A Solution to the Next Best View Problem for Automated Surface Acquisition

Richard Pito, Member, IEEE

Abstract—A solution to the “next best view” (NBV) problem for automated surface acquisition is presented. The NBV problem is to determine which areas of a scanner’s viewing volume need to be scanned to sample all of the visible surfaces of an a priori unknown object and where to position/control the scanner to sample them. It is argued that solutions to the NBV problem are constrained by the other steps in a surface acquisition system and by the range scanner’s particular sampling physics. A method for determining the unscanned areas of the viewing volume is presented. In addition, a novel representation, positional space (PS), is presented which facilitates a solution to the NBV problem by representing what must be and what can be scanned in a single data structure. The number of costly computations needed to determine if an area of the viewing volume would be occluded from some scanning position is decoupled from the number of positions considered for the NBV, thus reducing the computational cost of choosing one. An automated surface acquisition systems designed to scan all visible surfaces of an a priori unknown object is demonstrated on real objects.

Index Terms—Active vision, next best view, sensor planning, range imaging, reverse engineering, automated surface acquisition, model acquisition.

1 INTRODUCTION

In recent years there has been a growing interest in the acquisition and representation of the surface geometry and topology of physical objects for use in reverse engineering and areas related to computer graphics. Interest in reverse engineering has been spurred by the growing number of legacy parts for which no CAD model exists. Interest in model acquisition has received attention from graphics related fields because of the need to populate 3D computer simulations, training programs, and virtual environments with realistic models of free-form physical objects. Using a range scanner to sample all of the visible surfaces of any but the most trivial of objects, however, requires that multiple range images be taken from different vantage points and integrated, i.e., merged, to form a complete model. An NBV algorithm determines each subsequent vantage point and offers the obvious benefit of reducing and/or eliminating the labor required to acquire an object’s surface geometry. It also enables the possibility of creating a more accurate and complete model by utilizing a physics based model of the range scanner.

This work is part of a fully automated surface acquisition system [1] designed to automatically acquire a model of the scannable surfaces of an a priori unknown object using low to medium accuracy range scanners. The system operates by incrementally adding surface data to a partial model of the object until all of its visible surfaces have been scanned. The process begins by placing the object, in an arbitrary pose, into the viewing volume of the system. A range image is taken and becomes the partial model to which subsequent range data is added. The NBV is determined, the object/scanner moved, and a new range image is taken. The new range data is then registered and integrated with the partial model and the process is repeated until the NBV algorithm determines that all surfaces have been scanned.

Traditionally, work in the areas of scanning, determining the NBV, registration, and integration has proceeded without considering how each solution affects and is affected by the others. This is especially true for solutions to the NBV problem, which are affected by the requirements of every other stage of the process. In addition, solutions to the NBV problem traditionally have not exploited a detailed model of the scanner. One of the goals of this work is to demonstrate the interdependence between a solution to the NBV problem and the other stages of an automated surface acquisition system, as well as the necessity and benefits of utilizing a model of the scanner when determining the NBV.

A brief outline of the rest of this paper follows. Section 2 reviews related literature and Section 3 outlines when an NBV algorithm should be used, the constraints to be addressed when choosing the NBV, the requirements of an NBV algorithm, and how related work fares under these criterion. Section 4 describes a general strategy for a solution to the NBV problem and methods to determine and represent the information needed by an algorithm implementing such a strategy. Section 5 describes the positional space data structure and the PS Algorithm for solving the NBV problem. Section 6 describes an implementation of the PS Algorithm and the results of scanning several objects. Finally, the good and bad features of the PS Algorithm, as well as future work, are summarized in Section 7.
2 Literature Review

Investigation of the NBV problem for range scanners falls into the area of active vision [2], [3]. Related work, using either range or intensity cameras, can be categorized as to whether or not the object’s geometry (including extent) and pose are known beforehand. For example, in the “art gallery problem” [4], [5] the objects’ geometries and poses are known beforehand and the task is to optimally position guards to monitor them. In assembly tasks [6], model-based object recognition [7], [8], [9], object searching tasks [10], and semi-automated scene reconstruction [11], the object’s geometry and a rough estimate of its pose are known. In inspection tasks using both range scanners [12], [13] and intensity cameras [14], [15], [16], a nearly perfect estimate of the object’s geometry, and possibly its pose, are known and the task is to determine how accurately the object has been manufactured. In reverse engineering, nearly everything about the object’s geometry is unknown except that it has a certain extent. Reverse engineering is addressed in this work and the relevant literature will be reviewed below. In primitive object reconstruction [17] and depth estimation tasks [18] using mobile intensity cameras, the problem is to choose camera motions which minimize error in the parameter estimation algorithms. In surface reconstruction tasks using intensity cameras [19], the problem is to control the camera’s motion to guarantee local and hence global surface reconstruction. Finally, in scene reconstruction, the task is to build a model of an unknown scene [20], perhaps for path planning [21] and potentially with unknown extent [5], [22]. See [23] for a survey of the sensor planning literature.

There are two fundamental problems to be solved when determining the NBV: deciding which areas of the viewing volume need to be scanned and determining how to position the range scanner to sample them. Nearly all researchers address the first problem by identifying range discontinuities either in each range image or in the model under construction [24], [25], [26], [27], [28], [29], [30], [31], [32]. One exception is the work of Whaite and Ferrie [33], which showed that those areas of a parametric solid(s) that would sample the most was chosen as the NBV.

