Automatic 3D Modeling of Unstructured Scenes

Ph.D Thesis Proposal
Daniel Huber

The Robotics Institute
Carnegie Mellon University
Pittsburgh, Pennsylvania 15213

June 2000

Abstract

Many computer vision and robotics applications call for accurate three-dimensional (3D) models of real-world objects. Current 3D modeling techniques require significant manual assistance or make assumptions about the scene characteristics or data collection procedure. In this thesis work, we propose to fully automate the 3D modeling process without resorting to these restrictive assumptions. Given a set of unordered range images and no additional a priori information about the scene, our system will generate an accurate 3D reconstruction. Specifically, it is not necessary to know the relative pose between viewpoints or to indicate which views contain overlapping scene regions.

Our proposed automatic modeling system selects pairs of views that are likely to match and attempts to register them. The results are verified for consistency, but some incorrect matches may be locally undetectable and some correct matches may be missed. One of several discrete optimization techniques is employed to combine these potentially faulty pair-wise matches into a network of views called the model graph. Incorrect pair-wise matches are detected by the inconsistencies they produce elsewhere in the model graph, while missed matches are recovered by inferring new links in the graph between overlapping views. The overall model quality is improved by simultaneously registering all views before they are integrated together to form the final model. We demonstrate the utility of automatic modeling with an application called hand-held modeling, in which a 3D model is automatically created from an object held in a person's hand.

Table of Contents

1 Introduction

Computer vision researchers have long studied the problem of constructing three-dimensional (3D) computer models of real-world scenes and objects. With the advent of mass market 3D graphics cards, the demand for such models has skyrocketed. Corresponding advances in 3D sensors have led to relatively low cost range imaging devices that accurately measure the 3D structure of a scene from a single viewpoint. Generally, not every surface in a scene can be observed from a single range image, so data from multiple viewpoints must be combined in order to form a complete model. Although techniques for modeling from range images have advanced steadily over the past decade, current methods still require significant manual assistance or make assumptions about the scene characteristics or data collection procedure, limiting their applicability in many modeling applications. In this proposal, we present a general method to fully automate the 3D modeling process without resorting to these restrictive assumptions.

1.1 The 3D modeling problem

The 3D modeling process involves two main steps: registration, in which the 3D data sets (*views*) are all aligned in a common coordinate system; and integration, in which the registered views are combined into a single entity (figure 1). In this work, we concentrate on the registration phase because this is where the central automation issues lie. Three interrelated problems must be addressed in the registration phase: 1) determining which views overlap significantly (Views must contain overlapping scene regions in order to be registered); 2) determining the transform between each pair of overlapping views (*relative poses*); and 3) determining the position of all views in a global coordinate system (*absolute poses*).

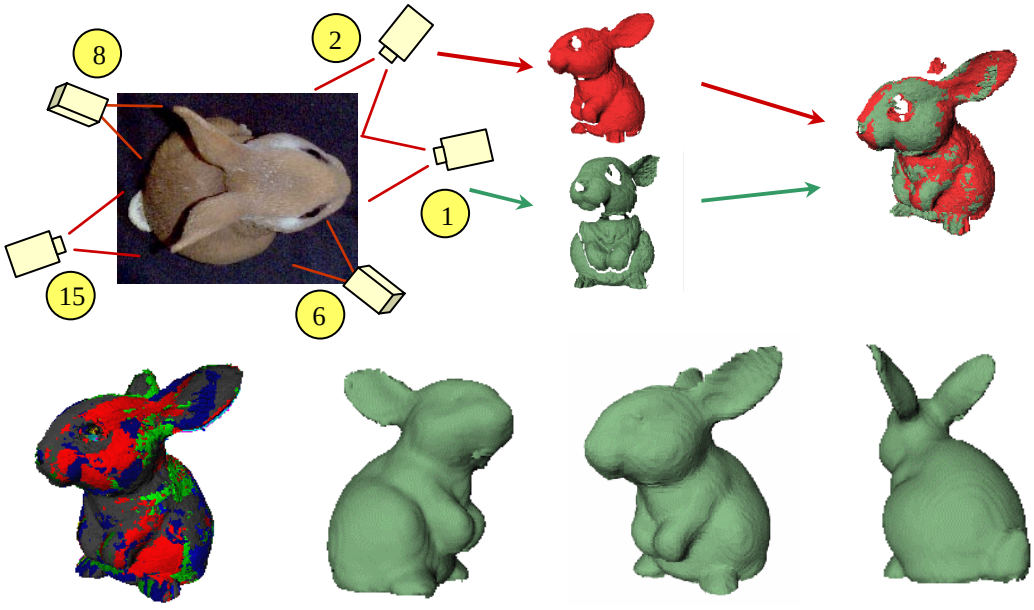


Figure 1: The 3D modeling process. Range images of a scene are obtained from several viewpoints (top left) and converted to triangular surface meshes (top center). Overlapping views are then registered with each other (top right), a process which is repeated many times. All of the registered views are transformed into a common global coordinate system (bottom left) and integrated together to form the final model (bottom center and right).

Existing systems solve these problems by mechanical methods, manual interaction, or by making simplifying assumptions about the scene. One mechanical approach is to mount the scanner on a robot equipped with an absolute positioning sensor. For example, Miller uses an autonomous helicopter with a differential global positioning system (DGPS) to construct terrain models [25]. For modeling smaller objects, absolute poses can be obtained by mounting the sensor on a robot arm [43] or by keeping the sensor fixed and moving the object on a calibrated platform [42]. Relative poses can be estimated by mounting the sensor on a robot equipped only with a relative positioning system such as wheel encoders or inertial sensors. Examples include Lu and Milios [24], El Hakim et al. [13], and Sequeira et al. [33], who modeled building interiors, and Shaffer, who mapped coal mines [34].

A common manual registration method is to specify corresponding feature points in pairs of range images, from which relative poses can be derived [27]. In some systems, corresponding feature points are automatically detected and then manually verified for correctness [13]. Alternately, the 3D data can be aligned directly through an interactive method [30]. In more advanced systems, a person indicates only which views to register, and system registers the views with no initial pose estimate [21][7]. With this approach, the user must manually verify the results.

Automatic 3D modeling aims to solve these registration issues without using mechanical pose-estimation methods and without requiring manual assistance. In our system, it is not necessary to know (or measure) the relative pose between viewpoints or to indicate which views contain overlapping scene regions. However, we will show that when this information is available, our system is able to take advantage of the additional knowledge.

One question that often arises is whether automatic 3D modeling is a useful goal to pursue. There are three primary reasons why automatic 3D modeling is important. First, it allows new applications, such as handheld modeling, that would be impossible otherwise (figure 2). With handheld modeling, the user holds an object before a desktop 3D scanner, obtaining range images from various directions without recording the object's pose or whether consecutive views overlap. Once the data is collected, the system automatically determines which views overlap, registers them, and integrates them into a final 3D model. Automatic modeling makes data collection almost trivially easy, an advantage that applies to all existing 3D modeling applications as well. A second argument for automatic modeling is that manually assisted modeling is tedious and time-con-

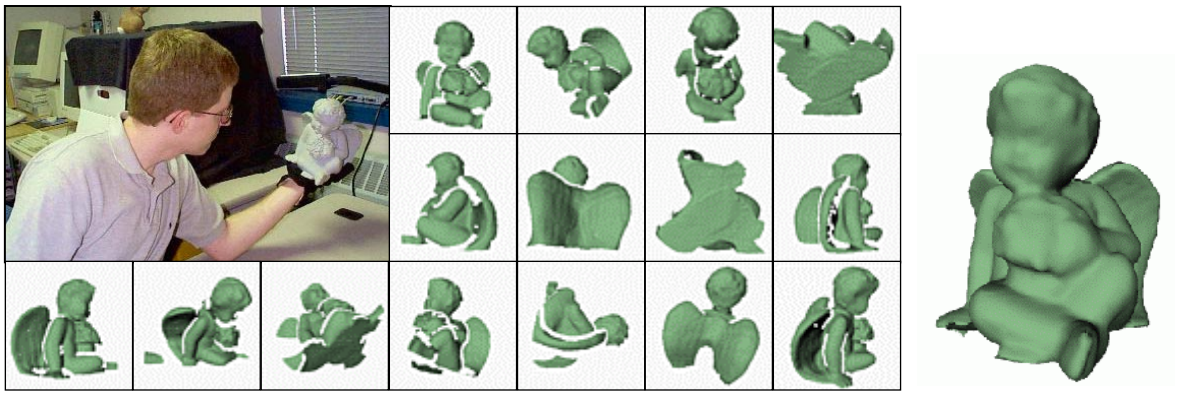


Figure 2: The handheld modeling application. A set of range images is obtained by holding an object before a 3D scanner (left). The automatic modeling system converts the images to surface meshes (center), determines which views overlap, registers them, and integrates them into the final model (right).

suming. Current data sets can contain hundreds of range images, and manually specifying correspondences takes many hours of work. Automatic modeling will allow us to tackle even larger data sets, limited only by computing power. Finally, automatic modeling offers an alternative to mechanical solutions, which add cost and complexity to a system while limiting its domain of applicability. For example, a system with GPS will not work for indoor modeling, and a calibrated turntable system cannot model buildings or terrain. Automatic modeling can handle all of these domains equally well.

1.2 Problem statement

Given an unordered set of range images of a scene, automatically and robustly construct a consistent 3D model under the following assumptions:

1. *The relative pose between view pairs may be unknown.*
2. *It is not known which view pairs contain overlapping scene regions.*
3. *The data may be noisy and may contain spurious views*[1].
4. *The scene may be unstructured. There is no guarantee that it will contain man-made objects or easily detectable features such as corners or lines.*
5. *The scene is static, containing only rigid, non-moving objects.*

In the event that a single model using all the images cannot be found, determine a set of partial models.

1.3 Proposed approach

The difficulties of automatic modeling stem from the three registration-phase issues mentioned previously: determining overlapping views, relative poses, and absolute poses. The goal of the registration phase is to find the absolute poses of all views. Excluding mechanically assisted systems that measure absolute poses directly, almost all current approaches begin with the assumption that approximate relative poses are given. From this, the overlapping views can be determined by examining the 3D data, and then the absolute poses can be found by simultaneously registering all overlapping views (multi-view registration). Johnson took the opposite approach, which is more difficult. He assumed that the overlapping views were given, and found relative poses using pair-wise registration [19][21]. In our case, neither the relative poses nor the overlapping views are known, and this makes the problem much harder.

