ELSEVIER

Image and Vision Computing 21 (2003) 637–650

image AND vision COMPUTING

www.elsevier.com/locate/imavis

Fully automatic registration of multiple 3D data sets

Daniel F. Huber*, Martial Hebert

Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, USA

Abstract

This paper presents a method for automatically registering multiple rigid three dimensional (3D) data sets, a process we call multi-view surface matching. Previous approaches required manual registration or relied on specialized hardware to record the sensor position. In contrast, our method does not require any pose measuring hardware or manual intervention. We do not assume any knowledge of initial poses or which data sets overlap. Our multi-view surface matching algorithm begins by converting the input data into surface meshes, which are pair-wise registered using a surface matching engine. The resulting matches are tested for surface consistency, but some incorrect matches may be indistinguishable from correct ones at this local level. A global optimization process searches a graph constructed from the pair-wise matches for a connected sub-graph containing only correct matches, employing a global consistency measure to eliminate incorrect, but locally consistent, matches. From this sub-graph, the rigid-body transforms that register all the views can be computed directly. We apply our algorithm to the problem of 3D digital reconstruction of real-world objects and show results for a collection of automatically digitized objects.

Keywords: Three dimensional modeling; Object modeling; Surface matching; Visibility consistency; Model graph; Registration

1. Introduction

The advent of relatively low-cost, commercially available laser range sensors has greatly simplified the process of accurately measuring the 3D structure of real-world environments, driving the need to automate the processing of 3D data. One problem frequently encountered in 3D data processing is registration, the process of aligning multiple 3D data sets in a common coordinate system. In existing applications, registration is accomplished either by hand or through the use of an external position measurement device such as a calibrated turntable. This paper introduces a third alternative: multi-view surface matching, which does not require any external measurements or manual intervention. Formally, we want to solve the following problem:

> Given an unordered set of overlapping 3D views of a static scene and no *additional information*, automatically recover the viewpoints from which the views were originally obtained, thereby registering the views in a common coordinate system.

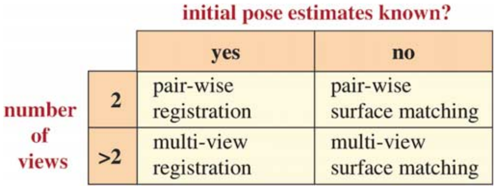
We do not assume any prior knowledge of the original sensor viewpoints or which views contain overlapping scene regions (*overlaps*). Furthermore, the views are unordered, meaning that consecutive views are not necessarily close together spatially. This problem is analogous to assembling a jigsaw puzzle in 3D. The views are the puzzle pieces, and the problem is to correctly put the pieces together without even knowing what the puzzle is supposed to look like.

The relationship between multi-view surface matching and existing 3D registration problems is illustrated by the simple taxonomy shown in Fig. 1. The taxonomy contains two axes: the number of input views and whether initial pose estimates are known. A pose estimate is a rigid body transform (e.g. three rotations and three translations) and can be specified for a single view in world coordinates (*absolute pose*) or between a pair of views (*relative pose*). When an initial estimate of the relative pose is known, aligning two views is called *pair-wise registration*. When more than two views are involved and initial pose estimates are given, the process is called *multi-view registration*. With unknown pose estimates, aligning a pair of views is called *pair-wise surface matching*. Finally, *multiview surface matching* occupies the fourth corner of this taxonomy, extending pair-wise surface matching to more than two views. The problems in the first three categories are interesting research topics on their own, but they are particularly relevant here because we use

* Corresponding author.

E-mail addresses: dhuber@ri.cmu.edu (D.F. Huber); hebert@ri.cmu.edu (M. Hebert).

0262-8856/03/$ - see front matter © 2003 Elsevier Science B.V. All rights reserved.
doi:10.1016/S0262-8856(03)00060-X



		initial pose estimates known?	
		yes	no
number of views	2	pair-wise registration	pair-wise surface matching
	>2	multi-view registration	multi-view surface matching

Fig. 1. A simple taxonomy showing the naming convention for the registration algorithms used in this paper.

algorithms from each category as components of our multi-view surface matching algorithm. Rather than invent these components from scratch, we build on existing algorithms in each category.