Because of the computational burden of ray tracing, he also proposed a normal algorithm which chose the NBV from just eight potential positions and didn’t take occlusions into consideration. Banta et al. [28] used uniformly sized voxels, also tagged as either empty or not, to represent the viewing volume. The NBV was chosen from a set of three candidate positions which would scan directly into the three largest range discontinuities in the range data. The NBV was identified as the one that would sample the most nonempty voxels. Papadopoulos-Orfanos and Schmitt [30], [36] also utilized a volumetric representation, but concentrated on a solution to the NBV problem for a short field-of-view range scanner mounted on a robotic arm. Since the scanner had to be placed close to the object, special care was taken to avoid colliding with it.

Milroy et al. [29] represented the unsampled areas of the viewing volume with air vectors attached to each range sample on the boundary of a range image, called air points. Each air vector was oriented to lie along the boundary of the unseen space. Their orientation was based on whether or not the air point was on the boundary of the scanner’s imaging window, was along a range discontinuity, or the surface was too steep to be scanned beyond the air point. The NBV was determined by selecting a candidate air vector and ranking 125 potential positions based on how obliquely the range scanner would sample each air vector in its sampling window. Zha et al. [31] similarly represented the unsampled areas of the viewing volume with vectors “attached” to the boundaries of surface meshes.

3 The NBV Problem

This section discusses when an NBV algorithm should be considered for positioning a range camera. It also covers the general concerns which need to be addressed by a solution to the NBV problem, the specific constraints on the choice of the NBV and on the algorithm which computes it, and how existing NBV solutions fare under these criterion.

In general, an NBV algorithm should be considered when the class of objects to be scanned do not have a predictable shape or, if they do, when they do not have a predictable orientation. In these cases, no predetermined sampling pattern can guarantee that all of the visible surfaces of an object will be scanned. When used as part of an automated surface acquisition system, the primary
The purpose of an NBV algorithm is to ensure that all scannable surfaces of an object will be scanned. In addition, in an automated system, an NBV algorithm is necessary to determine when to stop scanning. Even in a supervised system, an NBV algorithm can be useful to position a range scanner to sample highly occluded surfaces, especially if the scanner has a nonuniform sampling pattern.

To demonstrate the utility of NBV algorithms, the outside surface of a coffee mug was scanned with and without the use of one. Fig. 1a is a rendered image of the surface of a mug that was automatically constructed using the NBV algorithm described in this work. After sampling all of the mug’s visible surfaces with eight range images, the algorithm self-terminated. The enlargement in Fig. 1b shows that the inside of the handle has been fully sampled. This is significant because that surface can only be sampled from a few positions in the scanner’s workspace. In contrast, Fig. 1c shows the same area of the inside of the handle as reconstructed from eight range images taken from equally spaced positions in the scanner’s workspace. In this case, the inside of the handle was not completely scanned.

The two fundamental problems to be addressed by a solution to the NBV problem are to determine what to scan and how to scan it. In addition, there are several design choices which influence the effectiveness of an NBV algorithm. Foremost of these is whether or not a full model of the range scanner is used when solving for the NBV. Range scanners sample in many different and sometimes exotic patterns. Without the use of a model of the scanner and its sampling pattern it is not possible to guarantee that an NBV algorithm can determine a position which will sample some area of the viewing volume. Algorithms which do not explicitly account for the sampling pattern of the range scanner when determining the NBV may not be quantizable for use with any range scanner.

Another critical design choice is whether or not a solution to the NBV problem will address constraints from the registration and integration steps of the model acquisition process. Registration is the process of removing the mechanical and/or calibration noise introduced into the global position/orientation of each range image by the movement of the scanner and/or object prior to the acquisition of each range image. Integration is the process of merging a set of range images into a consistent surface description of an object. Because the best performing registration algorithms [39], [40] and many algorithms for integrating range data [41], [42], [43], [44] require or perform best when the range data overlaps, registration and integration impose an overlap constraint on the choice of the NBV. Specifically, the overlap constraint requires that, from its next position, the scanner should resample part of the object already scanned. If a solution to the NBV problem does not address the overlap constraint [27], [34], [24], [28], [30], the registration and/or integration process may be unreliable and, so, the solution may not be suitable for use in an automated system. Other constraints arising from registration and integration will be explored in the next two sections.

3.1 What to Sample?
In order to keep track of which areas of the viewing volume need to be scanned, many techniques represent the viewing volume volumetrically and tag each area as either empty (no object) or unknown (possibly an object) [27], [34], [24], [28], [30]. Other methods record information about the 2D and 3D boundaries of range data [29], [31], [26] (and this work). The primary disadvantage of a volumetric approach is that a large object may be too costly to represent. All approaches, however, recognize the need to satisfy the fundamental constraint, the need to sample into some previously unseen portion of the viewing volume. Solutions which choose the NBV based on a global metric [31], [32], [33], however, are not guaranteed to satisfy the fundamental constraint, mostly because they do not check for occlusions.

The overlap constraint, related to the overlap constraint, is that the NBV algorithm should be able to identify those surfaces that will be rescanned from any position in the scanner’s workspace so it can be determined beforehand if registration using them will succeed. This is because registration algorithms generally perform better if the overlapping surfaces are smooth [31] and have normals which are not concentrated in one direction and which span the workspace volumetrically and tag each area as either empty or unknown (possibly an object) [27], [34], [24], [28], [30]. If the overlap identification constraint is not satisfied [27], [34], [30], [33], [32], it is not possible to verify that the surface(s) to be resampled have these properties and the registration procedure may not be robust.

3.2 How to Sample It?
A primary concern when designing an NBV algorithm is the representation of the workspace of the scanner, especially with respect to whether it is discrete or continuous. Let the scanning parameters $x$ be a set of variables which will uniquely determine a position, orientation, and set of sampling parameters, e.g., zoom factor, for a range scanner. Let the NBV be chosen from the set of potential scanning parameters $X$. For both continuous and discrete representations of $X$, it is desirable that it be complete enough to ensure that the scanner can be accurately positioned to sample highly occluded surfaces. Discrete representations, therefore, should be quantized finely enough to allow for a precise positioning of the scanner. In this case, a trade-off exists between the ability to
precisely control a scanner’s location, i.e., the size of X, and the computational burden of having to check more positions to determine the NBV.