Our approach to automatic modeling is to identify views which are likely to overlap, register them, verify the resulting matches, and construct a consistent model from the verified matches using discrete optimization techniques. Several key problems must be addressed. First, a method is needed to efficiently determine which views are expected to result in successful and stable registration. Second, pair-wise registration may find incorrect matches, so reliable verification procedures must be developed. Although local verification can eliminate many false matches, scenes can contain local symmetries and ambiguities that make it impossible to determine the correctness of a match based solely on the views being registered (figure 3). The third and most challenging difficulty is to construct a consistent model from these potentially faulty pair-wise matches. The key here is that the locally undetectable errors can be found through the constraints imposed by

1. Spurious views are data sets that contain substantial invalid data and should not be included in the final model.

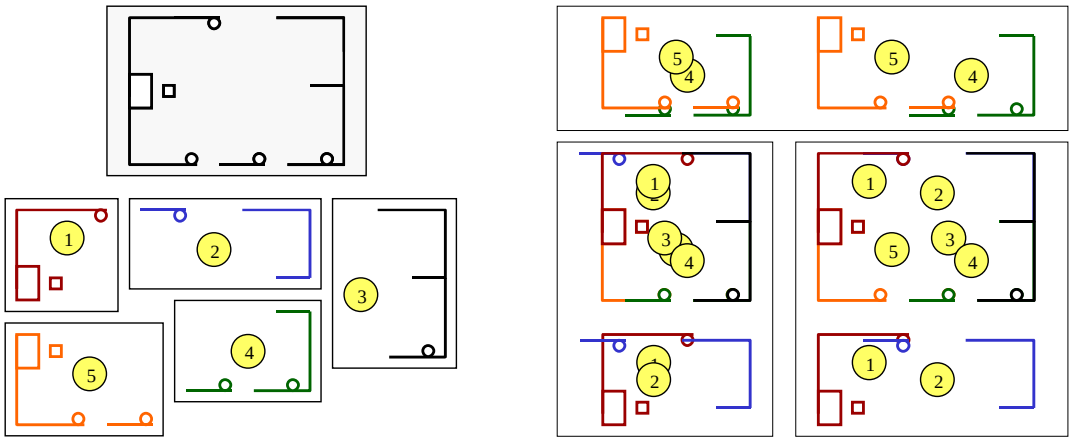


Figure 3: Example of a local ambiguity that can only be resolved globally. View from above of a hypothetical room (top left) and five scans of the room (bottom left). Registration of views 4 and 5 gives two equally likely matches (top right). If the incorrect match is used, views 1 and 2 are inconsistent (bottom center), but if the correct match is chosen, the entire model is consistent (bottom right).

the correct matches and from the inconsistencies that occur in the final model when incorrect matches are used. Finally, there is the problem of spurious views, which are views that contain invalid or insufficient data and should not be incorporated into the model. This problem can occur, for example, in handheld modeling when the object to be modeled is inadvertently outside the scanner's field of view. A modeling system should be robust to isolated spurious views.

Our solution to the automatic modeling problem is implemented in a framework consisting of two processes working in tandem – the *surface matching process* and the *model construction process* (figure 4). The surface matching process operates locally, selecting views that are likely to match, registering them, and performing local verification on the results. Verified pair-wise matches are added to a pool of candidates called the working set. The model construction process operates globally, combining matches selected from the working set into a network of views called the model graph. Matches missed by pair-wise registration can be recovered by inferring new links in the graph between overlapping views, a process we call topological inference. Inconsistent overlapping views (consistency conflicts) are evidence that the model hypothesis contains an incorrect

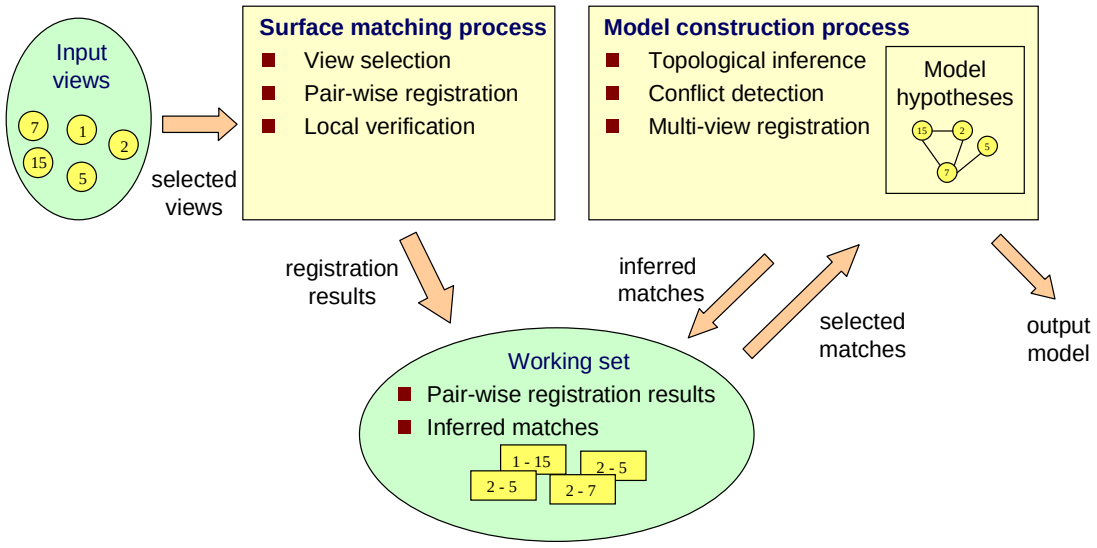


Figure 4: The automatic modeling framework. See text for details.

match, which must be removed. The overall error of the final model is reduced through multi-view registration.

We describe two classes of automatic modeling algorithms using this framework. First, we concentrate on the model construction process, effectively eliminating the view selection portion of the surface-matching process by registering all possible view pairs up front. Then, we consider the more difficult case in which the surface-matching process selectively registers views as needed, but before presenting these algorithms, it is necessary to elaborate on two concepts central to this framework: surface registration and the model graph

1.4 Surface registration algorithms

Surface registration algorithms play an important role in the modeling process. Their purpose is to align two or more surfaces represented in different coordinate systems, thereby determining the transform(s) between the surfaces. For rigid bodies, each transform can be represented by six parameters (e.g., three rotation angles and three translation distances). We call a transform between a pair of views a *relative pose* and a transform between a single view and the world coordinate system an *absolute pose*. The result of registration between two views (pair-wise registration) is a relative pose. When extending registration to more than two views, the solution could be specified in terms of relative poses between all the views, but such a representation does not ensure consistency among all the relative poses, a problem that will be discussed more in section 4. Instead, the solution to multi-view registration problems is represented in terms of each view's absolute pose.

Surface registration is generally formulated as an optimization problem involving the search for pose parameters that minimize a cost function which quantifies registration quality, such as the average squared distance between surfaces. We use the term *registration refinement* to refer to algorithms that begin with initial estimates of the unknown pose parameters and iteratively improve the registration until a minimum of the cost function is reached. Typically, these algorithms converge to a local minimum, so the initial pose estimates must be close to their true values. We use the term *uninformed registration* for algorithms that do not require initial pose estimates. Algorithms in this class search for the minimum of the objective function over the entire parameter space and therefore can succeed even when initial pose estimates are not known.

With these definitions in mind, the registration algorithms used in this research can be classified as follows:

- *uninformed pair-wise registration* – used to register two possibly overlapping views with no relative pose estimate (section 2.2)
- *pair-wise registration refinement* – used for refining the results of uninformed pair-wise registration (section 2.3) and for topological inference (section 5)
- *multi-view registration refinement* – used to ensure consistency in the relative poses between three or more views (section 4)

Throughout this paper, when it is obvious or irrelevant whether initial pose estimates are available, we will omit the "uninformed" and "refinement" qualifiers. In particular, we will refer to multi-view registration refinement simply as multi-view registration, since initial pose estimates are always required for this algorithm.

1.5 The model graph

The construction and topology of 3D models can be described in terms of a graph data structure called the *model graph*. A model graph, *G*, is a graph containing a node for each view of the scene. An edge in the model graph connects two nodes with overlapping views and is annotated with the relative pose of the two views[2] (figure 5). For non-overlapping views, the relative pose can be calculated by compounding the transforms along a path between the corresponding nodes in *G*. A connected model graph (i.e., all nodes are connected by some path) specifies a *complete model*, since every view can be transformed into a common world coordinate system. If, instead, *G* contains several connected components, each component is a *partial model*. A spanning tree of *G* is the minimum specification of a complete model. Additional edges will create cycles in *G*, which can lead to inconsistencies because compounding transforms along different paths between two views may give different results. The poses in a model graph are consistent if the relative pose of two views is independent of the path used for the calculation.

Using model graph terminology, the goal of automatic modeling is to recover the true model graph, G_{GT}, from an initial model graph, G_0, which contains no edges. *A priori* knowledge about the scene is easily encoded in G_0. For example, an initial relative pose estimate is represented by an edge in G_0 connecting the two views. If it is only known that the two views contain overlapping scene regions, the edge is added, but the relative pose is left unspecified. Transforming a model graph, *G*, from G_0 to G_{GT} involves adding, removing, and updating edges using the basic operations shown in figure 4. Uninformed pair-wise registration adds one or more edges between two nodes in *G*. Pair-wise registration refinement updates a single edge in *G*, while multi-view registration updates all the edges. Topological inference connects two non-adjacent nodes whose views overlap, creating a cycle in the graph. However, if the views contain conflicting surfaces, this indicates that an existing edge elsewhere in *G* should be removed.

1.6 Automatic modeling algorithms

Returning to the framework in figure 4, the performance of automatic modeling algorithms depends on the implementation details of the two processes: model construction and surface matching. First, we focus on the model construction process by simplifying the view-selection

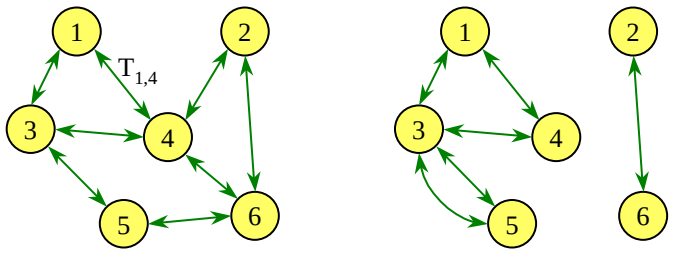


Figure 5: Model graphs contain a node for each input view and edges between overlapping views. Edges are annotated with relative pose of the two views (e.g., $T_{1,4}$). The left graph illustrates a complete model, while the model graph on the right shows two partial models. Multiple edges between nodes indicate alternative matches from pair-wise registration.

2. In practice, additional information, such as registration quality, can be associated with model graph edges.

component of the matching process. Then we consider algorithms that incorporate more sophisticated view-selection procedures.

The simplest view-selection algorithm is to try matching all possible view pairs and add all the results to the working set before the modeling process even begins. This brute-force tactic is reasonable for small numbers of views but does not trivially solve the automatic modeling problem because uninformed pair-wise registration can fail by finding false matches, missing true matches, or both. The challenge is to separate the false matches from the correct ones and to find the missing matches by alternate means such as topological inference. We will consider four algorithms based on this exhaustive matching approach:

1. *sequential removal* – start with all results in the model graph, then remove incorrect matches
2. *sequential addition* – start with an empty model graph and incrementally add matches
3. *hierarchical merging* – start by merging pairs of views into partial models, then repeatedly merge the partial models until a single model remains.
4. *randomized search* – start with a connected model graph and non-deterministically incrementally update the model.

