I. Abstract

This paper relates an attempt to construct reconstructions of arbitrary three-dimensional scenes with a system that uses a Bayesian evidence grid framework to integrate sparse samples of the plenoptic function into a voxel model of the environment of the system. This is attempted to be accomplished by analyzing the probability that a sample of the plenoptic function will contain a consistent color along all rays that pass through a given voxel. It is assumed that voxels along the surface of an object will always appear the color of the object at that point on the surface, discounting occlusion and specularities. It is further assumed that transparent voxels will vary greatly in appearance depending on direction they are viewed from. Given these assumptions, results that, while not strictly successful, indicate this approach could have some potentially useful applications have been achieved by performing such an analysis on synthetic scenes, and some potential improvements are discussed.

II. Introduction

Three-dimensional scene reconstruction can be stated as the problem of taking images of an environment or object as input and outputting sufficient information to render an image of the scene from an arbitrary viewpoint accurately. Such information often, but not always, takes the form of a model of the object or environment that explicitly defines the shape and appearance of the object being reconstructed. It is the goal of the system described here to integrate a system that performs such a reconstruction into a mobile agent that could act autonomously in an unknown environment to reconstruct the environment as it is explored. There are a few companion problems to this task. Since most robots are not particularly good at estimating their position as they move based on hardware alone, most autonomous agents performing mapping tasks must also perform some sort of continuous localization as the map is constructed [15]. Another component is motion and sensor planning to position the agent and it’s sensors in
locations that will allow data that can be used to improve the reconstruction as efficiently as possible [1]. While the system described here does not attempt to perform either of these tasks, it was designed to allow methods very similar to existing ones that have proven effective [1, 15] to be designed and integrated as easily as possible.

Most previous attempts to address the problem of 3D scene reconstruction rely on feature detection and association to create three-dimensional geometric models of objects. These work relatively well, but require feature rich objects and still often fail to capture the true shape of objects. Recently, new techniques have been introduced that rely upon dense plenoptic sampling and storing enough information to reconstruct the scene from arbitrary viewpoints, but none of the structural information of the scene is preserved. Voxel-based models, which make the most accurate and complete models of real objects of any techniques currently available, are relatively easy to construct for small objects, but normally require highly controlled situations, and few attempts have been made to construct voxel models of entire environments. The system described here attempts to do just that, while relaxing the precision needed of other voxel models. This system also attempts to evade the limitations of polygonal methods and the sampling density of methods that do not store structural information.

Other voxel-based techniques have operated by projecting each voxel back into each input image, and checking for correlations between the colors at those places in the images. One disadvantage to this is that all of the required input images have to be available when evaluating each voxel. This system achieves the same effect by projecting each pixel in each image to the voxel-space and finding each voxel that it intersects. It then employs Bayes’ theorem by considering each pixel as evidence toward each voxel being the color of that pixel. Since Bayes’ theorem is commutative, the images can be analyzed in any order, and more images can always be added to a given reconstruction to increase the accuracy. This means that a very rough and incomplete model can be reconstructed from a very sparse number of sample images, and the accuracy of the image can be improved as the sampling density increases. Additionally, since each image counts only as probabilistic evidence, errors in a few images or small random errors throughout the images will have a diminished effect as the number of images used increases. Also unlike other voxel-based schemes, the opacity of a given voxel is not directly calculated from correlations of colors in the input images. Since Bayes’ theorem provides a simple and elegant way to modify a probability distribution, each voxel consists of a probability distribution
of observed colors. Opacity is therefore calculated by analyzing these probability distributions, and is not a property that is attempted to be determined directly from the images themselves.

III. Previous Work

The light field (in Gershun’s terminology)[7] or plenoptic function (in Adelson and Bergen’s)[2] is a descriptive mechanism for characterizing any visual information received by any agent. The full plenoptic function can be expressed as a six-dimensional function, the value of which is the light value or luminance at a point in space and time along a vector. If the exact value of this function could be stored for an arbitrarily large area at an arbitrarily small resolution or sampling rate, the scene could be perfectly recreated from any viewpoint, since all possible visual information would be stored. Obviously, there is no way to either sample or store the absurd volume of information this would require. Since many environments and objects exhibit a large degree of visual stability over periods of time of arbitrarily length, the temporal dimension is typically dropped, reducing the function to a mere five dimensions. This is still a very large amount of data to store, and sampling at a sufficiently fine rate is still impractical. Furthermore, since air generally acts as a perfect medium of transmission, there is a lot of data redundancy, as light value as uniform along a vector until that vector intersects an object in the scene. Any system that attempts to extract information from input images can be technically considered to be performing plenoptic sampling. However, the term is generally applied to systems that extract information for the purpose of recreating the scene from various points of view. Often, but not exclusively, the term is applied to systems that attempt to do this without storing structural information at all. Various attempts have been made to sample the plenoptic function in such a way that it can be efficiently accomplish this task, store sufficient information such that arbitrary portions of the plenoptic function can be recreated later as images of the same scene. More broadly, one of the greatest challenges of computer vision is to extract enough three-dimensional information from images of objects or environments that the same objects or environments can be later rendered from arbitrary points of view.