Continuous representations of X have been demonstrated primarily for inspection tasks when using intensity cameras [16], time-of-flight range scanners [14], and triangulation-based range scanners in 3D [34], [35] and in 2D [27], [15]. Continuous representations and discrete ones that have topologies defined for them [12] have the advantage over other discrete representations in that they do not suffer from aliasing effects and well-understood minimization techniques can be used to solve for the NBV. On the other hand, implicit representations of X are difficult to construct, can be prohibitively complex to incorporate realistic models (hence, their appeal in sensor placement setup). Finally, when using a continuous representation, it can be prohibitively complex to incorporate realistic models of the sampling pattern of a range scanner into the representation of X, especially when the model relating image coordinates to world coordinates is highly nonlinear [47], [37]. The main disadvantage of a discrete representation of X, besides aliasing effects, is having to represent, and hence consider, a large number of potential scanning parameters for the NBV. In summary, although continuous representations of X can be more accurate and easier to use than discrete ones, they may become prohibitively complex when modeling scanners with complex sampling patterns and/or workspaces. Discrete representations, on the other hand, are more generalizable because they can be easily created for complex workspaces and scanners, but may require an NBV algorithm to consider a large number of potential scanning positions for the NBV.

An NBV algorithm must be able to determine which areas of the viewing volume are obscured from any particular position in the scanner’s workspace. This scanning constraint is perhaps the most important constraint to satisfy when choosing the NBV. For example, for an active triangulation-based range scanner to scan some area of the viewing volume, there must be an unobstructed line of sight into that area for both the projected light source and the receiving optics. In order to satisfy the scanning constraint, it is necessary to incorporate an accurate model of the range scanner, and, especially, its sampling pattern, into an NBV algorithm. Solutions which do not take into consideration occlusions [24], [29], [33], a complete and accurate model of the range scanner [27], [34], [31], [24], [28], [30], [29], or only consider a small number of scanning positions [29], [28] may not be able to accurately position the range scanner to sample the area(s) of interest.

Because satisfying the scanning constraint normally involves some form of ray tracing, it is computationally expensive to enforce and poses a serious burden when considering a large number of candidate positions for the NBV. Solutions which perform occlusion checking for a large number of potential scanning parameters [12] are generally not practical on-line solutions. The most serious challenge for any NBV algorithm that represents X discretely, therefore, is to maximize the size of this set while, at the same time, enforcing the scanning constraint and remaining computationally feasible.

An NBV algorithm should also be able to enforce a minimum tolerance on the accuracy of the range samples which make up the final model. Since the accuracy of range samples from nearly all range scanners depends on the orientation of the scanning direction to the surface normal [44], [48], the algorithm should be able to position the scanner to resample poorly sampled surfaces with at least a prespecified minimum confidence. In order to do this, the algorithm must satisfy the overlap identification constraint.

Finally, the NBV algorithm itself should exhibit self-termination, be generalizable to any range scanner and scanning setup, and make no assumptions about the geometry or topology of the object being scanned. Table 1 lists all of the criteria for the choice of the NBV and the desirable properties of an NBV algorithm. In summary, the main challenges to be overcome by a solution to the NBV problem are 1) to consider a large number of potential scanning parameters and to enforce the scanning and overlap constraints while at the same time remaining computationally feasible and 2) to incorporate an accurate model of range data acquisition into the algorithm.

4 Strategy

The algorithm presented here, the PS Algorithm, uses the common strategy of keeping track of the “unseen” portions of the viewing volume to identify what needs to be scanned. An area of the viewing volume is “unseen” if it hasn’t been scanned due to omission or occlusion. When a range image is taken, the viewing volume is partitioned into seen and unseen areas, see Fig. 2. Those areas into which the scanner did not sample are called the void volume, the scanned surface of the object the seen surface, and the surface separating the void volume from the rest of the viewing volume the void surface.

To satisfy the fundamental and overlap constraints, the scanner should, from its next position, scan some previously unseen portion(s) of the viewing volume and part of the object already scanned. To satisfy the fundamental constraint, the scanner is positioned to sample the void surface and, therefore, into the void volume, thus reducing the unseen portion of the viewing volume. To satisfy the overlap constraint, the scanner is positioned to resample part of the surface of the object already scanned. By repeatedly satisfying the fundamental constraint, the size of the void volume will decrease monotonically until no portion of it will be accessible by the scanner and the scanning process will terminate. This strategy is depicted in Fig. 3, which shows, in (a), the scanner at its next best position and, in (b), the partitioning of the viewing volume after a range image is taken.

Therefore, to satisfy the fundamental and overlap constraints three pieces of information are needed:

1. the void volume, i.e., a description of what must be scanned,
2. the sampling pattern of the range scanner and a description of its workspace, i.e., a description of what can be scanned from what vantage points, and
The following sections describe representations for the scanner (Section 4.1), the partial model (Section 4.2), and the void volume (Section 4.3) that are used in solving for the NBV. Section 4.3 also describes a technique to represent the areas of the viewing volume that must be scanned without actually determining the void volume. All three pieces of information will be used to solve for the NBV by the PS algorithm described in Section 5.