For large numbers of views, the combinatorics of view-pairing makes exhaustive matching computationally intractable, and intelligent view-selection algorithms are needed. One approach is to exploit the information inherent in individual views to sort or partition them in a way that decreases the number of pairs that must be registered. For example, if the views can be partitioned into groups with similar surface shape, exhaustive registration within each group would create a small number of partial models, which could then be exhaustively matched and combined into a final model. The exhaustive matching modeling algorithms can be adapted to take advantage of these selective matching techniques.

After summarizing the related work, we describe each component of our automatic modeling framework: pair-wise registration (section 2), local verification (section 3), multi-view registration (section 4), and topological inference and global verification (section 5). Then we give details of the automatic modeling algorithms introduced in this section (section 6) and describe the experimental procedure that we will follow (section 7).

1.7 Related work

Section 1.1 already described the main approaches to 3D modeling from range images. Another class of successful modeling algorithms use intensity images rather than range images. These methods can be further subdivided into those that use a time sequence of images (structure from motion) and those that just use a set of photographs.

Structure from motion relies on the ability to locate and track stable features (such as corners) over a sequence of images. Given a sufficient number of tracked 2D features, the 3D location of the features and the camera poses can be recovered simultaneously. Tomasi and Kanade developed a batch method called factorization that assumes an orthographic camera model, while Pollefeys et al. and Beardsley et al. employ sequential algorithms that estimate the motion between pairs or triplets of images [41][28][1].

There are many different approaches to modeling from a set of images. Debevec's Facade system exemplifies a number of methods for modeling buildings [11]. With Facade, an approximate model is hand-constructed from primitives such as blocks and cones, and an optimization algorithm finds the parameters of the primitives that minimize the disparity between the projected edges in the model and the original images. At the other end of the size spectrum, shape from silhouette methods have been used to create 3D models of small objects by intersecting the viewing cones of a few registered photographs. Sullivan and Ponce fit triangular splines to the resulting volume to enforce a smoothness constraint [40]. Kutulakos and Seitz extend the volume intersection idea to interior points [22]. Their space carving algorithm begins with a solid volume and removes voxels that do not have the consistent appearance (i.e., color) in all images in which they are observed.

2 Pair-wise registration

2.1 Converting to a surface-based representation

Rather than working directly on range images, each view is first converted into a surface mesh composed of triangles. Meshes are a flexible representation of surfaces, and many tools and libraries specialize in manipulating and displaying them. The conversion process is straightforward. Pixels in the range image become vertices in the surface mesh, and triangles are formed by connecting each pixel with its four cardinal neighbors and two of its diagonal neighbors. The main challenges are the detection and removal of range shadows and the handling of very large data sets.

Range shadows, which are discontinuities that occur at occluding boundaries in the scene, are handled either by thresholding mesh edge lengths, or using a range shadow detector [18]. If range shadows are not removed, a phantom surface will be introduced into the mesh when pixels on both sides of the discontinuity are connected (figure 6).

The range images from high resolution 3D sensors give rise to large surface meshes that exceed the capabilities of our computing hardware. For example, a 6000 by 1500 pixel range image from our Z&F/Quantapoint scanner equates to a mesh with nearly twenty million faces. This problem is addressed by a combination of down-sampling of the range image and simplification of the resulting mesh. We use Garland's quadric algorithm for mesh simplification for its speed and because it inherently simplifies low-complexity regions while maintaining the detail of high-complexity regions with more faces (figure 7) [14][15]

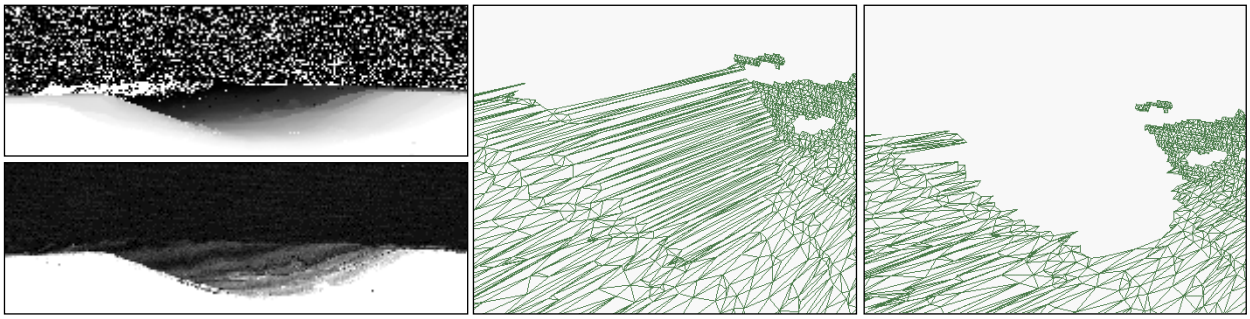


Figure 6: Range (left top) and reflectance (left bottom) images of terrain. The view shows a valley with a dirt road which runs between two hills and turns to the left in the distance. A wireframe view of part of the resulting mesh shows the surface before and after range shadow removal (center and right).

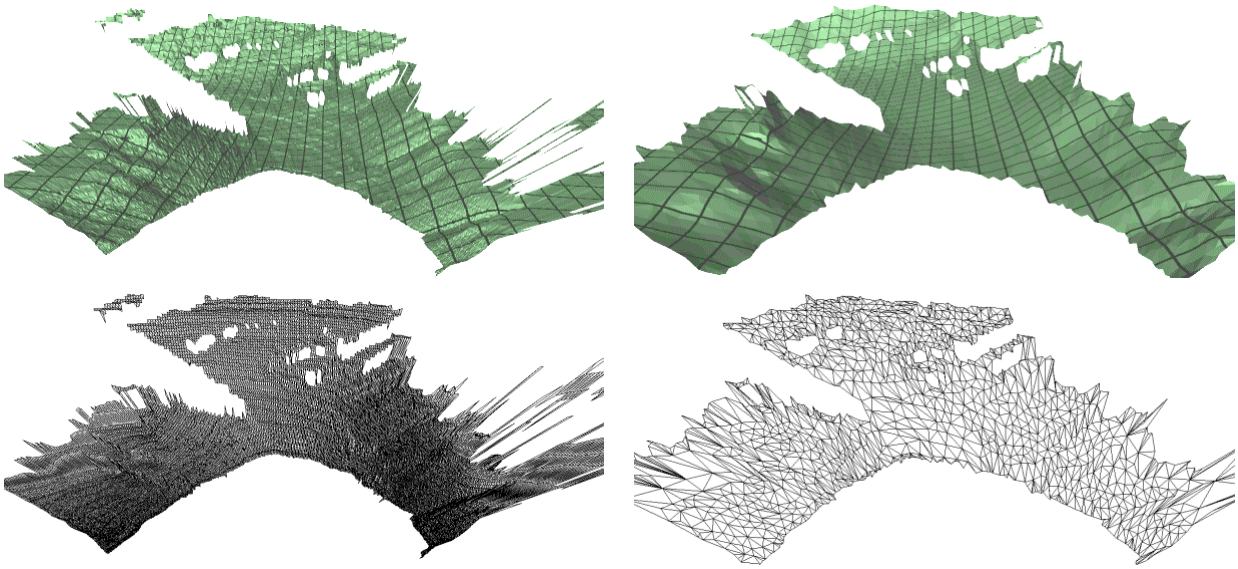


Figure 7: Mesh simplification. The original mesh shown with a grid texture map (left top) and in wireframe (left bottom). The mesh after simplification (right top and bottom).

2.2 Uninformed pair-wise registration.

Uninformed pair-wise registration finds the relative pose that aligns two views with no initial estimate of this transform. Our algorithm, based on Johnson's surface-matching engine, can be viewed as a black box which takes two surface meshes as input and outputs a ranked list of relative poses or no solution if the views could not be registered [19][21]. Inside the black box, the surface-matching engine searches for similarly shaped regions on two surfaces using point signatures. The signature associated with a given surface point encodes local surface properties, such as nearby shape, texture, or color. In this implementation, the surface matching engine uses a shape-based signature called spin images, but other representations are possible [10][38]. The algorithm depends on the fact that the same surface point in two different views will have a similar signature (i.e., point signatures are viewpoint invariant). Our implementation extends Johnson's original algorithm, making it independent of the mesh resolution and adding a number of speed enhancements [6][20].

Given two surface meshes, the uninformed registration algorithm begins by computing point signatures for a sampling of points on both surfaces. Each point signature from one surface is compared with those from the other surface, and the most similar pairs become candidate correspondences, some of which may be incorrect. The candidates serve as seeds in a clustering algorithm that finds groups of correspondences that could arise from a single rigid-body transform. The relative pose computed from each group that passes this geometric consistency test is a candidate match. Finally, the candidate matches are refined, verified, and ranked by a version of the iterative closest point (ICP) algorithm, and the top matches are provided as output.

No uninformed pair-wise registration algorithm is infallible, and this algorithm is no exception (figure 8). The algorithm outputs the best results under the assumption that the input views contain overlapping scene regions. If this assumption is violated, the algorithm may still find a reasonable result. Even if the views overlap, there is no guarantee that the transform that best aligns

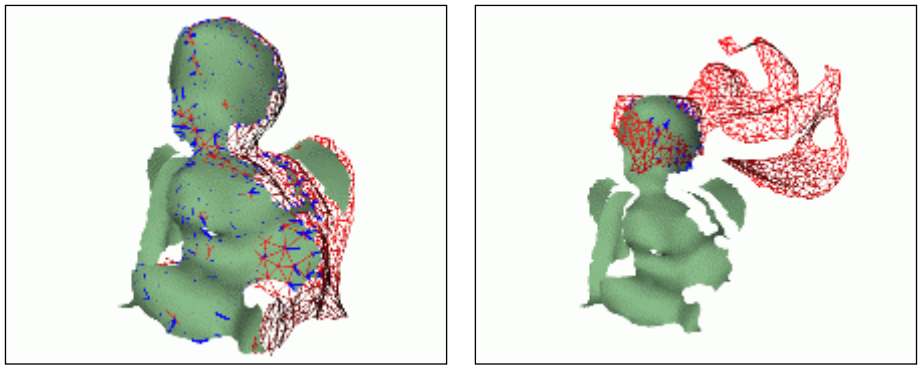


Figure 8: Uninformed pair-wise registration examples. A correct match between views 1 and 2 (left) and an incorrect match between views 1 and 9 (right). The surface matching engine reports the incorrect match because the overlapping region (the head) matches quite well.

the surfaces is the correct one. The local verification procedures described in section 3 can detect some additional registration errors, but the fundamental limitation is that pair-wise registration is a local operation. We address this limitation by using global consistency constraints to identify incorrect matches that cannot be detected locally.

