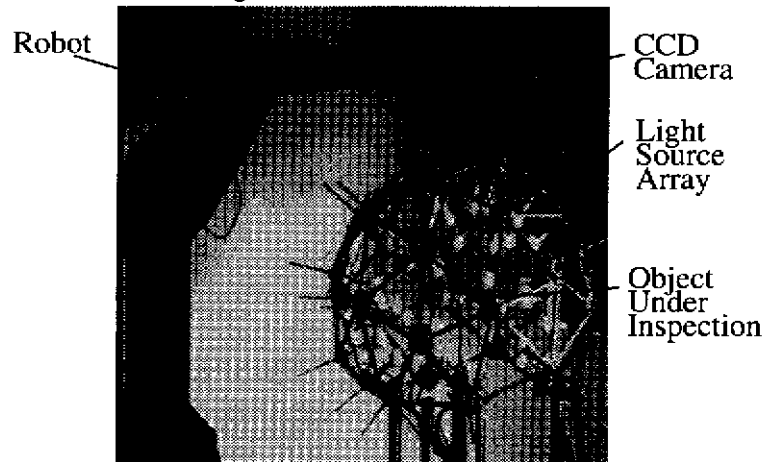

Fig. 10. Experimental Setup

6.1. Measurement of Light Intensity Variance

In order to measure σ_i, illumination, viewing geometry, camera temperature, and CCD voltages must be controlled. Any variance in any of these parameters will cause added variation to σ_i. We used a light source controlled with a linear DC power supply and used a linear DC power supply to power the Sony XC-57 camera (We found that controlling the camera with a standard switching camera power supply increased σ_i.) Camera temperature was held constant by allowing the camera to reach steady state temperature with respect to its environment. Viewing geometry was maintained by rigidly supporting the camera and target. Our target was of uniform matte reflectance.

We selected four pixels on the CCD, and took 1000 images of our fixed scene. Illumination was changed by varying the incident angle between the light source and target. We took measurements between mean intensities of 8 (dark current value) and 227. Normality was checked by plotting a histogram of intensity for each pixel, and by using the Kolmogrov-Smirnov test [29]. The average significance level of the Kolmogrov-Smirnov test was 0.342.
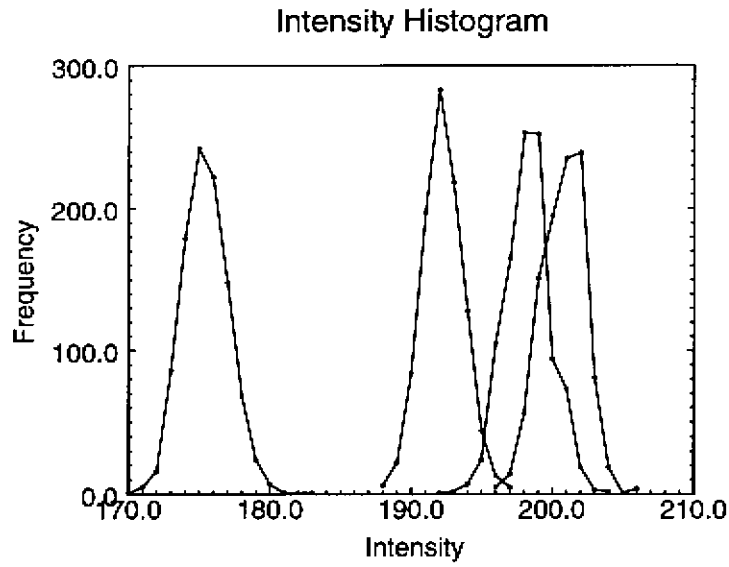
A representative histogram is shown below:

Fig. 11. Intensity Histogram

For each pixel we calculated the mean, variance, and standard deviation of the intensity measurements. We fitted a second degree polynomial to the raw data, and used this polynomial in our planner. The plot of the standard deviation of intensity is shown below. The fitted polynomial is shown as the solid line.
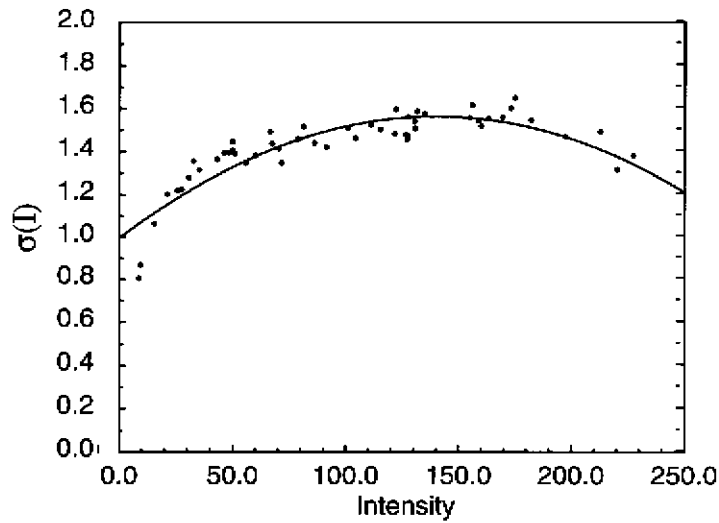

Fig. 12. Plot of σ_i

7. Experiments

Experiments were conducted to validate the lambertian illumination planner. We wanted to verify that the illumination plans generated by the planner were valid and accurate. In order to do this, we tested the results of the planner for a set of illumination plans. One set of plans tested was near the most accurate illumination plan generated by our illumination planner. The other set of plans tested was near the least accurate illumination plan generated by our illumination planner. The results of the two plans were compared with each other and with the planner's predictions. We also verified that the light source viewpoint visibility and camera viewpoint visibility predicted by the planner were correct.

7.1. Chalk Cube

We milled a cube out of 'railroad chalk'. The cube was mounted under our array of light sources. The cube was oriented so that the X, Y, and Z axes intersect at the center of the cube. The X axis intersects one edge of the cube. The Y axis intersects another edge of the cube. The Z axis intersects the center of the top face of the cube:

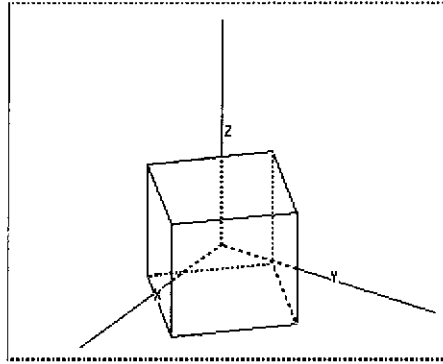
Fig. 13. Geometric Model of Chalk Cube

The nominal orientation of the cube's faces is shown in the table below.

Face	Nx	Ny	Nz
1	-0.707	0.707	0.0
2	-0.707	-0.707	0.0
3	0.707	-0.707	0.0
4	0.707	0.707	0.0
5	0.0	0.0	1.0
6	0.0	0.0	-1.0

The possible light source directions:

Light Source	Nx	Ny	Nz
1	-0.3090	-0.5000	0.8090
2	0.3090	-0.5000	0.8090
3	0.0000	0.0000	1.0000
4	0.0000	-0.8507	0.5257
5	0.5257	0.0000	0.8507
6	-0.5257	0.0000	0.8507
7	-0.8090	-0.3090	0.5000
8	-0.5000	-0.8090	0.3090
9	0.5000	-0.8090	0.3090
10	0.8090	-0.3090	0.5000
11	0.3090	0.5000	0.8090
12	-0.3090	0.5000	0.8090
13	0.0000	-1.0000	0.0000
14	0.8090	0.3090	0.5000
15	-0.8090	0.3090	0.5000
16	-0.8507	-0.5257	0.0000
17	0.8507	-0.5257	0.0000
18	0.0000	0.8507	0.5257
19	-1.0000	0.0000	0.0000
20	-0.5000	-0.8090	-0.3090
21	0.5000	-0.8090	-0.3090
22	1.0000	0.0000	0.0000
23	0.5000	0.8090	0.3090
24	-0.5000	0.8090	0.3090
25	0.0000	-0.8507	-0.5257
26	0.8507	0.5257	0.0000
27	-0.8507	0.5257	0.0000
28	-0.8090	-0.3090	-0.5000
29	0.8090	-0.3090	-0.5000
30	0.0000	1.0000	0.0000
31	-0.3090	-0.5000	-0.8090
32	0.3090	-0.5000	-0.8090
33	0.8090	0.3090	-0.5000
34	0.5000	0.8090	-0.3090
35	-0.5000	0.8090	-0.3090
36	-0.8090	0.3090	-0.5000

The possible viewing directions:

View	Nx	Ny	Nz
5	-0.3090	-0.1667	0.9363
15	-0.1667	0.9363	0.3090
16	0.1667	0.9363	0.3090
17	0.0000	0.3568	0.9342
18	0.0000	0.6667	0.7454
19	-0.3090	0.1667	0.9363
21	-0.5774	0.5774	0.5774
22	-0.2697	0.7697	0.5787
23	-0.5787	0.2697	0.7697
24	-0.7697	0.5787	0.2697
25	0.5774	0.5774	0.5774
26	0.2697	0.7697	0.5787
27	0.5787	0.2697	0.7697
28	0.7697	0.5787	0.2697
30	-0.7454	0.0000	0.6667
34	0.7454	0.0000	0.6667
37	-0.3568	0.9341	0.0000
41	0.3568	0.9341	0.0000

The best case inspection plans, assuming ideal, equal radiance, light sources, found by our planner are below. Cover 1 and cover 2 are not implementable because the indicated aspects are not viewable.

Cover	Aspect	Faces Covered	View	Light Sources
1	1	2, 3	none	13, 31, 32
1	2	1, 4, 5	18	11, 23, 24
2	1	1, 4	17	30, 34, 35
2	2	2, 3, 5	none	1, 8, 9
3	1	1, 2	19	19, 28, 36
3	2	3, 4, 5	30	5, 10, 14
4	1	3, 4	34	17, 26, 29
4	2	1, 2, 5	30	6, 7, 15

Inspectable faces are defined as the object faces that can be viewed from the set of possible viewpoints (face 6 is not inspectable). For each light source, we determine which of the inspectable faces are illuminated. (If a non-inspectable face is illuminated, we treat it as a "don't care".) The different combinations of illuminated faces form illumination aspects. These aspects are formed into exact illumination covers. We used our "largest aspect" heu-

ristic, with a search depth of one, to form the exact covers. It is possible that an illumination aspect contains a combination of object faces that is not viewable. This leads to the cases like covers 1 and 2, which both contain illumination aspects that are not viewable. (If a potential viewpoint views a non-inspectable face, we treat it a "don't care".)

We implemented a near best case inspection plan for faces 3 and 4 using light sources 22, 29, 33 and viewpoint 34. The near worst case plan for faces 3 and 4 was light sources 22, 26, 29 and viewpoint 34. The planner determined that faces 1, 2, and 5 could only be illuminated by light sources 6, 7, 15. The camera viewpoint for faces 1, 2, and 5 was viewpoint 30.