4.1 Representing the Scanner

Each sample of a dense sampling range scanner is taken along a predetermined ray, or ranging ray (RR), in the scanner’s coordinate system. The origin and direction of each RR is preset by the design of the scanner; only the distance to the scanned surface is determined when a range image is taken. For example, Fig. 4 shows the RRs of a laser stripe scanner which is moved along a linear stage to acquire a single range image. The RRs of many range scanners, however, are not so neatly arranged [37], [38]. If the scanner is triangulation-based, each RR also has associated with it one (or more) lines of sight from the receiving optics of the scanner to any point along the RR that is inside the scanner’s sampling area. Let \( \vec{r} \) denote an RR and \( \vec{l}_{\vec{r}} \) its line of sight. Fig. 5 depicts the relationship between \( \vec{r} \), \( \vec{l}_{\vec{r}} \), and the surface normal \( \vec{n} \) at the point where a range measurement is taken. The angle \( \theta \) between \( \vec{r} \) and \( \vec{l}_{\vec{r}} \) is determined by the design of the particular scanner used.

The confidence \( c_{\vec{r}} \) of a range sample \( \vec{r} \) depends on the physics of ranging used by a particular scanner and nearly always on the angle \( \theta \) between \( \vec{r} \) and \( \vec{n} \). For example, for triangulation-based range scanners, a typical measure of confidence is \( c_{\vec{r}} = \cos(\theta) \) [44], [48]. Because of
In this relationship, the confidence in a range sample can be used to determine if it is acceptable or not. If \( \theta_r \) is greater than the breakdown angle \( (\theta_b) \) of the scanner, then the range measurement is unacceptable. The breakdown angle is determined as the point of unacceptable degradation in range sample accuracy. It must be the case that \( \theta_b < 90^\circ \), however, in most scanners, \( \theta_b \approx 60^\circ \). The scanner is represented by its RRs, their lines of sight if it is a triangulation-based scanner, and its breakdown angle.

### 4.2 Representing the Seen Surface

The seen surface of the object is represented by the partial model of the object constructed so far and is used to satisfy the scanning, overlap, and overlap identification constraints. Because only gross geometry is needed, the partial model is decimated [49]. The light gray triangles in Fig. 6 represent the partial model of a coffee cup. The overlap and overlap identification constraints can be satisfied by noting which parts of the partial model are intersected by the RRs of the scanner when it is placed at some position in its workspace. Similarly, the scanning constraint can be satisfied by noting which areas of the scene are obscured by the partial model. The partial model, therefore, acts as both a target (overlap/overlap identification constraint) and obstacle (scanning constraint) when determining the NBV.

Each triangle of the partial model is scannable only from those directions that form an angle of \( \theta_b \) or less with the triangle’s normal, see Fig. 7. Each of these directions, or observation rays (ORs), represents a “potential” RR and, so, has a confidence equivalent to that of a collinear RR.

### 4.3 Representing the Void Volume

The representation of the void volume is more problematic than that of the seen surface because a faithful reproduction would require intersecting the set of boundary or solid model representations of the unseen areas of the viewing volume determined by each range image taken. Such an approach could be computationally burdensome, even on a moderately complex or large object, although see [34] for an example of such an approach.

It is not necessary, however, to explicitly represent the entire void surface for the purpose of determining the NBV. Since the object’s surface must continue into the void volume from the boundary of the seen surface, this is the best place to look for more of the object’s surface. Because of this, the function of the void surface as a target for satisfying the fundamental constraint can be replaced by a series of small rectangular void patches attached to the edges of the partial model and oriented to lie on the void surface. The two darker triangles in Fig. 6 together comprise a single void patch. The orientation of a void patch is, of course, crucial and, if each range image is considered in isolation, can be computed from the orientation of the RR that sampled the boundary of the seen surface [29], [31]. The true orientation of each void patch, however, is defined by the void surface and, so, must take into consideration all range images taken so far. Section 4.3.1 describes how to determine which areas of space around an edge of the model are free of other surfaces of the object based on the RRs which have passed.
near that edge and Section 4.3.2 describes an algorithm for orienting each void patch based on this information.

4.3.1 Determining the Free Space Around an Edge

The orientation of the void surface near the seen surface is determined by keeping track of the areas around each boundary edge of the seen surface which are known to be free of any surfaces of the object, i.e., the free space around each edge. Each RR which passes near a boundary edge determines which regions near it are free of surfaces of the object based on how it interacts with the object’s surface geometry. There are four type of interactions, or different ways, an RR classifies an edge: outer jump edge, inner jump edge, line-of-sight jump edge, and regular.

For example, referring to Fig. 8, the way in which an RR classifies the edge e of triangle T is determined by ray-tracing its “neighboring” RR through the viewing volume. The RR is equivalent to except that its origin is offset in the direction orthogonal to the edge e and away from T by the maximum distance between the origins of and any of its closest neighbors in a range image. In the figure, e lies along a range discontinuity with measured point p and q on either side of it. In this case, because q lies farther from the scanner than p does, the edge is classified as an outer jump edge and e’s void patch, shown as two dark triangles, is oriented to represent the boundary of the scanned and unscanned areas of the viewing volume near e.

Considering each edge as the central axis of a cylinder, regions of space around it can be specified by a solid angle. In order to interpret such angles, a local coordinate system is defined for each edge. Referring to Fig. 9a, the edge e of triangle T induces a local coordinate system defined by the three vectors, e, n, and z, where n is T’s normal, e lies collinear with e, and z = e × n. The determination of the orientation of the edge’s void patch, specified as the angle , is done in the plane containing z and n which, respectively, define the x and y axes for angular measurements. For example, the angle of n in this plane is π/2 and of z is 0.

The free space map induced by the RR around an edge e is a partition of the interval [0...2π) that classifies the spatial regions around e as either free, conditionally free, or unknown. Any area classified as free is guaranteed not to have any surfaces in it, while any classified as unknown may or may not (conditionally free regions will be discussed later). An edge’s void patch is oriented to have an angle , which is the maximum angle such that the interval [0...] has been classified as free. Fig. 9b shows the free space map induced by on the edge shown in Fig. 8.