Several other methods for uninformed pair-wise registration have been proposed, primarily for the purpose of object recognition. Stein and Medioni define the splash, which is an encoding of surface normals along a circular slice through the object at a fixed radius from the basis point [38]. The encoded splashes are used for object recognition in a surface matching algorithm similar to Johnson's surface matching engine. Chua and Jarvis also use the idea of a circular slice at a fixed radius, but instead of recording surface normals for points on the intersecting curve, they measure the distance to the tangent plane passing through the basis point [10]. Surfaces are matched by searching for three pairs of matching signatures. Chen et al. dispense with point signatures entirely and use a random sampling approach to search for three or more pairs of points on the two surfaces that are geometrically consistent under rigid transformation [7]. Higuchi et al. use a representation called the Spherical Attribute Image (SAI), which an encodes an object's surface curvature on the surface of a sphere [17]. Surfaces are matched by rotating the SAI's to maximize their similarity.

2.3 Pair-wise registration refinement

The results of uninformed pair-wise registration can be improved by following up with registration refinement, which aligns two surfaces under the condition that a good estimate of the relative pose is already known. We have implemented two algorithms for registration refinement. One minimizes the distance between pairs of points on the two surfaces, while the other minimizes the distance between points and tangent planes. The first algorithm, a version of the ICP algorithm extended to handle partially overlapping surfaces [4][44], capitalizes on the fact that, for relative poses near the solution, corresponding points on two surfaces can be approximated by closest points on the two surfaces. The objective function minimized by the algorithm is the sum of squared distances between closest points on two surfaces, S_i and S_j:

$$E = \sum_k \left\| \boldsymbol{R}\boldsymbol{p}_k + \boldsymbol{t} - \boldsymbol{q}_k \right\|^2 \tag{1}$$

$$\boldsymbol{q}_k = \underset{\boldsymbol{q} \in S_j}{\operatorname{argmin}} \left\| \boldsymbol{R}\boldsymbol{p}_k + \boldsymbol{t} - \boldsymbol{q} \right\| \tag{2}$$

where $\boldsymbol{R}$ and $\boldsymbol{t}$ are the rotation and translation components of the relative pose, $\boldsymbol{p}_k$ are the sampled points on S_i, and $\boldsymbol{q}_k$ are the corresponding closest points on S_j. Each iteration establishes correspondences between closest points on the two surfaces, computes the transformation that minimizes the objective function for these fixed correspondences, and compounds the relative pose estimate with this incremental transformation.

The main drawback to the ICP algorithm is slow convergence for certain surfaces. Often, surface registration passes through two phases. In the first phase, the residual is reduced quickly as the two surfaces snap together. Then, in the second phase, the reduction slows because the surfaces must slide relative to one another (figure 9). The point-to-point correspondences act to hold the surfaces in place, slowing or sometimes stopping the sliding motion altogether.

The second algorithm alleviates the slow convergence problem of ICP by effectively allowing correspondences to slide along the surface. In this algorithm, instead of minimizing point-to-point distances, we minimize point-to-surface distances, approximating the surface by its tangent plane. The implementation iterates over the same steps as the ICP algorithm except that the objective function (1) is replaced by:

$$E = \sum_k [(\boldsymbol{R}\hat{\boldsymbol{n}}_k) \cdot (\boldsymbol{R}\boldsymbol{p}_k + \boldsymbol{t} - \boldsymbol{q}_k)]^2 \qquad (3)$$

where $\hat{\boldsymbol{n}}_k$ is the unit normal at $\boldsymbol{p}_k$. Unlike ICP, where the transform that minimizes the objective function for fixed correspondences can be found in closed form, this algorithm requires non-linear optimization to find the transform at each iteration. Consequently, even though fewer iterations are necessary, each iteration requires more computation than with the ICP algorithm. Our algorithm is similar to that of Chen and Medioni [8].

3 Local verification

Local verification is the process of deciding whether two surfaces are correctly registered in terms of the quality and stability of the match between the surfaces. For a high-quality match, the surfaces should overlap significantly and be close together wherever they overlap, and for a stable match, the uncertainty of the relative pose should be small in all dimensions. Local verification checks can be applied directly to registration results or indirectly through compounded transforms

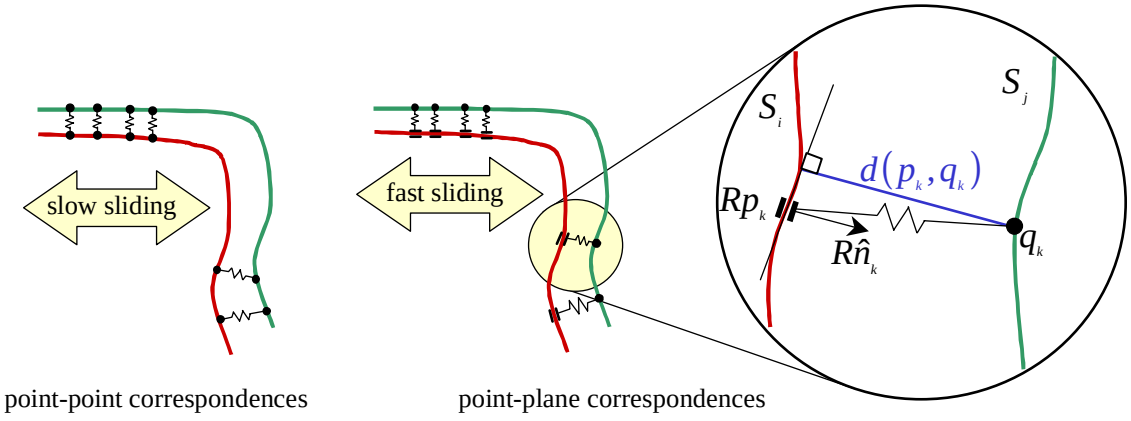


Figure 9: Minimizing closest point distances leads to slow convergence rates because the surfaces cannot slide (left), but minimizing point to plane distances does not have this limitation (center and right).

as part of the topological inference algorithm. Additionally, unthresholded versions of the quality and stability measures can be used during model construction for ranking matches. In this section, we describe two approaches for measuring registration quality (overlap distance measures and visibility consistency checks) and one for measuring registration stability (differential constraints).

3.1 Overlap distance

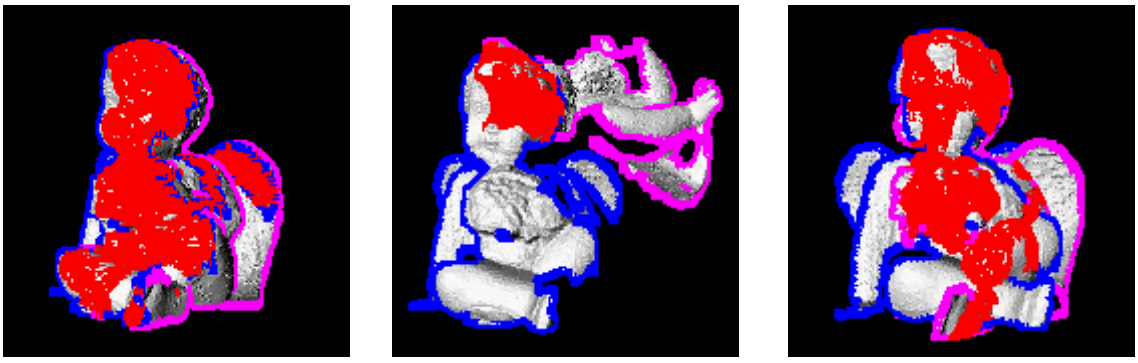
One way to judge the quality of registration between two surfaces is to directly measure the distance between the surfaces in overlapping regions (figure 10). For this measure to be meaningful, the surfaces should overlap substantially, so the amount of overlap is considered as well. We use the following definition of overlap:

> A point, p, on surface S_1 overlaps surface S_2 if 1) the point, q, on S_2 closest to p is an interior (non-boundary) point of S_2; 2) the angle between the surface normals at p and q is less than a threshold, T; and 3) the Euclidean distance, d, between p and q is less than a threshold, D.

This definition is relatively insensitive to the choice of thresholds since most non-overlapping points fail the first criterion. One way to choose the thresholds is based on statistics of the distribution of point distances and normal differences for hand-labeled overlapping points on correctly registered training sets.

To compute the amount of overlap for surface meshes, we iterate over the vertices of the first mesh, computing whether each point overlaps the second mesh according to the definition above. Then we sum the surface area of all the faces which contain overlapping vertices. Faces that straddle the boundary of the overlapping area will have only one or two overlapping vertices and therefore contribute 1/3 or 2/3 of their area respectively. Alternately, the exact intersection of the face and the boundary can be computed, but such an approach gives similar results and requires much more computation.

The computation of overlap distance follows essentially the same algorithm except, in this case, we compute the overlap distance by averaging the distance at the vertices of each face and



Views 1 and 2: 37%, 1.9mm | Views 1 and 9: 12%, 2.8mm | Views 1 and 12: 39%, 4.7mm

Figure 10: Overlap measurement examples for one correct match (left) and two incorrect matches (center and right). The overlapping region is highlighted in red, and the captions indicate the overlap amount and distance. Notice the overlap distance for views 1 and 9 is relatively small.

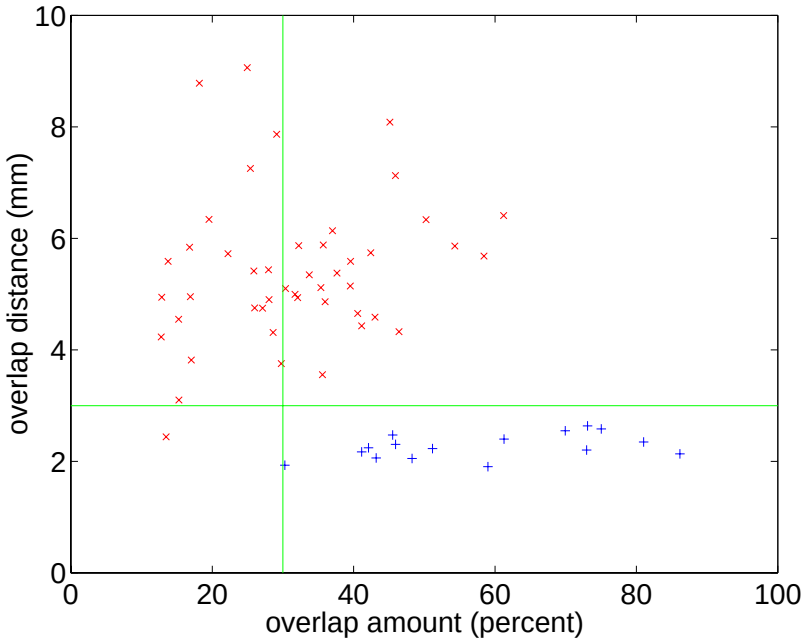


Figure 11: Using overlap measures to verify matches. The graph shows overlap amount and distance for the matches from all pairs of views the angel data set. Successful matches are labeled '+' and incorrect matches are labeled 'x'.

weighting by the face area. As with overlap estimation, border faces contain one or two overlapping vertices and are weighted by 1/3 or 2/3 of the face area respectively. Summing the area-weighted overlap distance for all faces and dividing by the total overlap area gives the average overlap distance.