We implemented these inspection plans on our experimental setup. The results, using measured radiance light sources:

Face	Light Sources	Pixel Number	Planner Predictions		Measurements	
			$\overline{\theta}_{err}$ - Degrees	$\sigma(\theta_{err})$ - Degrees	$\overline{\theta}_{err}$ - Degrees	$\sigma(\theta_{err})$ - Degrees
3	22,29,33	1	1.21	0.68	1.11	0.62
3	22,29,33	2	1.21	0.68	1.13	0.61
3	22,26,29	1	1.55	1.05	1.70	1.14
3	22,26,29	2	1.55	1.05	1.75	1.18
4	22,29,33	3	1.21	0.67	1.24	0.64
4	22,29,33	4	1.21	0.67	1.32	0.69
4	22,26,29	3	1.64	1.12	1.79	1.14
4	22,26,29	4	1.64	1.12	1.85	1.13
1	6,7,15	5	1.10	0.67	1.16	0.68
1	6,7,15	6	1.10	0.67	1.12	0.66
2	6,7,15	7	1.06	0.62	1.31	0.75
2	6,7,15	8	1.06	0.62	1.28	0.73
5	6,7,15	9	1.18	0.61	1.06	0.57
5	6,7,15	10	1.18	0.61	1.20	0.71
5	6,7,15	11	1.18	0.61	1.08	0.57
5	6,7,15	12	1.18	0.61	1.04	0.58

Measurements were made by taking 1000 images with each light source. A small number of pixels (4) were recorded from each image. This produced a data stream for that pixel. By combining data streams for 3 light sources, we calculated a mean surface orientation, a mean surface orientation error, and the standard deviation of surface orientation error.

In general, the measured mean orientation error and standard deviation orientation error are within 20% of the planner. Many are within 10%.

The difference between the near best case and near worst case plans should be noted. For Face 3, the near best case illumination plan results in a 2.04° ($\overline{\theta}_{err} + 3\sigma(\theta_{err})$) predicted error.

In contrast, the near worst case illumination plan results in 4.7° predicted error. For Face 4, the near best case illumination plan results in a 3.22° predicted error. In contrast, the near worst case illumination plan results in 5.0° predicted error.

Images of Faces 3 and 4 using light sources 22, 29, and 33. (Face 5 is not illuminated.)
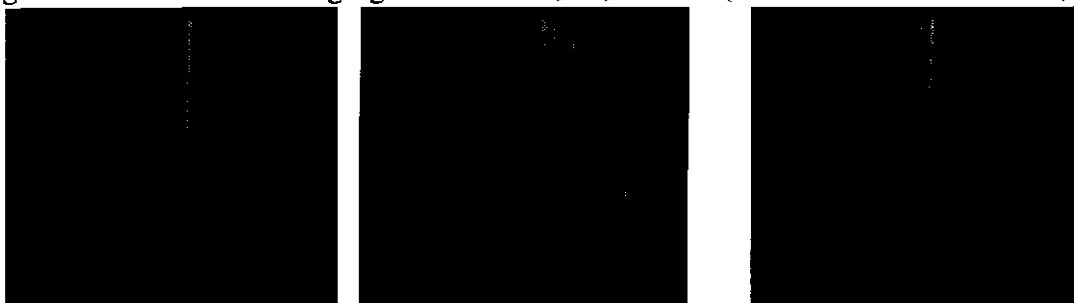

Fig. 14. Intensity Images: Faces 3 and 4 of cube

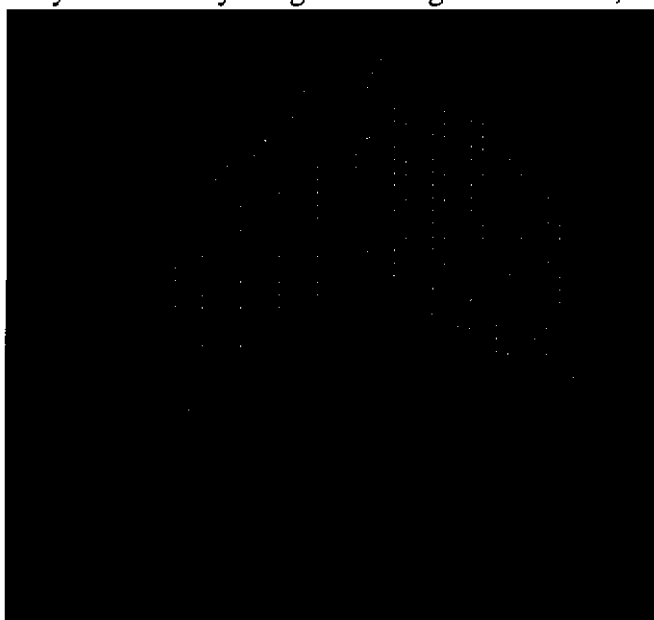Needle map produced by the intensity images from light sources 22, 29, and 33.