The most conventional methods of reconstructing real objects or scenes involve constructing polygonal models that approximate the scene. When attempting to render images that mimic the real scene, one of the simplest methods is, after a polygon model has been created, to segment the input images based on the reconstructed models. These image segments are then transformed into appropriate images that can be texture mapped onto the appropriate polygons.
While feature detection can be used to calculate the position of prominent features of objects, features that can be easily identified from one image to another rarely give much information of the shape of the objects. This is especially a problem if there are large spaces between surfaces. The shape of the polygonal objects is sometimes calculated by identifying occlusion boundaries in the input images and using those edges as surface contours that are in profile of the camera. The limitations of these methods are obvious, as the world is generally non-polygonal, and accurate polygonal models are difficult to acquire when it is, since tracking features from one image to another and determining the relationship of the between images is a daunting task. The most successful of these methods perform extensive feature detection and matching in addition to contour detection so that the relative position of each contour can be more easily determined and so that textures can be more correctly mapped onto the surfaces created. Often, calibrated stereoscopic cameras are used such that depth information associated with each of the features in the input images can be calculated. Other reasonably successful methods rely on planning the motion of the camera at each step and taking many densely spaced, well-planned samples, such that a near-smooth surface can be reconstructed. While these methods are moderately successful, and create models that are easily rendered in many common graphics environments, they tend to be most useful for feature-rich, yet geometrically simple surfaces.

Strictly plenoptic sampling methods have been developed that can model either the appearance of an object or of an environment by storing enough of the plenoptic function itself that missing pieces can be interpolated in. Similar methods use mosaics to recombine the appropriate parts of input images to ‘render’ the environment from new viewpoints. Unfortunately, these methods typically can be used to model either objects or environments, but not both. The amount of information contained in the plenoptic function is simply immense. However, to correctly capture enough of the complexity of a scene to convey depth information, it is necessary to densely sample a sufficiently large portion of it. Therefore, to even begin to store meaningful portions of it that are still useful, it has always been necessary to select a four-dimensional slice from the five-dimensional plenoptic function. This translates into defining a two-dimensional surface that takes the form of a convex hull that completely encases either the object to be modeled or the possible positions of the camera for reconstruction. Whichever side of the surface the camera resides on, it is necessary for the volume containing the camera to be free space. That is, it can contain no objects that could neither occlude the object or environment being modeled, nor obstruct the motion of the camera, nor in any way affect the appearance of the light after it has passed through the surface. As such, all true three-dimensional information
about the object or environment that has been modeled is lost. While accurate renderings can often be made from this information, it is lost for any other purpose that might need information of an object’s three-dimensional structure, such as collision detection and other sorts of information necessary to even vaguely accurately model the physics of objects in virtual environments. Additionally, all of these methods require a rigorously dense sampling density, and the amount of information that is necessary to store is often extremely large. Partial reconstructions could theoretically be constructed from an arbitrarily small number of samples, but would generally interpolate between the samples in a way that would convey a distorted and erroneous impression of spatial relations, which would most likely not be useful. Some methods rely on using more sophisticated interpolation functions to be able select a set of samples to store that is smaller than the set originally collected [8]. This reduces the data storage needed, which is still immense, but still requires a very large number of images to be collected and processed.

Other techniques rely on using stereo information directly instead of using to provide depth information for image features. These methods overall have the fewest limitations and require the fewest number of input images. However, they often rely upon specialized hardware, and stereo correspondence algorithms are only accurate in certain situations. However, given sufficiently accurate depth information, a number of modeling and reconstruction methods are possible. For instance, since surface normals can be estimated from the change in depth, as well as lighting cues then a large number of flat particles oriented along the surface can be used to render a convincingly smooth surface[5]. Another solution is to generate a mesh using surface points located with stereo information as vertices[6]. Alternately, if the stereo information is both accurate and dense enough, a voxel model can be generated directly from the surface points. One system that does just that that in a way that closely resembles the system described here was designed by Moravec [13]. Moravec uses stereo information in much the same way as he uses sonar information, which is described below, to accumulate evidence about the location of surfaces in an evidence grid. However, using stereo information allows for a much higher resolution than sonar information, and easily allows the evidence grids to be three-dimensional, such that a full three-dimensional scene is reconstructed. However, all of these approaches require highly calibrated stereo cameras, and rely on stereo-correspondence algorithms, which depend on feature-rich surfaces. One approach that overcomes most limitations on scene composition is to integrate images with a depth map from a laser range finder instead of relying on stereo information [14]. There are few foreseeable complications to this method, apart from the availability of the necessary equipment, of course. The complications it does have are shared
by all reconstruction methods, and are ones that are generally not attempted to be addressed here, although the technique described here is geared toward allowing these problems to be solved as easily as possible. Examples of these sorts of problems include highly specular or reflective objects, selecting an appropriate sampling density, correctly determining camera position, and path planning to ensure the most complete model of a given environment from the fewest number of images possible. Simplifying this last task, determining the most efficient set of camera positions to acquire images to provide information missing from a partial reconstruction, and making it possible to add such information into a partial reconstruction, was a primary motivation for developing the system described here.

As should already be obvious, there is a wide range of useful techniques for acquiring a voxel model of a given scene [4]. One such technique that has been highly successful for object reconstruction is voxel-based space carving, which has been primarily pioneered by Seitz[16]. This technique scans through the space to be modeled expanding away from a surface formed between the camera positions where the input images were taken, so that self-occlusion of an object is handled correctly. This technique requires that no points in the scene be contained within the hull of the camera centers. The algorithm proceeds in an order away from the camera volume, such that no voxel is processed before any voxel that could occlude it in any image. Each image also has an associated depth map of one bit per pixel, which records if a voxel that projects into the image at that pixel has been filled in or not. If one has, then all voxels considered afterwards that project into the same pixel should be assumed to be occluded. As each voxel is considered, it is projected into each image and color values are recorded from all images that do not yet have depth map entries. The color values of the pixels that voxel is projected into are compared. If there is a small enough variation between the color values, the voxel is filled in and colored appropriately, and the depth bit is set for the appropriate pixels in the input images. This technique has been further expanded in such a way as to exploit motion data from a sequence of images taken from multiple cameras simultaneously [19]. This allows the voxels (or ‘hexals’) to be carved out in time as well as space, such that geometric constraints on the motion of voxels can add information that is used to calculate a dynamic model of a scene over a period of time. However, the technique remains the same, despite the increased number of dimensions it is applied to.