By considering these two measures – amount of overlap and average overlap distance – as features, a classifier can be created to decide whether the two surfaces are correctly registered. The current implementation uses a linear decision boundary with manually specified parameters, but more complex classifiers could be implemented if needed and the parameters could be learned from hand-labeled training data. Figure 11 shows the classification of the results of pair-wise registration between the fifteen views shown in figure 2.

3.2 Visibility consistency

If the surfaces are derived from images, we can derive a more principled verification measure by looking at the consistency of the two surfaces from the perspective of one of the sensors. For example, consider the surfaces in figure 12 viewed from the sensor position C_1. For a correct registration, the two surfaces are consistent wherever they overlap. For an incorrect registration, two types of inconsistencies can arise. A *free space violation* occurs when a region of S_2 blocks the visibility of S_1 from C_1, while an *occupied space violation* occurs when a region of S_2 is not imaged by C_1, even though it should be. These inconsistencies serve as the basis for a verification procedure that can detect mis-registration in cases where the overlap distance tests will fail. Furthermore, the same procedure can be used to check for inconsistencies from the perspective of C_2.

Visibility tests have been used previously in other 3D vision contexts, including hypothesis verification, surface registration, and multi-view integration. Bolles and Horaud use the difference

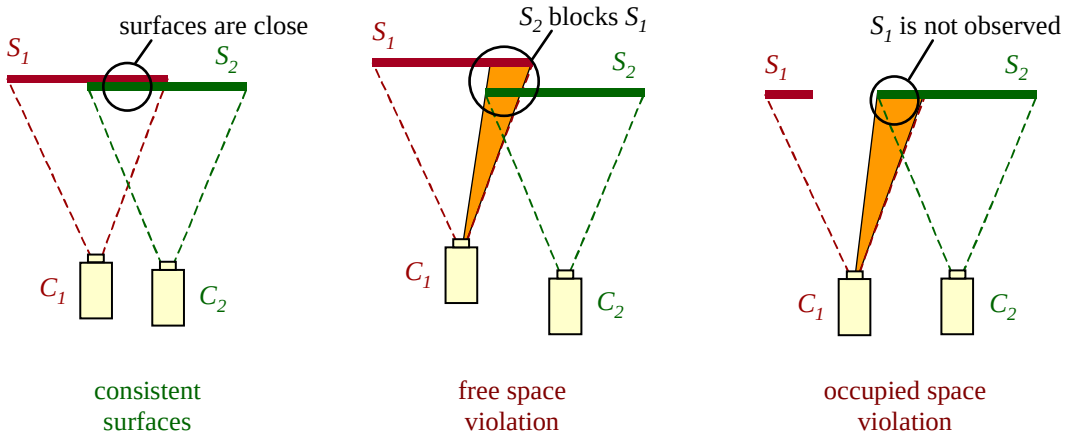


Figure 12: Visibility consistency verification. For correct registration, the two surfaces are consistent when viewed from the perspective of sensor C_1 (left), but for an incorrect registration, free space violations occur wherever surface S_2 is significantly closer than S_1, and occupied space violations occur wherever S_2 is not seen by C_1 even though it should be detected (right).

between predicted and measured range to verify hypothesized model poses in 3DPO, an object recognition system [5]. The system classifies range data into one of three categories: positive evidence - the measured and predicted ranges are approximately equal; negative evidence - the predicted range is significantly shorter than the measured range; and neutral evidence - the predicted range is significantly longer (since other objects in the scene may be occluding the target). Eggert et. al. use visibility tests to speed up convergence of their pair-wise registration refinement algorithm [12]. Not only do they explicitly detect and eliminate occluded points from being matched during the mapping of corresponding points, but points on the occluding side of step discontinuities are marked as “occluders” and mapped to the nearest occluded point on the other surface. Soucy and Laurendeau define the Spatial Neighborhood Test for integrating registered range images [37]. Points from one range image are projected into the other, and tested for closeness based on the assumption of a constant error bound. Robert and Minaud use a similar test on depth differences between views to eliminate outliers when integrating range-images from stereo cameras [31].

3.2.1 Measuring free space violations

Free space violations can be detected by projecting a ray from the origin of C_1 through a point, p_1, on S_1. If the ray passes through S_2 at a point, p_2, which is significantly closer to C_1 than p_1, then p_2 is an inconsistent point. We must test whether the point is *significantly* closer because even for correctly registered surfaces, the two points will not have exactly the same range. These points can be excluded with a suitably chosen threshold. However a better test of significance is to use an error model for the data. Range data errors can be approximated by a Gaussian distribution in range and a two-dimensional (2D) Gaussian distribution perpendicular to the range. With this model, we can use a statistical test to estimate the probability that the two points are samples of the same surface. Additional improvements may be possible by incorporating the difference in surface normal estimates for the two points.

The free space violation detection algorithm is straightforward and can be implemented efficiently using z-buffers:

1. Transform S_2 into the coordinate system of C_1
2. Project the faces of S_1 and S_2 into separate z-buffers.
3. For each pixel that is occupied in both z-buffers,
   3a. Compute the distances, d_1 and d_2, from the origin of C_1 to S_1 and S_2 respectively
   3b. If d_2 is significantly less than d_1 (according to the significance test described above), mark the pixel as a free space violation.

3.2.2 Measuring occupied space violations

Occupied space violations are more complicated to detect than free space violations. Recall that an occupied space violation occurs when the sensor fails to see a surface that should be seen according to the data from another view. A number of reasons can account for this sensing failure. The surface may face away from the sensor, may be occluded by another surface, or may be out of sensing range. Depending on the sensor, certain surface materials and geometries produce low-quality measurements that may be removed during the conversion to a surface mesh. We can account for many of these effects with an appropriate imaging model, which estimates the probability of observing a surface under different conditions such as distance to the surface and viewing angle (the difference between the surface normal and sensor viewing direction).

The occupied space violation detection algorithm can be implemented using the same z-buffers computed in the free space violation detection algorithm:

1. Transform S_2 into the coordinate system of C_1.
2. Project the faces of S_1 and S_2 into separate z-buffers.
3. For each pixel, k, that is occupied in the z-buffer for S_2 only, use the imaging model with parameters $\boldsymbol{a}(p_{2k})$ from S_2, compute $\mathrm{P_d}(\boldsymbol{a}(p_{2k}))$, the probability of detecting the point, p_{2k}, from the perspective of C_1.

One way to combine the individual probabilities is to assume each measurement is conditionally independent given the underlying scene. Then the probability of failing to detect all the missing data is:

$$P_{\bar{d}} = \prod_k (1 - P_d(a(p_{2k}))) \tag{4}$$

3.3 Differential constraints

The shape of the surfaces to be registered affects the uncertainty of the resulting registration. For example, after successfully registering two planes, all translations along the plane and rotations about the surface normal have no effect on the distance between the surfaces. We analyze the constraints imposed by surface geometry by considering the effect of small changes in relative pose on the distance between overlapping surfaces. The differential transform that induces the smallest change in the overlap distance corresponds to the most unconstrained motion. This idea is based on the work of Simon, who used constraint analysis to select points on a surface to maximize expected registration accuracy [35].

Although a full derivation of the geometric constraint measure would be too lengthy, some intuition can be gained by considering registration as a least squares optimization of the closest point distance error metric. Near the minimum, the cost function can be approximated by a six-dimensional paraboloid with curvature defined by the Hessian of the distance function [29]. Principal component analysis on the Hessian gives a set of eigenvectors and eigenvalues. The eigenvector corresponding to the smallest eigenvalue represents the least constraining differential transformation. Simon lists several measures that summarize the degree of constraint based on the relative magnitudes of the eigenvalues. These measures can be used to estimate the reliability of a registration result. Differential constraint analysis may prove to be useful for view selection as well. For example, the constraints imposed by registering a surface with itself can be used to predict the reliability of future registration with other surfaces.

4 Multi-view registration

Multi-view registration extends pair-wise registration refinement to work with more than two views, aligning them all simultaneously. The benefits of multi-view registration can be seen during in the modeling process when we construct model hypotheses from pair-wise matches. Small errors in relative poses accumulate whenever the transforms are concatenated, but when the model graph contains cycles, the alternate paths between views provide constraints that can reduce the accumulated error. For partial models, this reduction increases the chance of successful topological inference, since the views involved will be better aligned. In the final model, the effect of accumulated error is an obvious gap or seam in the model surface. Multi-view registration enforces consistency among the relative poses, ensuring that the relative pose between any two views is independent of the path over which the calculation is performed and distributing the accumulated error over the entire model.

The two registration algorithms described in this section are the multi-view analogs of the two pair-wise registration refinement algorithms in section 2.3, and both reduce to the pair-wise versions when registering only two views. As with the pair-wise refinement algorithms, a good initial estimate of the pose parameters is required. The consistency of relative poses is ensured by working with absolute poses of each view. One view is selected as the base view, and its absolute pose is set to the identity transform and held constant throughout the process. The initial absolute poses of the remaining views are computed by concatenating relative poses with respect to this base view. Before presenting details of the two multi-view algorithms, we describe hierarchical registration, an alternative approach used by Johnson [19].

With hierarchical registration, surfaces are registered in a pyramid fashion, alternating between registration and integration. At each level of the pyramid, two or more pairs of surfaces are pair-wise registered and integrated to form a single surface at the next higher level, and the process is repeated until all surfaces are combined. As a final step, all the original views are integrated using improved relative poses derived from the registration pyramid. The disadvantage of this method is that the integration at each level is a forced commitment to the relative pose parameters between those surfaces. If, later in the process, another surface shows that the original registration was incorrect, it is too late to change. As a result, even though hierarchical registration tends to smooth the visual effects of accumulated error, it does not take advantage of all the available overlapping surface constraints. The multi-view registration algorithms do not suffer from this early commitment problem because the views are all registered simultaneously.

The first multi-view registration algorithm extends the ICP algorithm. At each iteration, the squared distance of closest points on all overlapping surfaces is minimized:

$$E = \sum_i \sum_{j \neq i} \sum_k \left\| \boldsymbol{R}_i \boldsymbol{p}_{ijk} + \boldsymbol{t}_i - \boldsymbol{R}_j \boldsymbol{q}_{ijk} - \boldsymbol{t}_j \right\|^2 \tag{5}$$

$$\boldsymbol{q}_{ijk} = \underset{\boldsymbol{q} \in S_j}{\operatorname{argmin}} \left\| \boldsymbol{R}_i \boldsymbol{p}_{ijk} + \boldsymbol{t}_i - \boldsymbol{R}_j \boldsymbol{q} - \boldsymbol{t}_j \right\| \tag{6}$$

where $\boldsymbol{R}_i$ and $\boldsymbol{t}_i$ ($\boldsymbol{R}_j$ and $\boldsymbol{t}_j$) are the rotation and translation components of the absolute pose of view i (view j), $\boldsymbol{p}_{ijk}$ are the sampled points on the region of S_i that overlaps S_j, and $\boldsymbol{q}_{ijk}$ are the corresponding closest points on S_j.