An inner jump edge is similar to an outer jump edge except that the edge in question is farther from the scanner than the surface on the other side of the range discontinuity. The free space map for such an edge is shown in Fig. 10a. A line-of-sight jump edge is similar to an inner jump edge except that the line of sight back to the receiving optics of a triangulation-based range scanner is obstructed instead of its projected light source. The free space map for a line of sight jump edge is shown in Fig. 10b. Finally, the most common (and most complex) classification is that of regular edge which is identified when doesn’t intersect any surface of the partial model that it could have actually sampled. The free space map of a regular edge is shown in Fig. 10c. The regular edge contains two unknown, one free, and one conditionally free region. The largest unknown region, counterclockwise from v to in the figure, is bounded by v, which represents the orientation of a surface which would be just slightly too oblique to have been scanned by . This is because the normal of such a surface, represented by the radial marked as w, forms an angle of with . Similarly, the smaller unknown area between and the radial u is unknown because u represents the orientation of a surface which is too oblique to have been scanned since
its normal $x$ also forms an angle of $\theta_b$ with $\ddot{r}$. The light gray region is marked as conditionally free because it can be considered free only if it can be established that there are no surfaces in the smaller unknown region.

### 4.3.2 Orienting a Void Patch

The orientation of the void patch of each edge $e$ of the partial model is determined separately and takes into consideration all of the RRs which have an unobstructed view of $e$. The void patches of the partial model collectively define the void surface near its boundary.

Let $R = \{\ddot{r}_1, \ldots, \ddot{r}_m\}$ be the set of RRs which pass through the viewing volume near a boundary edge $e$ such that either endpoint of $e$ is visible by $\ddot{r}$. A point $p$ is visible by an RR $\ddot{r}$ if the ray $\langle \ddot{r}, p \rangle$ and $\ddot{r}$, if the scanner is triangulation-based, originating at $p$ does not intersect the partial model. All RRs are stored in a coarse voxel partition of the viewing volume so that, given an edge $e$, the members of $R$ can be easily determined. Furthermore, let the members of $R$ be sorted in increasing order of their angle about the edge $e$ in its local coordinate system, except that $\ddot{r}_1$ is always the RR which sampled one of the two points of $e$. The free space map for $e$ is initially set to the free space map defined by $\ddot{r}_1$ and is grown by merging it, in order, with the free space maps defined by the remaining RRs $\ddot{r}_j \in R, j = 2, \ldots, m$. The free space map of $\ddot{r}_j$ can be merged with that of $e$ if $\ddot{r}_j$ approaches $e$ from an angle which has been designated as free. If this is not the case, then $\ddot{r}_j$ can tell us nothing about the free space around $e$ since we don’t know if $\ddot{r}_j$ could have actually sampled $e$. The free space map of each subsequent $\ddot{r}_j$ is merged with $e$’s free space map until one of them cannot be merged due to this restriction. Two free space maps are merged by taking an inclusive OR of their ranges marked as free, converting every conditionally free region that does not have an unknown region preceding it to a free region, taking an inclusive OR of the remaining conditionally free regions, and setting the remaining regions in the resulting free space map to unknown, see [1]. Fig. 11 shows the result of merging two free space maps derived from regular edge classifications.

#### 5 THE PS ALGORITHM

To determine the NBV, the PS Algorithm makes use of the RRs generated by the scanner from each position in its workspace $X$ and the partial model and its void patches. The algorithm relies on the observation that if an RR from the scanner hypothetically placed at a particular location in its workspace is collinear with an OR of a surface to be scanned, the scanner will sample that surface if a range image were taken from that location. The algorithm works by generating the ORs of the void patches and the partial model and finding the positions in $X$ where the scanner’s RRs are collinear with these ORs. Positional space (PS) is a placeholder for RRs and ORs which facilitates the determination of how many RRs and ORs are collinear, aids in satisfying many of the constraints/properties listed in Table 1, and significantly reduces the computational burden of determining the NBV. Section 5.1 defines positional space, Section 5.2 describes how the three important kinds of information needed to solve for the NBV are represented in positional space, and Section 5.3 details the PS Algorithm.

#### 5.1 Positional Space

In order to act as a placeholder for ranging/observation rays (ROs), positional space consists of two scalar fields which respectively encode a point on and orientation of an ROR. An ROR is encoded as a single point in PS. One part of the PS representation, the positional space surface (PSS), is represented as one or more reference surfaces enclosing the viewing volume. An ROR’s point of intersection with the
PS encodes, in the parameters spanning the PSS, a point on the ROR. The second portion of the PS representation, the positional space directions (PSD), is represented as a polar coordinate system attached to each point of the PSS. The ROR’s local direction, when it intersects the PSS, encodes, in the parameters spanning the PSD, the direction of the ROR. Similar data structures, e.g., light slabs [50], have been used to represent the radiance along lines passing through a scene for the purposes of creating images of the scene from arbitrary viewpoints.

Both the PSS and PSD are discretized for representation. In the generic case, the PSS is best represented as a tesselated sphere which encloses the viewing volume. Fig. 12 shows a meridians and parallels tesselation of a spherical PSS, although others tesselations are possible [51]. The PSD is best represented as a local polar coordinate system for each cell of the PSS. Fig. 12 also shows the PSD for one cell of the PSS and that cell’s normal. In this case, therefore, positional space is represented as a 4 dimensional scalar field \( P(w, y, \theta, \phi) \), where the PSS is spanned by \( w \) and \( y \) and the PSD is spanned by \( \theta \) and \( \phi \).