This technique requires highly accurate internal and external camera calibration to get accurate models. Like the system described here, these methods only require a moderate number
of images as input, and accuracy increases with the sampling density. However, the set of images, as well as the size of the set, needs to be predetermined, and there is no way to create partial reconstructions from any smaller subset. Also, there is no intuitive way to further improve a model once it is generated using Seitz’s algorithm, for instance, so the model would have to be completely recalculated if more input images were taken. In contrast, any reconstruction by the system described here can be considered a partial reconstruction, as one can be made from any number of input images, and more input images can always be integrated in an attempt to increase the accuracy of the reconstruction. Also, the cameras still have to move within free space for Seitz’s algorithm to work correctly. For these reasons, this technique has been mainly used to create voxel models of objects, and while it would in principle work for environments, few attempts have been made to do so. Additionally, it must be considered that most sets of images can be generated from more than one coherent scene. Seitz’s technique will select the surfaces closest to the hull of camera positions that coherently accounts for the input images. The method described in this paper, in contrast, will reconstruct objects as occurring in all probable locations. This does not necessarily lead to scenes consistent with all input images, and does lead to a more imprecise reconstruction, but is expected that the system described here will be employed in situations where in input images are less than precise themselves. Also, the system described here theoretically applies just as well to voxels placed between camera positions.

The concept of a voxel in terms of the plenoptic function is that of a small region of space that is bounded by a four-dimensional slice of the plenoptic function. This slice is assumed to be completely transparent if the voxel is empty, and have a uniform light value throughout all four dimensions if full. There has been some work done using an expanded concept of a voxel which is uniform over u and v but variable based on the angle of incident light, these ‘procedural’ voxels are usually referred to as reflectance fields or lumispheres [20]. It is in principle possible to expand the technique described here to use such an expanded concept of a voxel, but the analysis of the resulting reflectance field to determine opacity of a give voxel would involve a non-intuitive procedure. Such an option will explored later in this paper, but for now it will be assumed that the plenoptic function along surface of each voxel will be sufficiently uniform as to be statistically significant.

It then is proposed that by performing a statistical analysis of samples of the plenoptic function, the most statistically likely voxel model for a given environment could be generated, and further updated with each sample taken. It is further hoped that such a statistical analysis
would permit the limitations of the other methods discussed to be dispensed with. If so, such a system could be integrated into an agent which could be left to freely roam within a complex environment collecting samples of the plenoptic function, and generating a voxel model of such an environment. The inspiration for such an approach to the voxel-based space-carving problem comes from the concept of an evidence grid as described by Moravec [12].

An evidence grid is a generalized way to solve any problem that involve breaking the environment of a agent into discrete spatial and/or temporal blocks and collecting sensory information about these regions. Such systems generally operate by selecting various attributes of each region of the environment and representing information in a way that preserves both the most likely value of each attribute and the certainty of that value. They were first developed by CMU for the task of robot navigation. In this scenario, the environment is broken into regions that the robot might potentially want to occupy, and thus the important attribute of each region that is investigated is if each region is already occupied or not. Since most robots to be faced with this type of navigation problem only have two free degrees of motion (ignoring orientation), such systems usually partition the environment into a two-dimensional grid of square regions. It is initially completely uncertain if each region is occupied or not. All sensory information is translated into a likelihood of a set of regions being occupied, and this information is integrated into the previous information for the appropriate regions. As more information is collected, the certainty of the information about each affected region and the environment as a whole goes up.

Moravec is one of the principal researchers who have concerned themselves with addressing the space-occupation problem as it relates to robot navigation. Moravec lays out a reasonably robust evidence grid framework based on Bayesian statistics that has become one of only a few standards for evidence grid frameworks. Let \( P(A \mid B) \) mean “the probability of proposition A being true given that B is true”. Bayes’ theorem states that

\[
P(B \mid A \cup C) = \frac{P(B \mid A) \times P(C \mid A \cup B)}{P(B \mid A) \times P(C \mid A \cup B) + P(B \mid A) \times P(C \mid A \cup B)}.
\]

In Moravec’s space-occupancy evidence grid, he lets \( C \) equal the \( j^{th} \) observation from sensor data, \( o_j \). When \( j = 0 \), \( A = K_a \), the \textit{a priori} knowledge of the environment, such that \( P(B \mid A) \) is the \textit{a priori} probability of a region being occupied, which Moravec sets to 0.5. When \( j > 0 \), \( A = \{K_a\} \bigcup_{j=0}^{a-1} \{o_j\} \), such that the probability \( P(B \mid A) \) is defined inductively by Bayes’
theorem. Thus, \( P(B \mid A \cup C) \) will be the probability that a region is in fact occupied given observations \( o_0 \cdots o_j \). The probability of observing \( o_j \) assuming that a given region is occupied, \( P(C \mid A \cup B) \), is defined by what Moracev refers to as a sensor model, a set of probabilities that are a function of both \( o_j \) and a given region’s position relative to the sensing device. His sensor model for sonar range finders assigned a high probability of receiving a reading of \( x \) if a region was in fact occupied at a distance \( x \) from the range finder. Conversely, the sensor model assigned a low probability of receiving a value of \( x \) if the regions of a distance less than \( x \) from the range finder were occupied.