Like ICP, each iteration of the algorithm involves calculating closest points followed by minimization of the objective function using those fixed correspondences. However, in this case, the minimization cannot be solved in closed form, so an inner loop is necessary. Our implementation, based on that of Benjemaa and Schmitt, operates efficiently by summarizing the closest point correspondences in a matrix that is only computed once per outer iteration, reducing the computation of the inner minimization loop [2]. Unfortunately, the algorithm as a whole suffers from the same slow convergence problems as ICP, only more so, because the correspondences established between many different overlapping surfaces form a tight web that often entirely prevents the surfaces from sliding. As with pair-wise registration, the multi-view solution to this problem is to minimize point to tangent plane distances instead, replacing (5) with:

$$E = \sum_i \sum_{j \neq i} \sum_k \left[(\boldsymbol{R}_i \hat{\boldsymbol{n}}_{ijk}) \cdot (\boldsymbol{R}_i \boldsymbol{p}_{ijk} + \boldsymbol{t}_i - \boldsymbol{R}_j \boldsymbol{q}_{ijk} - \boldsymbol{t}_j) \right]^2 \tag{7}$$

where $\hat{\boldsymbol{n}}_{ijk}$ is the unit normal at $\boldsymbol{p}_{ijk}$.

Unlike the multi-view ICP algorithm, this objective function cannot be manipulated to avoid iterating over all correspondences, so a slower inner loop minimization is required. The current implementation uses the Levenburg-Marquardt algorithm for this non-linear optimization. Multi-view registration using point-plane correspondences was originally proposed by Neugebauer [27].

Figure 13 shows a cross-section through several versions of a merged model. The position of the cross-section is synchronized so that the same slice is shown for all algorithms. Both multi-view algorithms perform better than the hierarchical method, but the point-plane algorithm is visually better.

A number of other multi-view registration algorithms have been developed. Bergevin et al. repeatedly perform pair-wise registration on pairs of overlapping views using a modified version of Chen and Medioni's algorithm [3]. Pulli uses a similar approach based on the ICP algorithm [30]. Lu and Milios derive uncertainty measures (covariance matrices) for the relative poses and solve for a consistent set of relative poses that minimize the total Mahalanobis distance between the two pose vectors for all overlapping view pairs [24]. Stoddart and Hilton use the analogy of a mechanical system in which corresponding points are attached by springs to derive a set of force-based incremental motion equations that are equivalent to gradient descent [39]. Eggert et al. also use a mechanical system analogy, but they include an inertia term and also update the correspondences over time [12]. Goldberger uses a model-based approach based on the EM algorithm [16]. At each iteration, every view is registered to the current model and then a new maximum likelihood model is created from the updated views.

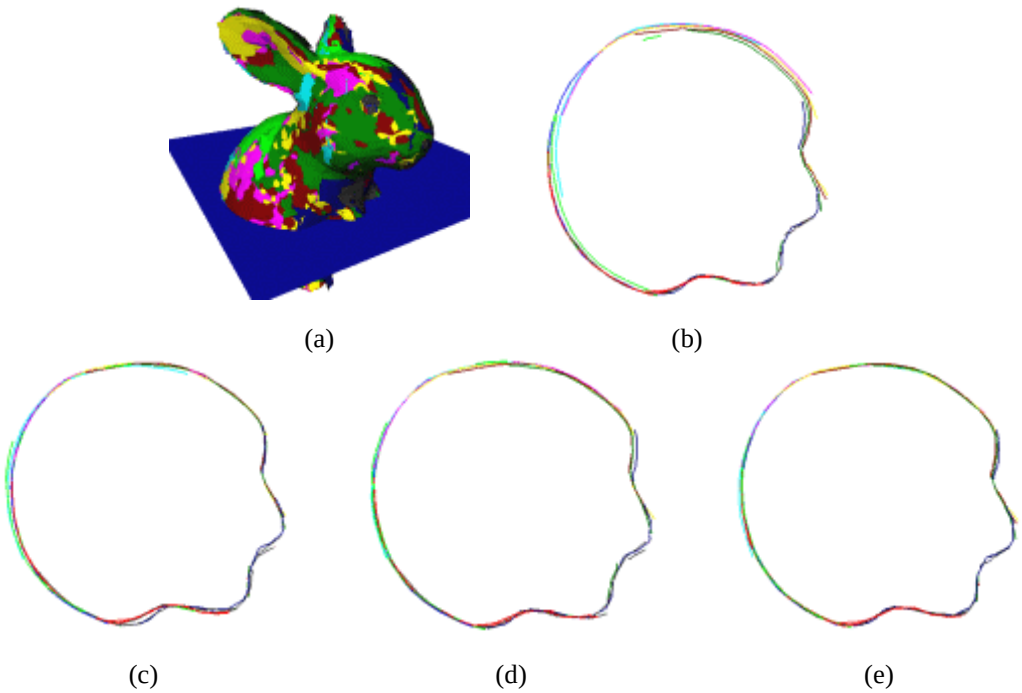


Figure 13: Comparison of multi-view registration algorithms. a) a cross section through the model (all slices are synchronized); b) pair-wise registration only; c) hierarchical registration; d) point-point multi-view registration; and e) point-plane multi-view registration.

5 Global verification

In a correctly constructed model, all of the views should be consistent with one another. Just as local verification checks the consistency of pair-wise matches, global verification checks the consistency of an entire model. In this section, we discuss two closely related topics: topological inference and conflict detection.

5.1 Topological inference

When constructing a model, some model graph edges simply cannot be found using uninformed pair-wise registration, either because the surfaces do not overlap sufficiently or because the surface shape is not unique enough to register. Topological inference addresses this problem by actively detecting overlapping surfaces based on the model topology and adding the inferred links to the model graph. Furthermore, uninformed registration is a more computationally intensive operation, making topological inference the logical choice in situations where both operations could be applied. The term topological inference was inspired by Sawhney et al., who used similar ideas in two dimensions to infer relations in a sequence of images for improved mosaicing [32].

Topological inference can be attempted between any two views that are connected in the model graph, but only when the surfaces overlap is there a chance of success. A simple, coarse test for possible overlap is to form a 3D bounding box around each surface and check for their intersection in a common coordinate system. The local verification procedures in section 3 provide a more precise secondary check. If the verification is successful, the corresponding edge can be

added to the model graph, and inferred edges with sufficient overlap can be improved with pair-wise registration refinement if necessary.

Topological inference will not work if the uncertainty in the relative pose is too large. For example, if two views are connected only by a long sequential path, the cumulative uncertainty along that path may be so large that the bounding boxes no longer intersect even though the views contain overlapping scene regions. In this case, we could use an estimate of the uncertainty between views to compute the likelihood that the views should overlap. These uncertainty estimates could be generated by propagating pair-wise registration uncertainty through the network [36], by considering all pair-wise uncertainty estimates simultaneously [24], or possibly using statistical parameter estimation techniques such as bootstrap [9]. Since such large uncertainties have not been observed in practice, this line of research will only be pursued if the need arises.

5.2 Conflict detection

What happens if the visibility consistency check in topological inference fails? In that case, the two views are in conflict, which indicates that an incorrect edge lies somewhere in the model graph. The conflict does not tell us which edge is incorrect; the only constraint is that it must lie in one of the paths connecting the two views in the graph.

Conflicts can occur between views even if the bounding boxes of their surfaces do not intersect. A coarse test for the possibility of conflicts must also take into account the free space between the surface and the sensor. This can be accomplished by including the sensor's viewing frustum in the bounding box intersection test. A correct model should be free of conflicts, but the method for removing detected conflicts depends on the automatic modeling algorithm and will be deferred to the next section.

6 Automatic modeling algorithms

Now that we have described the component tools – pair-wise registration, local verification, multi-view registration, and global verification – we turn our attention to the full problem of automatic modeling. As discussed in the introduction, our framework consists of two processes: the surface matching process, which selects pairs of views, registers them, and locally verifies the resulting matches; and the model construction process, which combines the pair-wise matches into a consistent model. Many of the pair-wise matches will be incorrect, and it is possible that even the best match between two views is not the correct one. A model will be corrupted if even a single incorrect match is included in its model graph.

In the most general sense, automatic modeling is an optimization problem. We search for the most likely model given the data. However, this problem is unique in that it lies in both the continuous and the discrete domains. In the continuous sense, the optimization is a search for the absolute pose of each view, which is a search over the continuous space of pose parameters of all views. On the other hand, using the model graph approach, the optimization is discrete, based on the set of distinct relative poses given by pair-wise registration. The automatic modeling problem cannot be solved solely within the continuous or discrete domains. Instead, we must work with both. While the low-level operations (e.g., pair-wise registration refinement) optimize in the continuous domain, we approach model construction primarily from the perspective of discrete optimization. Unlike most discrete optimization problems, the possible choices are not all known in

advance. View selection implies that many pair-wise registrations will not be attempted, and even if we exhaustively match all pairs, some correct matches may be missed. Additionally, topological inference introduces new choices and constraints that depend on specific model graph configurations.

Much of the work to be done in this thesis research centers on the model construction algorithms. We can, however, present an outline of some of the algorithms we are considering. We begin by studying exhaustive registration algorithms and then extend the ideas to incorporate view selection. The performance of these algorithms will be compared using objective criteria including topological accuracy of the model graph, global model error, computational complexity, and robustness. Additional details of the experimental process will be presented in section 7.

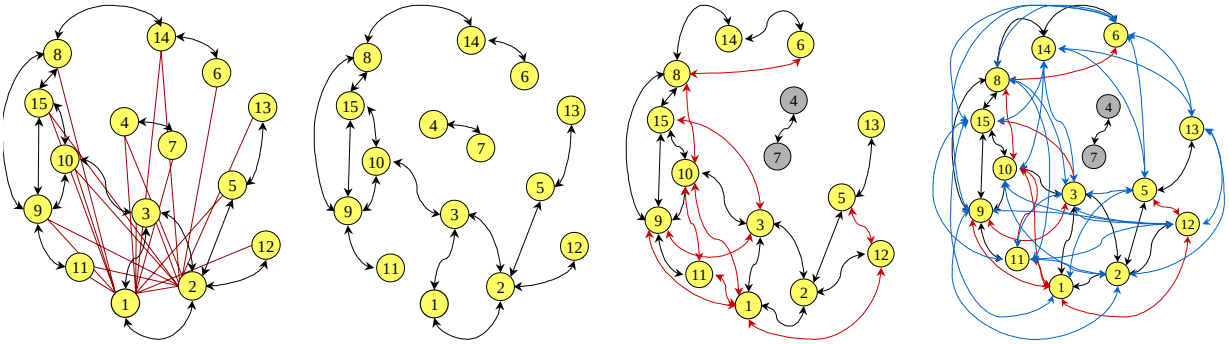
6.1 Exhaustive registration algorithms

With exhaustive registration algorithms, we first perform uninformed pair-wise registration of all possible view pairings. Since pair-wise registration can generate false matches and miss correct matches, the modeling problem is primarily a matter of separating the false matches from the correct ones and determining the missing matches by topological inference and conflict detection. Initially, we will consider four algorithms based on exhaustive registration:

- sequential removal
- sequential addition
- hierarchical merging
- randomized search

6.1.1 Sequential removal

This algorithm begins with a model graph containing all the matches from exhaustive registration and removes edges until all conflicts have been eliminated. Local verification can be used to sort the matches based on registration quality, and the worst edges can be removed until global verification determines that the remaining edges are consistent. Since the penalty for failing to detect an incorrect match is significant, it is better to remove some correct matches as long as incorrect ones are removed too. This is allowable, since topological inference can re-establish the edges



(a) (b) (c) (d)

Figure 14: Progression of the model graph for the angel model (sequential removal algorithm). a) The initial model graph with correct matches shown with arrows and incorrect matches shown without arrows (for clarity, only incorrect matches involving views 1 and 2 are shown). b) After removal of the bad matches using the overlap measurements, the model graph has two components. c) By applying topological inference with a threshold of 30% overlap, several new links are discovered. d) Repeating multi-view registration and topological inference with successively lower overlap thresholds gives the final model graph.

later. However, if removing an edge separates the model into two partial models, the partial models cannot be recombined. A similar problem occurs if the initial model graph is not connected, which happens when one or more views does not match any other view.