Fig. 15. Needle Map: Faces 3 and 4 of cube

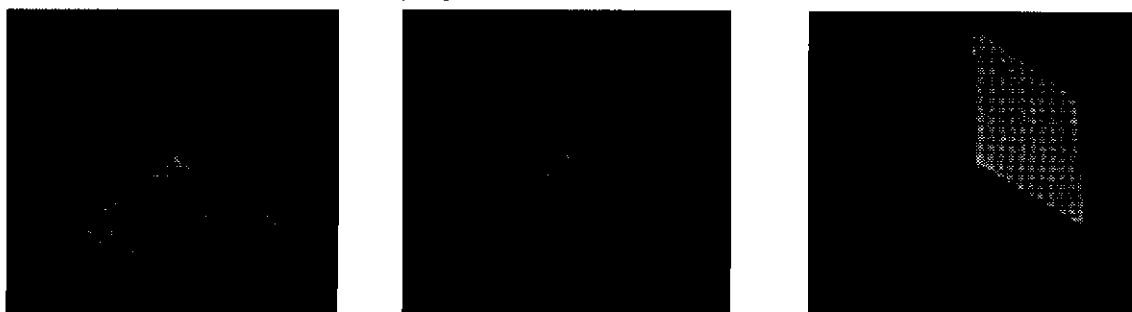Images of Faces 1, 2, and 5 using light sources 6, 7 and 15.

Fig. 16. Intensity Images: Faces 6, 7, and 15 of cube

Needle map produced by the intensity images from light sources 6, 7, and 15.
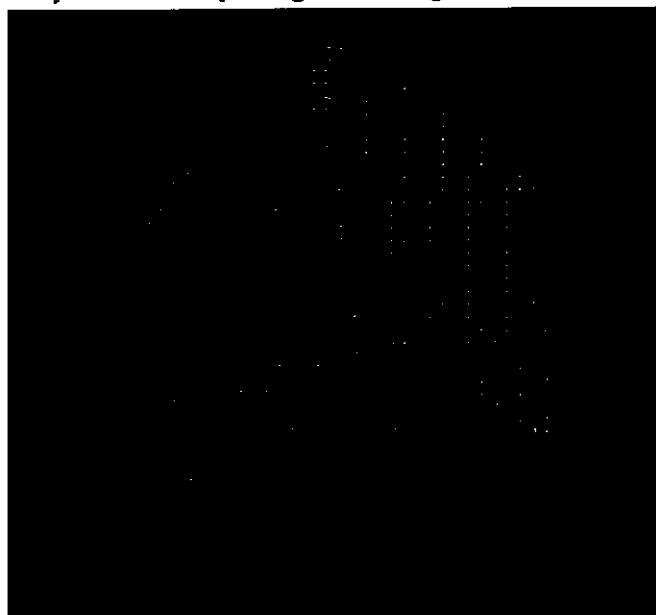