IV. Mathematical Foundation

This system follows Moravec’s Bayesian framework as closely as possibly, but makes several necessary modifications appropriate for the task at hand. First of all and most obviously, the environment is partitioned into a three-dimensional array of cubic regions, much like Moravec’s own three-dimensional evidence grids [13]. Secondly, the probability of occupancy of a particular region is not directly calculated. The attribute being investigated can be thought of as a probability function in 6 dimensions, \( p(x, y, z, R, G, B) \), which is the probability of a ray of light with a color value of \([R, G, B]\) to arrive at the camera from the direction of \([x, y, z]\). Note that all of \(x, y, z, R, G\) and \(B\) are all continuous variables. Thus, a continuous probability distribution \( p'(x, y, z, R, G, B) \) is averaged over a region in \(x, y\) and \(z\) such that

\[
p(x, y, z, R, G, B) = \frac{\int_{x-e_x}^{x+e_x} \int_{y-e_y}^{y+e_y} \int_{z-e_z}^{z+e_z} \int_{R-e_R}^{R+e_R} \int_{G-e_G}^{G+e_G} \int_{B-e_B}^{B+e_B} p'(x, y, z, R, G, B) \cdot dx \cdot dy \cdot dz \cdot dR \cdot dG \cdot dB}{2e_x \cdot 2e_y \cdot 2e_z}
\]

The values of \( e_R, e_G \) and \( e_B \) define the range of color values to be examined and the values of \( e_x, e_y \) and \( e_z \) are all determined by the size of the cubic regions that the environment has been partitioned into. Thus, given a single region \([x, y, z]\), there still exists the continuous probability distribution \( p'(R, G, B) \). Since some light value will be recorded whenever a camera at an arbitrary location looks in an arbitrary direction, the total area under \( p'(R, G, B) \) for every region \([x, y, z]\) will always equal one. This constraint is also necessary to be able to apply
Bayes' theorem to any given \( p(R, G, B) \). Note that for probability distributions with an arbitrarily large number \( n \) of events, which are often used to simulate continuous probability distributions, Bayes’ theorem reduces to \( P(B_i \mid A \cup C_j) = P(B_i \mid A) \times P(C_j \mid A \cup B_i) \), as long as it is insured that \( \sum_{i=0}^{N} P(B_i \mid A \cup C_j) = 1 \) after the \( j^{th} \) application of the above equation integrates \( o_j \) into each \( P(B_i) \). To conserve storage space, this single three-dimensional probability distribution was broken into two. One distribution contained only one dimension, \( p'_i(I) \), where \( I = \frac{R + G + B}{3} \), or the intensity of the light arriving at the camera from a given direction. The other contained two dimensions, \( p'_e(r, g) \), where \( r = \frac{R}{3I} \) and \( g = \frac{G}{3I} \), or \([r, g]\) is the chromaticity of the light arriving at the camera in the same direction. Note that since 
\[ r + g = \frac{R + G}{R + G + B}, \]
it is necessarily the case that \( r + g \leq 1 \), thus the probability distribution is bounded within a triangular region. The value of \( e_r \) was set such that the range of light values that the camera is sensitive to was broken into \( N \) equal sections. The value of \( e_e \) and \( e_s \) were similarly set such that the possible chromaticities the camera could perceive were broken into 
\( M^2 \), equal sections.

Thus, the values of \( P(B \mid A \cup C) \) can be represented as the probability distributions \( p'_i(I) \) and \( p'_e(r, g) \). The values of \( P(B \mid A) \) when \( j = 0 \) was set to a even distribution, such that the light coming from the direction of every region in the scene could be thought of as having an equal probability of being every perceivable light value. The sensor model used calculated \( P(C \mid A \cup B) \) by first calculating the average light value arriving at the camera from the direction of a given region. Given that, continuous probability distributions were calculated that assigned an equal probability of having perceived that color to each possible color value for the given region, except for all color values within some small distance of the color value actually observed, which were all assigned some slightly larger value. The probability of having observed light of the intensity actually observed was always integrated into \( p'_i(I) \), whereas the probabilities for the observed chromaticity were only integrated when the intensity values were neither very large nor very small. Note that the probability of a given region being occupied is
not directly stored. Instead, the occupancy of a given region is determined by analyzing the probability distributions. Regions that have a probability distribution with a high variance are assumed to be transparent. Regions with probability distributions with a low variance are assumed to be opaque and to emit light of the color of the mean of the probability distribution. Thus, voxels along the surface of an object will have always appeared the color of the object at that point on the surface assuming occlusion and specularities did not overly affect the appearance. Voxels large enough to always contain multiple colors will have been considered to always appear the average color value. Transparent voxels will have varied greatly in appearance depending on direction they were viewed from. Voxels that lie on the surface of highly specular or translucent objects or on surfaces that were often occluded, or completely contain objects that appear different colors depending on which direction they are viewed from will unfortunately often be considered transparent.