A simple version of this approach was implemented as a demonstration of handheld modeling. Fifteen views of an angel statue were obtained by holding the object before a 3D scanner (figure 2). Exhaustive registration led to the initial model graph of figure 14a. The overlap measure classifier was used to select edges to be removed (figure 11). The remaining model graph contains two partial models (figure 14b). Views 4 and 7 registered only with each other, so they could not be connected with the main model. Next, we performed multi-view registration on the main model followed by topological inference (figure 14c). Repeating multi-view registration and topological inference with successively lower overlap thresholds (figure 14d) produced the final model in figure 15.

6.1.2 Sequential addition

Instead of pruning bad matches from the model graph, the opposite approach is to begin with a seed view and incrementally add additional views, building the entire model from this single base. This could be accomplished with a simple informed search algorithm such as best first search. To increase the chance of success, several models could be constructed starting from different seed views. The main danger with the sequential addition algorithm is that if an incorrect match is added to the model hypothesis at any time, the final model will be corrupted. This error may not be detected until several more views are added, limiting the effectiveness of backtracking, since we do not know how far to backtrack. The likelihood of adding incorrect matches could be reduced by preferentially selecting matches that create cycles in the model graph. These cycles, provide multiple paths of supporting evidence for a correct match, or looking at it another way, provide multiple chances for detecting an incorrect match. Given the consistency constraints imposed by a model graph with cycles, it is possible that sequential addition can be posed as a constraint satisfaction problem.

6.1.3 Hierarchical merging

One disadvantage of the previous two algorithms is that some views may not match well with any other views, which leads to multiple partial models even when a single model should be obtained.

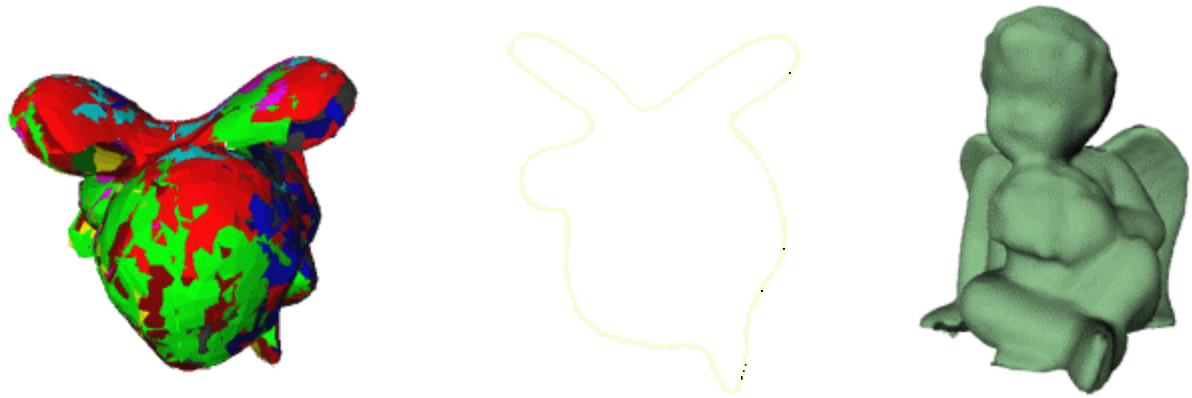


Figure 15: Three views of the final angel model: top view of merged data sets (left); a horizontal slice through the merged model (center); and a rendering of the completed model after integration (right).

For example, in figure 14b, the two partial models could not be joined by the sequential removal algorithm. However, if we were to attempt to register the two partial models, the complete model would be found. This suggests an algorithm in which we merge partial models. We call this approach hierarchical merging.

With hierarchical merging, all of the consistent pair-wise matches are placed in a pool of partial models. The best partial models are joined together to form larger partial models, which are added to the partial model pool. This process is repeated until no more partial models can be generated. The partial models can be joined using the existing pair-wise match information or by treating the partial models as "mega-views" and performing pair-wise registration between mega-views. The pool of partial models is, in effect, storing multiple model hypotheses.

6.1.4 Randomized search

Non-deterministic methods can avoid local minima and often give reasonable results with much less computation than deterministic searching. One approach is to use robust methods such as the random sample consensus (RANSAC) algorithm. RANSAC involves generating a possible solution using a randomly selected minimal subset of the data and evaluating the solution using the remaining data. In our case, the minimum specification of a complete model is a spanning tree of the model graph. The global verification procedures can then be applied to determine the quality of the hypothesized model. This process is repeated many times, and the best model is chosen as the output.

Simulated annealing is another non-deterministic approach we are considering. For this, we must derive an evaluation function with which we can compare the quality of different solutions. Such a function should minimize the number of partial models while maximizing the global consistency of each partial model. The algorithm would proceed as follows: The model graph is initialized with matches from exhaustive registration with the constraint that only one edge is allowed between each pair of views. At each iteration, a new edge can be added or an existing one removed. If the evaluation function is improved by the change, it is kept, otherwise, with probability P, the change is rejected. The probability of rejecting a change for the worse is increased over time according to the annealing schedule. In theory, simulated annealing is guaranteed to find a global maximum of the evaluation function.

6.2 Selective registration algorithms

To scale automatic modeling to large numbers of views, we must be selective about which views to attempt to register. One way to reduce the number of pairs that must be registered is to use the information inherent in each view to sort the views based on likelihood of a successful match or partition them into groups that are likely to match.

We will briefly outline a number of ideas for sorting and partitioning views. Then, we will suggest how the exhaustive registration algorithms could be adapted to incorporate these methods.

- *Method 1* – Partition the views into groups, either blindly or based on a shape similarity metric, then perform exhaustive registration within each group. Partitioning based on the view ordering may give good results because the sequential views are more likely to overlap in most data collection scenarios.

- *Method 2* – Given one view, perform pair-wise registration with the other views only until a sufficiently good result is found. Each view typically matches with several other views, it is not necessary to find all matches up front. By stopping after the first match is found, the number of registrations for each view is reduced by a constant factor.
- *Method 3* – Sort the views based on the degree of geometric constraint then apply method 2. It is a good idea to register the most constraining views first because the least constraining views are the ones for which registration is most likely to fail.
- *Method 4* – For each view, sort the remaining views based on pair-wise likelihood of registration. This can be accomplished efficiently using spin-image point signatures. These signatures can be viewed as points in a high dimensional space, in which the distance between two points is a metric of shape similarity. Point signatures from all views can be represented in this space. Then, for each view, an ordering of the remaining views can be determined based on the number of points close to the points for the given view.

Of the four types of exhaustive registration algorithms, all but the sequential removal algorithm could be adapted to handle the partial information supplied by selective registration. With the sequential addition algorithm, views could be selected based on their likelihood of matching with the main model. With hierarchical merging, the views could be partitioned into groups (method 1 above) and the resulting partial models merged. If we execute the surface matching and model construction processes in parallel, the non-deterministic algorithms could be adapted to operate on whatever matches are available at a given time. As the surface matching process produces new matches, the model construction process incorporates the results into its future model hypotheses.

7 Automatic modeling experiments

The automatic modeling algorithms will be demonstrated with several range sensors in a variety of environments to emphasize the generality of the approach (table 1). Results will be shown for three application scenarios: desktop/handheld modeling, building modeling, and terrain modeling. These scenarios cover a range of scales, from centimeters to hundreds of meters, and a range of scene types, from highly structured buildings to unstructured terrain.

Application	Sensor	Platform	Scene
desktop and hand-held modeling	Minolta laser scanner, Kodak 3D camera	desktop system	small objects
building modeling	Z&F/Quantapoint laser scanner	cart	building interiors and exteriors
terrain modeling	Z&F/K2T laser scanner	automobile and cart	terrain
	line-scan laser scanner	autonomous helicopter	terrain
	3D sonar	submarine	ocean floor

Table 1: Scenarios for which automatic modeling will be demonstrated.

7.1 Handheld modeling

This application was described in section 1.1. The Minolta laser scanner will be used to obtain range images of a variety of objects. Correspondences for some objects will be measured using fiducial markers (e.g., by marking the surface with dots) that can be identified in the color texture maps produced by the sensor. Ground truth relative poses can be derived from these correspondences, which will provide an objective measure of the quality of the automatic modeling results.

A second set of experiments will investigate the performance of the algorithms on data sets of varying surface complexity and size (i.e., number of input views). These experiments will look at specific cases where symmetries in overlapping views can only be disambiguated by considering global consistency constraints (figure 3). Additionally, the exhaustive registration algorithms will be compared to the selective ones in order to understand the trade-off between success-rate and scaleability. For these tests, we will attempt to obtain the large data sets from Stanford's Digital Michaelangelo Project [23].

A third set of experiments will examine the modeling algorithms' performance with degraded data. Two types of degradation will be studied: random noise and spurious data. For the noise experiments, successively larger magnitudes of random noise will be introduced into the range images. For the spurious data experiments, additional views not belonging to the object being modeled will be intermingled with the valid input views. Another variation will be to corrupt valid views with regions of spurious data, a situation that occurs in practice when objects that are not part of the target scene are accidentally scanned.

Finally, a fourth set of experiments will consider scenarios where some information (such as relative pose estimates or overlapping view labelling) is known *a priori*. These experiments will demonstrate that this information can be incorporated into the automatic modeling framework in a simple yet principled manner.

7.2 Terrain modeling

The terrain modeling experiments will be based on data from three sensing modalities: air, land, and sea. The air-based data was obtained from Carnegie Mellon's Autonomous Helicopter project and includes several data sets of rocky cliffs with ground truth [26]. The land-based data was collected using a cart-based system and a system mounted on an off-road vehicle and covers large portions of a slag heap in the Pittsburgh area. One large data set of 34 scans includes ground truth obtained using fiducial markers which were surveyed with differential GPS. The sea-based data, courtesy of the Applied Research Labs (ARL) at the University of Texas, includes scans of a lake-bottom and the ocean floor with ground truth collected using ARL's 3D sonar sensor.