The PS representation of an ROR \( \mathbf{r} \) is determined by first intersecting \( \mathbf{r} \) with the PSS to determine values for \( w \) and \( y \) and then converting its direction, in the coordinate system defined for that cell of the PSS, into polar coordinates to determine values for \( \theta \) and \( \phi \). This is referred to as projecting \( \mathbf{r} \) to PS and is written as \( \mathbf{r} \rightarrow (w, y, \theta, \phi) \). Key to using PS when solving for the NBV is to note that an RR and an OR are collinear if they project to the same point in PS. When using a discrete representation of positional space, two RORs which project to the same cell are collinear to within an error determined by its quantization.

5.2 Images in Positional Space

The image in PS of a scanner at some point in its workspace is formed by projecting all of its RRs into PS. Similarly, the image in PS of the partial model or its void patches is formed by projecting all of the ORs of its triangles into PS. A PS image of the scanner at position \( x \in X \) in its workspace, denoted \( P_v^s \), is formed by setting to 1 the value of each cell of \( P_v^s \) that an RR \( \mathbf{r} \) of the scanner projects to:

\[
P_v^s(w, y, \theta, \phi) = \begin{cases} 1 & \text{if some RR } \mathbf{r} \rightarrow (w, y, \theta, \phi) \\ 0 & \text{otherwise.} \end{cases}
\]

Each image can be computed off-line and economically stored on disk since normally only a small fraction of the points in \( P_v^s \) will have been set. For certain kinds of workspaces and PSSs, however, it is possible to derive each \( P_v^s \) by translating a single image \( P_v^s \) in PS [1].

The void surface’s image in PS, \( P_v \), is the weighted sum of all of the confidences of the ORs of the triangles of all of the void patches that project to PS:

\[
P_v(w, y, \theta, \phi) = \sum_{\mathbf{r} \rightarrow (w, y, \theta, \phi)} ||e_T|| c_\mathbf{r},
\]

where \( \mathbf{r} \) is an OR of a void patch that projects to positional space, \( ||e_T|| \) is the area of the model’s triangle that the void patch is attached to, and \( c_\mathbf{r} \) is the confidence of \( \mathbf{r} \) as explained in Section 4.2.

The ORs of a triangle are triangle projected to PS by first enumerating the set of candidate cells of the PSS that ORs of the triangle intersect. The set is initialized to contain that cell of the PSS that is intersected by a ray that passes through the center of the triangle and is collinear with its normal. The set is then grown by adding cells that are adjacent to ones already in the set. An adjacent cell is added if the angle between the triangle’s surface normal and the ray connecting its center to that of the cell does not exceed \( \theta_s \). Once the candidate set is determined, an OR connecting the triangle’s center to each cell in the set is projected to PS only if there are no surfaces of the partial model which lie between the OR’s triangle and the PSS. Also, if a triangulation-based range scanner is used, the line-of-sight of an RR collinear with the OR must also be ray traced through the viewing volume to see if it intersects some part of the partial model. If it does, then the OR is not projected to PS. Each OR which is projected into PS represents a direction from which it is possible to scan its triangle. The culling of candidate ORs by seeing if they (or their lines of sight) intersect the partial model is how the scanning constraint is enforced.

Finally, the representation of the seen surface in PS, \( P_v \), is the weighted sum of all the confidences of the observation rays of the triangles of the partial model that project to PS:

\[
P_s(w, y, \theta, \phi) = \sum_{\mathbf{r} \rightarrow (w, y, \theta, \phi)} ||T|| c_\mathbf{r},
\]

where \( \mathbf{r} \) is an OR of a triangle \( T \) of the partial model and \( ||T|| \) is the area of \( T \).

5.3 Solving for the Next Best View

The NBV is chosen as the position \( x_{b} \in X = \{x_1, \ldots, x_n\} \) that samples as many void patches as possible (to satisfy the fundamental constraint) while, at the same time, resampling at least a certain amount of the partial model (to satisfy the overlap constraint). The determination of how “much” of the void surface and partial model can be scanned from each potential position of the scanner \( x \) can be quickly determined by intersecting \( P_v \) with \( P_v \) and \( P_s \). The NBV is determined by maximizing the objective function \( N() \) over the elements in \( X \):

\[
\max_{i \leq i \leq n} N(i) = o(\alpha(v(i), \alpha(s(i))).
\]
The parameters of $o$ are understood to be the confidence weighted area of void patch and partial model visible by the range scanner at position $x_i$. These are defined as:

$$o_v(i) = \sum_{w} \sum_{y} \sum_{\theta} \sum_{\phi} P^v(w, y, \theta, \phi) P^v(w, y, \theta, \phi)$$

$$o_s(i) = \sum_{w} \sum_{y} \sum_{\theta} \sum_{\phi} P^s(w, y, \theta, \phi) P_s(w, y, \theta, \phi).$$

In order to view as many void patches as possible while ensuring that at least a certain amount of the seen surface is resampled, let

$$o(v, s) = \begin{cases} v & \text{if } s > t \\ 0 & \text{otherwise} \end{cases}$$

for some threshold $t$. The choice of $t$ is roughly dependent on the complexity and size of the object and is a topic for future work, see Section 7.

The objective function is maximized by inspection; the value of $N(i)$ is computed for each element of $X$. This maximization is computationally $O(nC)$, where $C$ is the average number of nonempty cells in all of the $P_i$. In practice, this is by far the least computationally expensive step of the PS Algorithm. The process of ray tracing ORs through the viewing volume in order to form $P_s$ and $P_c$ is far more expensive. The algorithm terminates when the value of $o_v(i) \leq F$ for all $x_i \in X$. The PS Algorithm gives no consideration for the errors introduced into the global position of range data by the motion of the scanner and/or object between scans since these are minimized by the registration algorithm. Finally, the algorithm can be easily extended to satisfy the tolerance constraint, see the Appendix.