V. Assumptions

Although this system was designed to free itself from the free-space assumptions of other space carving and plenoptic sampling methods, occlusion still represents the main obstacle to this approach. The free-space assumption takes the form of specifying that along a given ray, all values of the plenoptic function with a position along that ray in the direction of that ray will have the same value. This reflects the fact that as light travels in straight lines and it is assumed that there is nothing which changes the light value as it passes through the medium. For the system described here, although it need not be assumed that the plenoptic function exhibits coherence along rays, it must be assumed that it does so a sufficiently large interval of the function as to do so on a statistically significant basis. Any system designed along the principles outlined here must either assume that occlusion is a statistically insignificant phenomenon, or deal with it explicitly in some way. It is unclear what the boundaries of statistical significance are in this case, but it seems to be an unwarranted assumption in most environments. This is especially true since concave objects and any objects falling within the volume bound by the path of the camera will be self-occluding. It was attempted to be empirically determined by test cases run on simple synthetic scenes what conditions contained only statistically insignificantly occlusion, and what situations this technique would therefore be useful in. The results of this will be discussed later.

There are additional assumptions about the slice of the plenoptic function that surrounds each voxel. Ideally, each voxel could be defined to be small enough that it would be reasonable
to assume a uniform light value over all four dimensions. However, to achieve more generality, it will be assumed that a given voxel, if filled, will be surrounded by a four-dimensional slice of the plenoptic function for which the sum over u and v is uniform over all viewing angles. That is, the average light value leaving the voxel will be the same when perceived from any direction. Conversely, a given voxel, if empty, will presumably be surrounded by the same slice of the plenoptic function, and it will be assumed to be distinctly non-uniform, since it should vary greatly depending on viewing angle. When such non-uniformity is observed, it will be inferred that the slice of the plenoptic function bounding that particular voxel, and therefore the voxel itself, is in fact transparent. This is in general a valid assumption, but breaks down for empty voxels that are near large uniformly colored objects. Since it is unlikely that these areas will be viewed from many directions that do not make them appear the color of the large object, they will most likely be considered opaque, thus causing such objects to erroneously occupy nearby empty space. It is also sometimes the case that certain voxels will simply line up with similarly colored objects in a statistically significant portion of the input images. In both cases, falsely opaque voxels, referred to hereafter as 'artifacts', will appear in the final reconstruction. The requirements for uniformity can be set more strictly to avoid the occurrence of such artifacts. Then, however, quite often any voxel which is in fact opaque, but which is ever occluded or exhibits specularities, and therefore its projection back into some number of the input images will appear a color distinct from the color of the voxel, and will therefore be considered transparent. This is demonstrated later.

This system was designed such that it could be easily implemented so that it could be used on a free-roaming mobile agent. It would be possible to use Tsai’s method for internal camera parameter calibration [18] to reduce ray position calculation errors, however, it might not be a significant reduction. Since there is no camera distortion when using synthetic images, parameter calibration was never added to this system. Due to imprecision of position estimation on most robotic agents, the external camera parameters would be generally only rough estimates. Thus voxels were made big enough that errors from rough estimates of both internal and external calibration parameters would have little effect. However, the larger the voxel size used, the more imprecise the resulting reconstruction will be. Additionally, larger voxels will more invalidate the assumption that the slice of the plenoptic function that bounds them has a constant value, since the slice will both be larger, and be more likely to contain multiple, possibly complex, objects. A thought experiment of possible situations in which it would be remotely feasible to rely on the positioning estimates of a mobile robot was considered. It would be expected that
given ray would still most likely project to the same voxel at some maximum viewing distance. Secondly, projection of the image at this maximum viewing distance should not occupy more voxels than there are pixels in the input image. In fact, ideally, there ought to be roughly four times more pixels in the input image than voxels at the maximum projection. This viewing distance was roughly, but reasonably, chosen to be within ten meters. Were this system to be implemented on a mobile robot, it would be expected that the sensor model would be modified so that rays are not projected past ten meters. Given these assumptions, a twenty-centimeter square cube was chosen as an appropriate voxel size. This allows ray projections to be off by at least a degree in either direction and still fall easily within the correct voxel at a distance of ten meters. Obviously, as translational errors approach twenty centimeters, an increasing number of rays will be projected into the wrong voxel. Note that these are still nearly unreasonably small margins of error. Any agent to employ this system would, under most circumstances, be required to be running a self-localization algorithm as well. This system could also be employed in situations where the position and orientation of the camera can be manually or mechanically controlled. For most of the synthetic scenes this system was tested on, it was assumed that the position and orientation of the camera was precisely determined. The effects of errors in the camera’s position and location will be discussed later.

VI. Procedure

This technique operates by projecting pixels in the input image as rays back into the space and calculating which voxels each ray primarily passes through. The view volume of the camera is stored as a trapezoidal shape. The current position of the camera and its orientation are input along with each image. These values are used to compute a transformation matrix, which is used to place the trapezoidal volume in the correct location in the scene. Each of the four corners of the image are projected along the four edges of the trapezoidal volume, with all of the other pixels interpolating between the corners to fill the volume. A version of Bresenham’s algorithm for the rasterization of a geometric line modified to work in three dimensions is then used to assign a set of rays that pass primarily through a given voxel to that voxel. Bresenham’s algorithm does not fill every pixel a given line will pass through, but only one pixel at each value along the axis of the greatest extension of the line. Therefore, not all pixels contained within the projection of a given voxel onto the input image will necessarily be assigned, but a region in the center of the projection containing the majority of the pixels that project onto the surface of the voxel will. Thus, not all possible pixels affected the slice of the plenoptic function bounding each
voxel will be considered when calculating the average perceived light value of the surface of the voxel, but all of the most major contributions are used in this calculation. After this set is calculated, the light values of this set are averaged and that average value is used at the contribution of this input image to each voxel’s light value.