The terrain data presents a challenging registration problem for several reasons. The data typically is relatively featureless, and contains large regions of constant or near-zero curvature. The data sets are extremely large, leading to input surface meshes with several million faces. The low-elevation, forward-looking placement of the ground-based sensor gives rise to data with widely varying resolution; scene regions sensed at high resolution may appear differently when the same region is sensed at a low resolution in another view. These issues were addressed initially in [18] and incorporating those results into the automatic modeling algorithms should enable the modeling system to handle terrain data.

7.3 Building modeling

The final demonstration of the automatic modeling system will test the algorithms on scenes of building interiors and exteriors. In addition to using existing interior data of collected by members of our lab [6], other interior and exterior scenes will be modeled using a laser scanner from Z&F/Quantapoint.

8 Conclusion

Automatic modeling is a challenging and useful problem that has not received significant attention from the 3D computer vision community. The ability to automatically create 3D models is a fundamental advancement in the field of 3D modeling, enabling the application of 3D modeling to new problems and simplifying existing applications. Unlike previous 3D modeling approaches, automatic modeling works in a variety of domains without relying on mechanical positioning systems or tedious manual operation.

We have proposed two classes of automatic modeling algorithms (exhaustive registration and selective registration) and placed them in a common framework based on the model graph. To support these algorithms, we have presented a number of component tools, including registration algorithms (uninformed pair-wise registration, pair-wise registration refinement, and multi-view registration), local verification procedures (overlap measures, visibility consistency checks, and differential constraint analysis), and global verification methods (topological inference and conflict detection).

8.1 Expected contributions

The fundamental contribution of the thesis will be the introduction of the concept of automatic 3D modeling and a proposed solution to the problem. In this work, we will:

- develop a framework for automatic modeling based on the idea of the model graph,
- implement modeling algorithms within this framework, and
- demonstrate new applications that would not be possible without automatic modeling

Additionally, we expect a number of smaller contributions to arise from this work, including visibility consistency checking, multi-view registration refinement algorithms, and techniques for 3D topological inference.

Summer 2000	• thesis proposal • visibility consistency checking • finish multi-view registration port to C++
Fall 2000	• implement exhaustive registration algorithms • topological inference
Spring 2001	• experiments using exhaustive registration algorithms • implement selective registration algorithms
Fall 2001	• experiments using selective registration algorithms • write thesis
Spring 2002	• defend thesis

Table 2: Automatic modeling research schedule.

8.2 Schedule

Table 2 provides a high-level timeline of the automatic modeling research agenda.

References

[1] P. A. Beardsley, P. H. S. Torr, and A. Zisserman. 3D model acquisition from extended image sequences. In *Proc. 4th European Conference on Computer Vision, LNCS 1065, Cambridge*, pages 683–695, 1996.

[2] R. Benjemaa and F. Schmitt. A solution for the registration of multiple 3d point sets using unit quaternions. In *Proceedings of the 5th European Conference on Computer Vision (ECCV '98)*, pages 34–50. Springer-Verlag, June 1998.

[3] R. Bergevin, M. Soucy, H. Gagnon, and D. Laurendeau. Towards a general multi-view registration technique. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 18(5):540–7, May 1996.

[4] P. Besl and N. McKay. A method of registration of 3-d shapes. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):239–256, February 1992.

[5] R. Bolles and P. Horaud. 3dpo: a three-dimensional part orientation system. *International Journal of Robotics Research*, 5(3):3–26, Fall 1986.

[6] O. Carmichael and M. Hebert. Unconstrained registration of large 3d point sets for complex model building. In *Proceedings 1998 IEEE/RSJ International Conference On Intelligent Robotic Systems*, 1998.

[7] C.-S. Chen, Y.-P. Hung, and J.-B. Cheng. Ransac-based darces: a new approach to fast automatic registration of partially overlapping range images. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 21(11):1229–34, November 1999.

[8] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and Vision Computing*, 10(3):145–55, April 1992.

[9] K. Cho, P. Meer, and J. Cabrera. Performance assessment through bootstrap. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 19(11):1185–98, November 1997.

[10] C. S. Chua and R. Jarvis. Point signatures: a new representation for 3d object recognition. *International Journal of Computer Vision*, 25(1):63–85, October 1997.

[11] P. Debevec, C. J. Taylor, and J. Malik. Modeling and rendering architecture from photographs: a hybrid geometry- and image-based approach. In *Computer Graphics Proceedings SIGGRAPH '96*, pages 11–20. ACM, August 1996.

[12] D. Eggert, Fitzgibbon, and R. Fisher. Simultaneous registration of multiple range views for use in reverse engineering of cad models. *Computer Vision and Image Understanding*, 69(3):253–72, March 1998.

[13] S. F. El-Hakim, P. Boulanger, F. Blais, and J.-A. Beraldin. A system for indoor 3-d mapping and virtual environments. In *Proceedings of Videometrics V (SPIE vol. 3174)*, pages 21–35. SPIE, July 1997.

[14] M. Garland. *Quadric-Based Polygonal Surface Simplification*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, May 1999.

[15] M. Garland and P. Heckbert. Simplifying surfaces with color and texture using quadric error metrics. In *Visualization '98*, October 1998.

[16] J. Goldberger. Registration of multiple point sets using the em algorithm. In *Proceedings of the Seventh IEEE International Conference on Computer Vision*, pages 730–6. IEEE Computer Society Press, September 1999.

[17] K. Higuchi, M. Hebert, and K. Ikeuchi. Building 3-d models from unregistered multiple range images. *Transactions of the Institute of Electronics, Information and Communication Engineers*, J79D-II(8):1354–61, August 1996.

[18] D. F. Huber and M. Hebert. A new approach to 3-d terrain mapping. In *Proceedings of the 1999 IEEE/RSJ International Conference on Intelligent Robotics and Systems (IROS '99)*, pages 1121–1127. IEEE, October 1999.

[19] A. Johnson. *Spin-Images: A Representation for 3-D Surface Matching*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997.

[20] A. Johnson, O. Carmichael, D. F. Huber, and M. Hebert. Toward a general 3-d matching engine: Multiple models, complex scenes, and efficient data filtering. In *Proceedings of the 1998 Image Understanding Workshop (IUW)*, pages 1097–1107, November 1998.

[21] A. Johnson and M. Hebert. Surface registration by matching oriented points. In *International Conference on Recent Advances in 3-D Digital Imaging and Modeling*, pages 121–128, May 1997.

[22] K. Kutulakos and S. Seitz. A theory of shape by space carving. In *Proceedings of the Seventh IEEE International Conference on Computer Vision (ICCV '99)*, pages 307–314. IEEE Computer Society Press, September 1999.

[23] M. Levoy. The digital michelangelo project. In *Second International Conference on 3-D Digital Imaging and Modeling*, pages 2–11. IEEE Computer Society Press, October 1999.

[24] F. Lu and E. Milios. Globally consistent range scan alignment for environment mapping. *Autonomous Robots*, 4(4):333–49, October 1997.

[25] J. R. Miller and O. Amidi. 3-d site mapping with the cmu autonomous helicopter. In *Proceedings of the 5th International Conference on Intelligent Autonomous Systems (IAS-5)*, June 1998.

[26] J. R. Miller, O. Amidi, and M. Delouis. Arctic test flights of the cmu autonomous helicopter. In *Proceedings of the Association for Unmanned Vehicle Systems International 1999, 26th Annual Symposium*, July 1999.

[27] P. Neugebauer. Geometrical cloning of 3d objects via simultaneous registration of multiple range images. In *Proceedings of the 1997 International Conference on Shape Modeling and Applications*, pages 130–9. IEEE Computer Society Press, March 1997.

[28] M. Pollefeys, R. Koch, M. Vergauwen, and L. van Gool. Metric 3d surface reconstruction from uncalibrated image sequences. In R. Koch and L. van Gool, editors, *3D Structure from Multiple Images of Large-Scale Environments. European Workshop, SMILE'98*, pages 139–54. Springer-Verlag, June 1998.

[29] W. Press, S. Teukolsky, W. Vetterling, and B. Flannery. *Numerical Recipes in C - The Art of Scientific Computing*. Cambridge University Press, 1992.

[30] K. Pulli. Multiview registration for large data sets. In *Proceedings of the Second Internation-*

al Conference on 3-D Digital Imaging and Modeling (3DIM '99), pages 160–8. IEEE Computer Society Press, October 1999.

[31] P. Robert and D. Minaud. Integration of multiple range maps through consistency processing. In R. Koch and L. van Gool, editors, *3D Structure from Multiple Images of Large-Scale Environments. European Workshop, SMILE'98*, pages 253–65. Springer-Verlag, June 1998.

[32] H. Sawhney, S. Hsu, and R. Kumar. Robust video mosaicing through topology inference and local-to-global alignment. In *Proceedings of the 5th European Conference on Computer Vision (ECCV '98)*, pages 103–19. Springer-Verlag, June 1998.

[33] V. Sequeira, K. Ng, E. Wolfart, J. Goncalves, and D. Hogg. Automated 3d reconstruction of interiors with multiple scan-views. In *Proceedings of Videometrics VI (SPIE vol. 3641)*, pages 106–17. SPIE, January 1999.

[34] G. Shaffer. *Two-dimensional mapping of expansive unknown areas*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, October 1995.

[35] D. Simon. *Fast and Accurate Shape-Based Registration*. PhD thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, December 1996.

[36] R. Smith and P. Cheeseman. On the representation and estimation of spatial uncertainty. *International Journal of Robotics Research*, 5(4):56–68, Winter 1986.

[37] M. Soucy and D. Laurendeau. A general surface approach to the integration of a set of range views. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 17(4):344–58, April 1995.

[38] F. Stein and G. Medioni. Structural indexing: efficient 3-d object recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 14(2):125–45, February 1992.

[39] A. J. Stoddart and A. Hilton. Registration of multiple point sets. In *Proceedings of the 13th International Conference on Pattern Recognition*, pages 40–4. IEEE Computer Society Press, August 1996.

[40] S. Sullivan and J. Ponce. Automatic model construction and pose estimation from photographs using triangular splines. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(10):1091–7, October 1998.

[41] C. Tomasi and T. Kanade. Shape and motion from image streams under orthography: a factorization method. *International Journal of Computer Vision*, 9(2):137–154, November 1992.

[42] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *Proceedings of SIGGRAPH 94*, pages 311–18. ACM, July 1994.

[43] M. Wheeler. *Automatic modeling and localization for object recognition*. PhD thesis, Carnegie Mellon University, Pittsburgh, PA, October 1996.

[44] Z. Zhang. Iterative point matching for registration of free-form curves and surfaces. *International Journal of Computer Vision*, 13(2):119–152, October 1994.