The PS algorithm is capable of considering many potential positions for the NBV because of the way in which positional space works as a placeholder of observation and ranging rays. Specifically, the expensive ray tracing operations needed to enforce the scanning constraint are performed only for those cells of the PSS from which a triangle could be scanned and not for each potential position of the scanner in its workspace. Thus, the computational burden of determining visibility is decoupled from the number of positions considered for the NBV. In addition, since each $P_i$ implicitly encodes the sampling pattern of the range scanner, the PS algorithm is capable of positioning a range scanner to sample even highly occluded surfaces regardless of the complexity of the scanner’s sampling pattern.

### 6 Experimental Results

The PS Algorithm was implemented as part of an automated surface acquisition system which used a triangulation based range scanner. The acquisition process is the familiar sequence of four repeated steps: scan, register the new data with the partial model under construction, integrate the new data with the partial model, determine the NBV, and repeat until the NBV algorithm terminates.

#### 6.1 Experimental Setup

A top-down view of the scanning setup is shown in Fig. 14. The object is placed on a turntable which is rotated between scans. The scanner, a Cyberware PS scanner, simultaneously acquires 250 range samples along a vertical stripe and produces a range image by rotating $90^\circ$ about a central axes while acquiring 240 such vertical stripes. The scanned surfaces of the object are highlighted in the figure. The viewing volume is defined to be a cylinder 30 cm in height and 30 cm in diameter with the turntable as its base. Therefore, a single parameter $\alpha$, i.e., rotation of the turntable, determines the location of the scanner in its workspace. The turntable could be rotated to every $4^\circ$ and, so, the set of scanning parameters is $X = \{x^1 = \{\alpha = 0^\circ\}, x^2 = \{\alpha = 4^\circ\} \ldots x^{256} = \{\alpha = 256^\circ\}\}$. 

---

Fig. 13. Steps of the PS Algorithm.
6.2 Representation of the PSS

The PSS for this setup is represented as a cylinder enclosing the viewing volume. It is discretized into uniformly shaped cells 20 along the height and 90 along the circumference of the cylinder. Fig. 15 is a representation of $P_v$ formed using the void patches shown in Fig. 6. Each cell of the PSS is shaded in proportion to the cumulative confidence of the ORs of all void patches that projected into that cell. The darker the cell, the more "void patch" can be observed (in all directions) from that vantage point. The dark cells on the far left were formed from ORs of void patches on the inside of the mug’s handle which are not visible from this angle. Since the RRs of the scanner are all nearly parallel to the turntable, all ORs which are also not parallel to the turntable could never align with an RR and, so, are not projected to PS. Fig. 16 similarly shows a representation of $P_s$ formed using the partial model of the mug. The darker a cell is in the image, the more surface of the partial model can be observed (in all directions) from that cell. Finally, Fig. 17 shows a representation of $P_1$, i.e., a representation of the image in positional space of the range scanner at position 1 in its workspace. Each cell of the PSS which is intersected by an RR of the scanner is colored gray. The directions of some of the RRs of the scanner are shown as lines extending into the viewing volume.

6.3 Acquired Models

Fig. 18 shows photographs of a telephone receiver and rendered images of a surface mesh automatically acquired of it. Since the scanner cannot be positioned to sample the top or bottom of the object, it was automatically scanned in three different orientations on the turntable. The three composite meshes were manually registered and integrated to form the final model. It is necessary to acquire multiple composite scans not only due to the limited workspace of the specific scanner used in this work, but also, more generally, because there are currently no techniques which can be used to suspend an object so that it is scannable from all directions. The final model consists of 25,000 triangles and was integrated from 17 range images consisting of 161,665 triangles. Fig. 19 shows top-down views of the six range images taken of the receiver when it was placed on its side on the turntable. The final mesh and the general directions of the RRs of the scanner are superimposed in each image. The darker regions represent the data actually acquired in each range image. Note how each range image overlaps slightly with the data already acquired while sampling a new portion of the object (the last range image sampled a small area on the mouthpiece which had escaped the previous scans).

Fig. 20 demonstrates how the PS Algorithm deals with self-occluding objects by showing the acquisition process of
a coffee mug. Images (a) through (h) show the eight range images taken of the mug and the general directions of the scanner’s RRs, while image (i) shows the final model. Fig. 21 shows two rendered images of the model of the mug showing that the surface on and behind its handle was completely scanned. When a predetermined, i.e., nonda driven, scanning strategy was used to acquire this mug with the same number of range images the highly occluded surface on the inside of the handle was not acquired, see Fig. 1, Section 3.

Fig. 22 shows a photograph and four rendered images of a 207,000 triangle model of a 15 cm tall gargoyle candleholder created from three composite scans, 31 range images, and 1,119,000 triangles. Small details such as those around the eyes were not acquired due to the sampling limitations of the range scanner.

Finally, Table 2 shows the time consumed by each phase of the acquisition process of the coffee mug. In this case, eight range images consisting of a total of 70,151 triangles were taken of the mug to produce a model of 28,755 triangles. Time is measured in elapsed wall-clock time. The total time spent minimizing \( N() \) was less than 8 seconds, a small fraction of the time spent computing \( P_l \) and \( P_r \). The hardware used was a Silicon Graphics Indigo Impact with a 250MHz IP22 processor.