This light value is then used to calculate a probability distribution that is combined with the existing probability distribution of each voxel. This probability distribution represents the probability that that voxel will appear a given color when perceived. The color value that was actually perceived in a given input image is integrated into this probability by combining this probability distribution with the distribution that represents the probability of observing each possible color value assuming that the voxel really is the color that was observed. The light value is broken into an intensity component and a chromaticity component. The probability distribution is broken into a fixed number of regularly sized buckets or bins that cover the color space. The probability distributions are stored as an array of values that represent the area under the distribution curve within each bin. Given that there are $N$ bins, the a priori value of each bin is set to $\frac{1}{N}$. For the new probability distribution to be added, a value approximately equal to $\frac{2}{2N+1}$ is assigned to each bin that does not contain the color value to be integrated into the voxel. If the new color value lies in the center of a bin, this bin is given a value approximately equal to $\frac{3}{2N+1}$. Otherwise, the value $\frac{1}{2N+1}$ is divided between that bin and the next nearest bin (or, in two dimensions, the nearest one, two or three bins) by a linear interpolation based on the distance from the actual color value and the center of each bin. This is equivalent to calculating the area under a distribution is at an even value except for within a distance half the width of a bin from the observed color value, where the distribution has a slightly higher even value. The fact that square bins are used to cover the triangular area that forms the domain of the chromaticity probability distribution causes various special cases along the diagonal boundary when calculating these probabilities. Seize assumes that the distribution of sensor error present will most likely be normal, however since a step function is used as a probability distribution, care should be taken to overestimate sensor error. $N$ should be chosen such that the width of the bin, and therefore the raised area of the probability distribution, is larger than the standard deviation of the sensor error multiplied by a sufficiently large constant (such as 2 or 3). Note that
this is only an issue when reconstructing real environments, for synthetic scenes, the sensor error will be 0.

When this new probability distribution is calculated, it is combined with the existing one by multiplying the area under the distribution in each bin with the area under the other curve in the corresponding bin. The area under the resulting curve is then renormalized so that the area under the entire curve is equal to one. This process is always performed on the intensity distribution. However, color values with very high or very low intensity values can vary greatly in chromaticity with small changes in color values. Also, specularities on an object will produce a chromaticity that is much closer to white or gray than the actual chromaticity of the color of the object. For these reasons, if the intensity of the color value is either very high or low, the chromaticity probability distribution is not updated. This process is repeated for each input image taken. As long as the probability distributions are stored, additional sets of input images can be added to provide additional data without recalculating the effect of the initial input set. Note that since Bayes’ theorem is commutative, as it treats events as sets and not sequences, the procedure is commutative as well. That is, images taken from any camera position can be integrated into the probability distribution in any order, as there are no theoretical constraints on the position of the camera for any input image, or the order in which different images be integrated.

VII. Reconstruction

To reconstruct the most likely three-dimensional scene, the probability distributions for each voxel are analyzed. To do so, the mean and variance of each distribution is calculated. Since the area under the total distribution has been set to one, the mean, $\sigma$, is calculated as

$$\sigma = \sum_{i=0}^{N} c_i \cdot P_i$$

where $P_i$ is the probability in each bin, and $c_i$ is the color value at the center of each bin. Both $\sigma$ and $c_i$ will be a single value for the intensity distribution, and the vector $[r, g]$ for the chromaticity distribution. The variance, $\upsilon$, is calculated as

$$\upsilon = \sum_{i=0}^{N} P_i \cdot (c_i - \sigma)^2.$$ 

The variance and means of both distributions are each combined to get a three-dimensional variance vector and a mean vector. The magnitude of the variance vector is compared to an arbitrary threshold. If the magnitude of this vector is above the threshold, the voxel is considered to be transparent. Otherwise, the voxel is opaque with a color value equal to the mean vector. The
exception to this is that voxels with a low intensity variance and an intensity value that is either very high or low will always be opaque, regardless of the chromaticity variance. A voxel model of the scene can then be stored to be rendered at later times.

The value of the threshold determines how much variation is considered acceptable in the appearance of a voxel that should be considered opaque. Selecting an acceptable value for this proves difficult. If the value is set too high, then more artifacts are created. If the threshold is set too low, any voxel that is part of an object that is ever occluded is incorrectly determined to be transparent. Most scenes will have many false positives (artifacts) and false negatives (missing objects that should be there) even for optimal values of this threshold. Also, the optimal value for this threshold various from scene to scene. Fortunately, reconstruction is a fairly trivial process, so since the distribution should be saved anyway, so that more images can be added at later points in time, it is simple to manually select an appropriate threshold when reconstructing the scene.

VIII. Results

The simplest environment this system was tested with was a synthetic one containing a simple white cube in an otherwise black scene. Figures 2 and 3 below are reconstructions from 20 input images such as figure 1. These images spanned approximately a 90 degree range, where the camera was positioned on a horizontal plane below the cube. Figure 4 is a cross section of the variance map for the scene. The green area represents the places that were filled in as white, and the red box shows the actual position of the cube.
The elongated shape behind the cube and the region in front of the cube which are filled represent areas which were constantly either in front of or behind the cube from most camera positions. Thus, the system would have little information that would indicate that these areas were not in fact filled. This shows that this system can be used to calculate all positions for which it is probable that it is occupied by a given object.