Fig. 17. Needle Map Faces: 6, 7, and 15 of cube

Illumination aspects from the 1, 1, 1 viewing direction (Numbers are light source numbers. Light sources that are shaded the same belong to the same illumination aspect.):
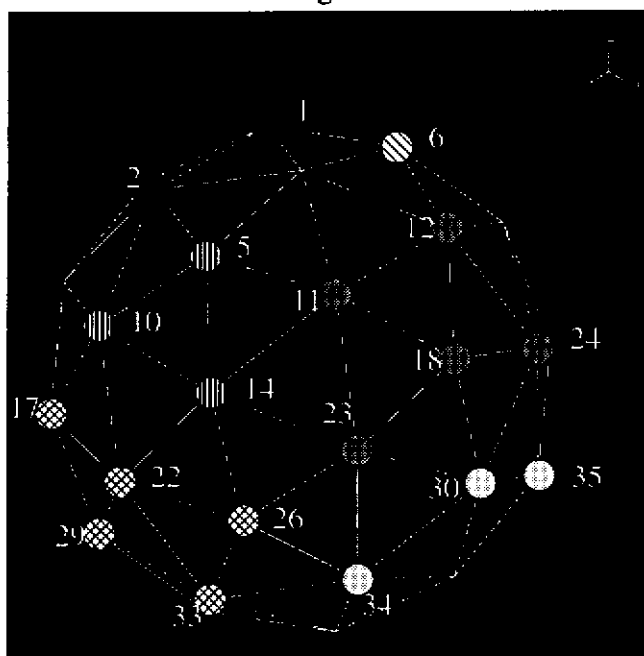

Fig. 18. Illumination aspects from the 1, 1, 1 viewing direction

Illumination aspects from the -1, 1, 1 viewing direction (Numbers are light source numbers.

Light sources that are shaded the same belong to the same illumination aspect.):
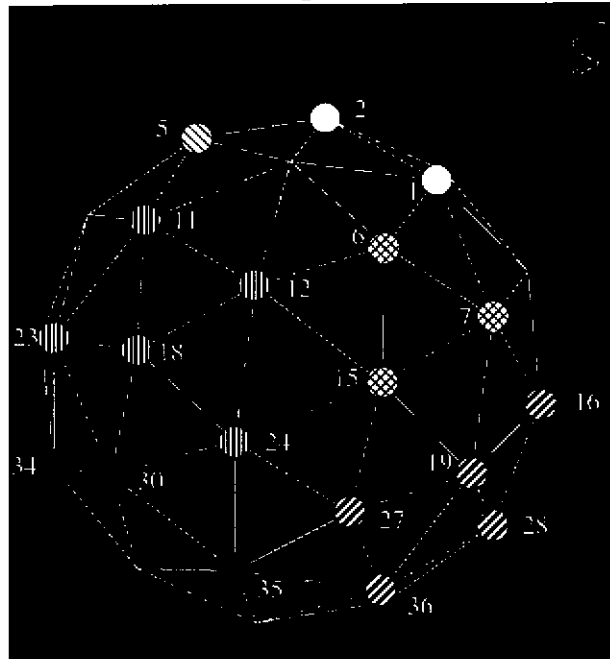

Fig. 19. Illumination aspects from the -1, 1, 1 viewing direction

Illumination aspects from the 0, 0, 1 viewing direction (Numbers are light source numbers. Light sources that are shaded the same belong to the same illumination aspect.):
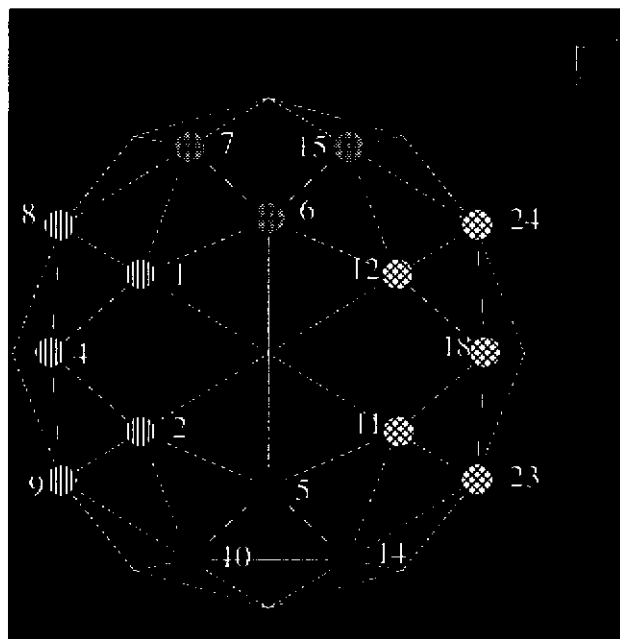