7 CONCLUSIONS

It has been argued that the core issues of the NBV problem are 1) to enforce the scanning and overlap constraints while at the same time maximizing the size of the set \( \mathbf{X} \) of candidate positions for the NBV and maintaining a computationally feasible solution and 2) to incorporate an accurate model of range data acquisition into the choice of the NBV. It has been shown how considering the NBV in the larger context of an entire surface acquisition system leads to the identification of the overlap, overlap identification, and tolerance constraints for the choice of the NBV. In addition, it is argued that it is necessary to incorporate into an NBV algorithm a complete model of the range scanner’s sampling pattern and method of sampling range. In
conjunction with an analysis of how ranging rays determine the free and unknown space around the edges of a surface model, void patches are proposed as an efficient way to represent the unseen parts of the viewing volume that must be scanned. Finally, to solve for the NBV, positional space has been introduced as a useful representation for what has been scanned (the partial model of the object), what must be scanned (the void patches), and what can be scanned (the images in PS of the range scanner for each element of X).

The salient features of the PS Algorithm are that it can consider a large number of potential scanning positions, it can accurately position a range scanner to sample highly occluded surfaces and it can be used with a wide variety of scanners. The first feature is true because it decouples expensive ray tracing operations from the number of positions considered for the NBV and the last two are true because it incorporates a model of the range scanner into its choice for the NBV. In addition, the PS Algorithm addresses the fundamental, scanning, overlap, overlap identification, and tolerance constraints when choosing the NBV, while the algorithm itself is (theoretically) self-terminating. Putting aside performance considerations which may arise from a highly discretized positional space representation or, to a lesser extent, a range scanner which has a very large workspace (issues to be addressed in future work), the PS Algorithm is generalizable to any range scanner and scanning setup in which the scanner is not positioned inside the convex hull of the object. Finally, the algorithm makes no special assumptions about the geometry or topology of the object being scanned except that it fit within the viewing volume of the system, a standard assumption when performing object model acquisition.

Future work for the PS Algorithm will center around increasing its speed, eliminating the need for threshold parameters, and improving the metric used to enforce the
TABLE 2
Wall-Clock Time Consumed by Each Phase of the Acquisition Process

<table>
<thead>
<tr>
<th>category</th>
<th>NBV scanning</th>
<th>registration</th>
<th>integration</th>
<th>overhead</th>
<th>total</th>
</tr>
</thead>
<tbody>
<tr>
<td>time (min:sec)</td>
<td>13:15</td>
<td>5:01</td>
<td>7:28</td>
<td>7:37</td>
<td>3:53</td>
</tr>
</tbody>
</table>

overlap constraint. The PS Algorithm spends a lot of time projecting ORs through the viewing volume. A simple extension which would dramatically increase its speed and allow for a higher discretization of positional space would be to avoid projecting all of the void patches and triangles of the partial model into PS. This can be done by noting that any void patch \( t_v \) that does not project to PS should be scanned. Once such a \( t_v \) projects to PS, only its neighboring void patches and the nearby triangles of the partial model need be projected to PS. Also, when the range scanner has a very large workspace, the computational burden of choosing the NBV by computing \( N() \) for each position in the workspace may be too costly. In this case, the PS Algorithm would benefit from extensions that would trim the number of positions considered, for example by considering only those positions which are near the current position.

In addition, due to the discrete nature of positional space, there is a trade-off between increasing the resolution of positional space in order to increase the accuracy of positioning the scanner and decreasing its resolution to reduce the number of ORs which must be projected in order to form \( P_s \) and \( P_v \). The discrete nature of positional space and the aliasing effects it produces necessitate the use of an experimentally determined nonzero termination threshold \( F \). Methods to automatically determine this threshold or obviate its need should be investigated.

Methods also need to be developed to automatically determine the value for \( t \), the minimum confidence weighted area of the partial model that should be rescanned from the NBV. In addition, other metrics should be investigated for \( t \) because robust registration depends more on the geometry of the overlapping regions than on their area [45], [46]. Since the PS Algorithm does satisfy the overlap identification constraint, metrics such as the smoothness of the resampled surfaces could be used [31], [40].

APPENDIX

SATISFYING TOLERANCE CONSTRAINTS

To sample all visible surfaces of the object with at least a minimum confidence \( u \), ORs of those triangles of the partial model with lower confidences than \( u \) can be projected to \( P_v \) so that an NBV will be chosen that will resample them with a higher confidence. The confidence of a triangle \( T \) is defined as \( \epsilon_T = \min(p_1, p_2, p_3) \), where the \( p_i \) are the vertices of \( T \). The confidence of a vertex \( p \) is \( \epsilon_p = \bar{n} \cdot \bar{r} \), where \( \bar{n} \) is the average normal of all triangles \( p \) is a member of and \( \bar{r} \) is the RR which sampled \( p \). To take into account tolerance constraints, \( P_v \) is redefined to be:

\[
P^T_v(w,y,\theta,\phi) = \sum_{\tilde{a}_i \sim \{w,y,\theta,\phi\}} \|e_T\|c_{\tilde{a}_i} + \sum_{\tilde{a}_i \sim \{w,y,\theta,\phi\}} \|U\|c_{\tilde{a}_i}(w - c_U),
\]

where \( \tilde{a}_i \) is an OR of a triangle \( U \) of the partial model and \( \tilde{a}_v \) is an OR of a void patch attached to the triangle \( e_T \) along edge \( e \). The computational cost of satisfying the tolerance constraint is practically zero since nearly all of the work is done when projecting the ORs of the partial model to PS when forming \( P_s \).

ACKNOWLEDGMENTS

I wish to thank Ruzena Bajcsy for her support and for making this work possible. This work was funded by Carnegie Mellon University under subcontract 1-41480 to the University of Pennsylvania under MCC/Air Force Contract F33615-94-C-4400. I was with the MIT Department of Ocean Engineering when this paper was written.

REFERENCES