A checkered wall was selected as a standard test case for this system because first of all, it is an object with detail on a barely small enough scale to give the system something to work with, yet large enough that there is little chance of the detail getting lost. Secondly, since there are predominately only two color values in the image, standard reconstruction techniques could often have difficulties with this scene. The reconstruction shown in Figures 5 and 6, below, is done from 20 input images, such as Figure 4, spread across a horizontal line in front of the wall. As well as having a number of artifacts behind the wall, there are a number that extend forward toward the line the camera moved along.
The reconstruction shown Figure 7 is done from 20 images taken from positions on a circle parallel to the wall. The artifacts in this reconstruction take the form of areas directly in front of each checker square of the opposite color. The third reconstruction, shown in Figure 8, is done from the combination of the two sets. Once again, Bayes’ theorem is order independent, so the same result occurs if either set is integrated into the probability distribution resulting from the other. The effects of the types of artifacts in each of the previous reconstructions are reduced in this third reconstruction, providing the most accurate reconstruction of the checkered wall. This shows that it is at least possible for a human operator to detect artifacts present in the reconstruction and seek out viewpoints that would provide evidence to help eliminate them. It may also be possible that there could exist an automated path-planning algorithm that could try to progressively reduce the error in the reconstructed scenes by seeking out appropriate new viewpoints.
In the reconstructions below, the checkered wall is again used, but this time with a multi-colored sphere in front of it. The first reconstruction, shown in Figure 9, is done with a very low variance threshold, whereas the second one, shown in figure 10, is done with a slightly higher one. The sphere is reconstructed reasonably well in both examples. However, this example shows how the selection of the threshold affects how scenes are reconstructed. Even in an example as simple as this one, it is difficult to successfully reconstruct objects that are ever occluded without greatly increasing the number of artifacts present in the reconstruction.
The checkerboard wall is again used in the reconstruction shown in figure 11, below. Once again, the camera moves along a line parallel to the wall. However, in this example, the camera’s position was artificially perturbed by up to half a meter and the orientation was perturbed up to 5 degrees to simulate camera position errors. Despite the differences between this reconstruction and the ones above, the primary features of the wall are still clearly constructed and the artifacts introduced are relatively minor. Thus, for scenes not containing small detail, the system proves to be reasonably robust in compensating for random error. The level of error simulated here should be within reasonable demands on the odometry systems of many available mobile agents.

Figure 11

The reconstruction shown below in Figures 14-18 is done from 100 images, such as the ones in Figures 12 and 13, of a brightly colored room with striped walls and various colored cubes along the walls. Two of the four walls of the room were mostly reconstructed, and another wall was partially reconstructed. Many of the cubes were partially reconstructed, and a couple of them were fully reconstructed, but produced large artifacts. Also, the striped wall was in general not reconstructed in places where it was occluded by the cubes at various points. This example shows a wide variety of the results this system can be expected to produce, as well as a wide assortment of the problems that this system regularly encounters. The position and orientation of the viewpoints was attempted to be roughly evenly distributed. However, it is clear that imbalances of the amount of information provided for various areas of the room were present, as the wall that escaped reconstruction was apparently often partially occluded and little other information was provided. Note that even though the variance threshold was set rather low, and
large parts of the walls were not reconstructed as a result, the walls that were reconstructed and a couple of the cubes have large associated artifacts present.
One attempt was made to use this system to model a real room. A Magellan Pro with a Cannon digital surveillance camera on a pan-tilt-zoom mount was driven down a table, stopping often to collect pictures of the room the table is in. One such picture is shown in figure 19. A view of the reconstruction from a similar angle is shown in figure 20. Due to the fact that the scale of detail in the scene is much smaller than the size of a voxel, the reconstruction is rather fuzzy and imprecise. However, there is enough correlation with the real scene to indicate that this method could prove fruitful given sufficient further development of the ideas presented here.

With the aid of an available robot control system with a vision module capable of performing face detection [11], which was slightly modified to periodically export processed images, the same Magellan Pro was driven through a room with three people in it. Skin tones
were located, and made into binary images, which were white where the faces, arms and legs of the occupants of the room were observed. These images were fed into this system, and the possible locations of the faces were located. One of the processed images is shown in Figure 21, and the reconstructed scene is shown in Figures 22 and 23. Figure 22 is a reconstruction from a camera position somewhat above the image in Figure 21. The person appearing in the middle in Figure 21 is in fact slightly in front of the other two people. The shape that appears on the bottom in Figure 22 and on the right in Figure 23 corresponds to that person’s face and the arms of the person on the left.

IX. Discussion

This system does have some possible applications without modification to the basic algorithm by introducing some pre-processing into the images, as in the face detection example. If patterns in the input images are recognized, and the regions occupied by distinct patterns are color coded in the input images, then this system could be used to report the statistically most likely positions of each pattern. Thus, if a moderately large number of patterns can be
recognized, this algorithm can be used to calculate the most likely positions of multiple instances of each of the patterns simultaneously. This in demonstrated with only a single pattern in the face-finding example above. This technique could be further enhanced if the color values assigned to each pattern are known to the reconstruction routine. In this case, instead of the variance from the mean of the distribution, the discrepancy, $\Delta$, of the distribution from a specified value could be calculated by the calculated as $\Delta = \sum_{i=0}^{N} P_i \left(c_i - \bar{\delta}\right)^2$, where $\bar{\delta}$ is the given color value and $c_i$ and $P_i$ are used as defined above. The voxels that report the lowest value for this discrepancy could be located, and the system could be asked to specify the N most likely voxels to contain the specified pattern, where N can be easily set by the user. However, the goal that this system was designed to achieve, the reconstruction of an arbitrary three-dimensional scene, remains a little more elusive that these simpler applications.