Fig. 20. Illumination aspects from the 0, 0, 1 viewing direction

8. Conclusions

We have developed an illumination planner for convex lambertian objects. The planner determines how to position light sources around an object so that we illuminate a specified set of faces in an efficient manner. The planner uses an efficient heuristic, that we have developed, for finding exact illumination covers in 3D. Using a noise model of the CCD, a CAD model of the object we are inspecting, and the lambertian reflectance model, the planner performs a statistical simulation to determine how much uncertainty we can expect in our shape measurement for a given light source configuration. The planner generates an illumination plan that illuminates the specified set of object faces while having a minimum amount of uncertainty.

We have verified that the illumination plans generated by the planner are valid and accurate. In most cases the uncertainty predictions made by the planner were accurate to within 10%.

This work has potential applications to inspection problems in industrial environments. Although the photometric stereo method requires careful calibration. We have shown that it is possible to accurately predict measured shape uncertainty. This ability can be used to reliably detect defects.

9. References

[1]   C. Cowan, "Automatic Sensor Placement from Vision Task Requirements", *IEEE Trans. on Pattern Analysis and Machine Intelligence*, Vol. 10, No. 3, pp. 407-416, May, 1988.

[2]   C. Cowan, *"Determining the Camera and Light Source Location for a Visual Task"*, 1989 IEEE International Conference on Robotics and Automation, pp. 509-514.

[3]   C. Cowan, *"Automatic Camera and Light Source Placement Using CAD Models"*, IEEE Workshop on Directions in Automated CAD-Based Vision", June 1991, Maui, Hawaii, pp. 22-31.

[4]   R. Tsai and K. Tarabanis, *"Occlusion Free Sensor Placement"*, Yorktown Heights, NY: T.J. Watson Research Center, RC 14593, March 1989.

[5]   R. Tsai and K. Tarabanis, *"Satisfying the Resolution Constraint in the "MVP" Machine Vision Planning System"*, Proceedings DARPA Image Understanding Workshop, Pittsburgh, P.A., September 11-13, 1990.

[6]   R. Tsai, K. Tarabanis, and P. Allen, *"Automated Sensor planning for robotic vision tasks"*, 1991 IEEE International Conference on Robotics and Automation, pp. 76-82.

[7]   G. Hager, *"Active Reduction of Uncertainty in Multi-sensor systems"*, University of Pennsylvania, Philadelphia, Pennsylvania, Ph.D. Thesis, MS-CIS-88-47.

[8]   G. Hager, *"Computational Methods For Task-Directed Sensor Data Fusion and Sensor Planning"*, University of Pennsylvania, Philadelphia, Pennsylvania, MS-CIS-90-38.

[9]   S. Sakane, M. Ishii, and M. Kakikura, *"Occlusion avoidance of visual sensors based on a hand-eye action simulator system: HEAVEN"* , Advanced Robotics, Vol. 2, No 2, pp. 149-165.

[10]   S. Sakane, T. Sato, and M. Kakikura, *"Automatic Planning of Light Source Placement for an Active Photometric Stereo System"*, 1990 IEEE International Workshop on Intelligent Robots and Systems, pp. 559-556.

[11]   S. Sakane and T. Sato, *"Automatic Planning of Light Source Placement and Camera Placement for an Active Photometric Stereo System"*, 1991 IEEE International Conference on Robotics and Automation, pp. 1080 - 1087.

[12]   R. Niepold, S. Sakane, T. Sato, and Y. Shiraii, *"Vision Sensor Set-up for a Hand-Eye System Using Environmental Model"*, SICE 87, pp. 1037-1040.