The main reason that this system is difficult to successfully apply to arbitrary situations is that occlusion is a very common phenomenon, especially if the scene to be modeled were, for instance, an area containing multiple rooms. Even when, as discussed above, the rays are not projected past a certain distance from the camera, there will be many cases where certain voxels which are occluded will project into the scene and appear an incorrect color. This makes selecting an appropriate variance threshold as discussed above an almost impossible task. One possible way to address this problem could be to develop a multiple-pass sensor model, in which the effects of the input image are different in each pass. One possibility is to add a second pass in which the probability distributions of the voxels affects those of their neighbors such that regions of coherence in the image are reflected by regions of similarities in the probability distributions of voxels in directions parallel to the image plane. Another is to specify that after a given number of images has affected a given region of voxels, the effects of occlusion are attempted to be counteracted. This could be attempted by re-applying all of the images to the probability distributions in such a way that the effects of a given ray do not strongly apply to voxels beyond the first few that that ray encounters which have a mean that closely corresponds to the color value of the pixel the ray represents. This assumes that such voxels would be in fact filled, and would occlude voxels behind them, so the effects of the ray should not heavily influence those regions. However, it would be difficult to implement this in such a way that it is not artifacts that are in fact strengthened, and the voxels that contain the actual object are not strengthened. Another possibility is to save images of various regions of the scene and not apply the affects until a moderate number of them have been collected. Various algorithms for stereo
correspondence with multiple cameras could be used to estimate depth information for each image, and the affects of each pixel could only be applied to voxels near that depth from the camera. However, most algorithms of this type require a greater accuracy of camera position than this system requires, so such a process would limit the applicability of this system to independent agents. Naturally, if a laser range finder could be used to generate a depth map of each image as it was taken, many of the problems inherent in this approach would simply go away [14].

Other methods could keep the same fundamental goal as this system, that is, building a voxel model of an arbitrary scene from a small number of samples of the plenoptic function, but approaching the problem in a fundamentally different way. If the voxels were treated as lumispheres [20] or another form of procedural voxel, each direction a given voxel was viewed from would have its own most likely color value, such that voxels appear different when viewed from different angles. This could be done by using all input images taken from within a small angular range to calculate the most likely color for that voxel to appear from that viewing range. Alternately, the value from the image taken closest to the voxel could be used, to guard against the effects of occlusion. The task then would be to analyze each voxel and to designate voxels as transparent or opaque by determining which voxels exhibit color maps that constitute believable reflectance fields. Since a scene can contain an arbitrary number of light sources, and indirect lighting may play a large role in the appearance of a given voxel, this must be based on the environment map of that voxel. It would be either necessary to use images from nearby camera positions to calculate such environment maps, or to render the environment maps using a partial scene reconstruction. This would make reconstruction take the form of a feedback process which would have to be set to attempt to converge on a configuration in which a given proportion of the voxels are filled. Also, even given an accurate environment map, extracting the effects of each component of the environment map and determining if the observed reflectance function is in fact a realistic one is a highly complex task. This is further compounded given that different materials generate highly different reflectance fields, with velvet being a simple example of a material with a highly anomalous reflectance field.

If one or more of these ideas are incorporated to a system such as this one, there are several advantages to the theoretical framework suggested here. The most obvious one is that additional input images can always be incorporated into any scene. The importance of selecting an appropriate sample set of images is shown above with the example with the checkered wall.
The most obvious way to address this by exploiting the principle that more data can always be incorporated is to design an interactive system based on this framework. In such a system, an operator can compare the output scene to the reconstructed scene and capture more input images and integrate them until she is satisfied with the result. However, it is most likely possible to automate this process such that it can be carried out by an independent agent. Since the probability distributions will always report a best guess as to the topology of the scene, it can be analyzed to suggest the most useful camera positions and orientations to increase the accuracy of the model. One such metric could be to attempt to move the camera into regions that are suspected to be empty but are near questionable areas that are currently being observed, but would allow the agent to view the area from an angle as greatly different from the current one as possible. The angles from which a given voxel has been viewed could also be stored, and attempts could be made to achieve the greatest coverage of viewing angles for all voxels in the scene. Another general advantage to using a Bayesian framework is that random errors in sampling are in general acceptable. By using probability distributions to integrate each image, given a high enough sample size, the effects of each image individually are negligible, so small, random errors will generally not adversely affect the reconstruction. Note that systematic errors will still cause problems, however, most errors of these types can be dealt with in other ways. Errors in position estimation when relying on wheel encoders are an example, since they gradually build up over time, but continuous localization systems can be constructed to mostly counteract this.

X. Conclusion

The system described here represents a novel approach to the task of the reconstruction of arbitrary scenes. This approach relies on a Bayesian statistics model to integrate samples into a continually evolving voxel representation of the scene being reconstructed. While successes at achieving this stated goal were minimal, it is still reasonable to believe that this framework can be further developed to address many of the problems encountered with this system. Furthermore, this system also presents a number of alternative applications, such as a statistically based position estimation system for target patterns. Overall, it is suggested that integrating a statistical analysis into certain types of vision tasks, such as the one addressed here, could prove to be quite fruitful if such techniques were further explored.
XI. References