[13]   S. Yi, R. Haralick, and L. Shapiro, *"Automatic Sensor and Light Source Placement for Machine Vision"*, University of Washington, Seattle, Washington, CS-89-11-03.

[14]   S. K. Nayar, K. Ikeuchi, T. Kanade, "Surface Reflections: Physical and Geometrical Perspectives", *IEEE Trans. of Pattern Analysis and Machine Intelligence*, Vol. 13, No. 7, pp. 611-634, July, 1991.

[15]   J. J. Koenderink and A. J. van Doorn, "The Internal Representation of Solid Shape with Respect to Vision", *Biol. Cybernetics*, Vol. 32, pp. 211-216, 1979.

[16]   K. Ikeuchi and T. Kanade, "Automatic Generation of Object Recognition Programs", *Proceedings of the IEEE*, Vol. 76, No. 8, pp. 1016-1035, August, 1988.

[17]   M. Hebert and T. Kanade, "The 3D-Profile Method of Object Recognition",1985 IEEE Conference on Computer Vision and Pattern Recognition, pp. 458-463.

[18]   Z. Gigus, J. Canny, and R. Seidel, "Efficiently Computing and Representing Aspect Graphs of Polyhedral Objects", *IEEE Trans. of Pattern Analysis and Machine Intelligence*, Vol. 13, No. 6, pp. 542-551, June, 1991.

[19]   D. Kriegman and J. Ponce, "Computing Exact Aspect Graphs of Curved Objects: Solids of Revolution", *International Journal of Computer Vision*, Vol. 5, No. 2, pp. 119-135.

[20]   J. Stewman and K. Bowyer, "Aspect Graphs for Convex Planar-Face Objects", IEEE Workshop on Computer Vision", Miami, Florida, 1987.

[21]   H. Plantinga and C. Dyer, "Visibility, Occlusion, and the Aspect Graph", *International Journal of Computer Vision*, Vol. 5, No. 2, pp. 137-160.

[22]   K. Bowyer and C. Dyer, "Aspect Graphs: An Introduction and Survey of Recent Results", *International Journal of Imaging Systems and Technology*, Vol 2, No. 4, 1990, pp. 315-328.

[23]   B. G. Batchelor, D. A. Hill, and D. C. Hodgson, *"Automated Visual Inspection"*, 1985, IFS (Publications) Ltd. , Bedford, U.K. .

[24]   B. G. Batchelor, "A Prolog Lighting Advisor", *Intelligent Robots and Computer Vision VIII: Systems and Applications*, Bruce G. Batchelor, Editor, Proc. SPIE 1193.

[25]   B. G. Batchelor and F. M. Waltz, Editors, "Machine Vision Systems Integration", 1990, SPIE Optical Engineering Press, Bellingham, Washington.

[26]   G. Healey and R. Kondepudy, *"CCD Camera Calibration and Noise Estimation"*, 1992 IEEE International Conference on Computer Vision and Pattern Recognition, pp. 90 - 95.

[27]   G. Healey and R. Kondepudy, "Modeling and Calibrating CCD Cameras for Illumination Insensitive Machine Vision", *Optics, Illumination, and Image Sensing for Machine Vision VI*, Proc. SPIE 1614.

[28]   R. Ray, J. Birk, and R. Kelley, "Error Analysis of Surface Normals Determined by Radiometry", *IEEE Trans. of Pattern Analysis and Machine Intelligence*, Vol. 5, No. 6, pp. 631-645, November, 1983.

[29]   W. Press, B. Flannery, S. Teukolsky, W. Vetterling, *"Numerical Recipes in C"*, Cambridge University Press, New York, 1990, pp. 490 - 494.

[30]   B. Kumar, J.C. Robert, R. Hoffman, K. Ikeuchi. and T. Kanade, *"Vantage, A Frame Based Geometric Modeling System Programmer/Users Manual V2.0"*, CMU-RI-TR-91-31, December 1991.