Multi-view surface matching is a difficult problem because it involves solving three interrelated sub-problems: (1) determining which views overlap; (2) determining the relative pose between each pair of overlapping views; and (3) determining the absolute poses of the views, which is the ultimate goal of multi-view surface matching. There is a mutual dependency between the overlaps and relative poses. If the relative poses are known, the overlapping views can be determined by applying a suitable definition of overlap to the registered pairs. If the overlaps are known, the relative poses can be found using a pair-wise surface matching algorithm. In this case, the results must be manually verified for correctness. Once the overlaps and relative poses are known, the absolute poses can be determined by multiview registration.

For multi-view surface matching, both the overlaps and the relative poses are unknown, which makes the problem considerably harder. We approach the problem by dividing it into two phases: a local registration phase, which operates only on pairs of views, and a global registration phase, which involves all of the views (Fig. 2). In the local registration phase, the N input views (V_i, $i \in 1\ldots N$) are converted to surface meshes (S_i, $i \in 1\ldots N$), and a surface matching algorithm [1] is applied to all view pairs. The resulting matches are verified for surface consistency, but some incorrect matches may be indistinguishable from correct matches at this local level. As a result, the multiview surface matching problem cannot be solved just by looking at the consistency of pairs of views. Instead, we must consider the global consistency of an entire network of views. This is accomplished in the global registration phase. First, the filtered matches from pair-wise registration are collected in an undirected graph called the model graph, which encodes the connectivity between overlapping views. In the global registration phase, we search this model graph for a connected sub-graph containing only correct matches. We pose the search as a mixed continuous and discrete optimization problem. The discrete optimization performs a combinatorial search over the space of connected sub-graphs of the model graph, using a global surface consistency criterion to detect and avoid incorrect, but locally consistent, matches. The continuous optimization adjusts the absolute pose parameters to minimize the distance between all overlapping surfaces, distributing small pair-wise registration errors in a principled way. The final output, the absolute poses of the input views, can be computed directly from the resulting graph[1].

We demonstrate and test our algorithm in the context of 3D object digitization, the purpose of which is to create a 3D digital reproduction of a real-world object (Fig. 3). In our application, which we call hand-held modeling, the object to be digitized is held before a laser scanner while range images are obtained from various viewpoints. This is an easy data collection method, requiring no specialized hardware, minimal training, and only a few minutes to scan an average object. Alternatively, the model can be placed on a table during each scan, or a portable scanner can be moved around while the scene remains stationary. Once data collection is complete, our application produces a digital model of the original object by automatically registering the input views and then merging the registered views into a single entity. Although we illustrate our algorithm with 3D object digitization, the method is general and can be applied in any situation where multiple 3D data sets must be registered.

This paper is organized as follows: we begin by summarizing the related work in Section 2. Section 3 provides the necessary background on the model graph concept. In Section 4, we define and compare three measures of surface consistency. Sections 5 and 6 give the details of the multi-view surface matching algorithm, with Section 5 focusing on the local registration phase and Section 6 dealing with the global registration phase. Section 7 presents our results, including a comparison of three versions of our algorithm on a set of real test objects, an analysis of the accuracy and computational complexity of the algorithms, and examples of our algorithm applied to data from alternate sensors and data collection methods. Finally, in Section 8 we summarize our algorithm and discuss areas of future work.

2. Related work

We begin with a review of the relevant registration algorithms according to the taxonomy in Fig. 1. A survey by Campbell and Flynn provides additional details on these algorithms [2].

Pair-wise surface matching algorithms register two surfaces without requiring any initial estimate of the relative pose, searching globally over the space of relative pose parameters. Such algorithms are often used in 3D object recognition systems, where the goal is to determine the pose (if any) of one or more 3D model objects within a cluttered 3D scene. Typically, these algorithms work by finding

[1] The absolute poses can be determined only up to a rigid body transform. In practice, we express the poses with respect to one of the input views, which is selected arbitrarily.

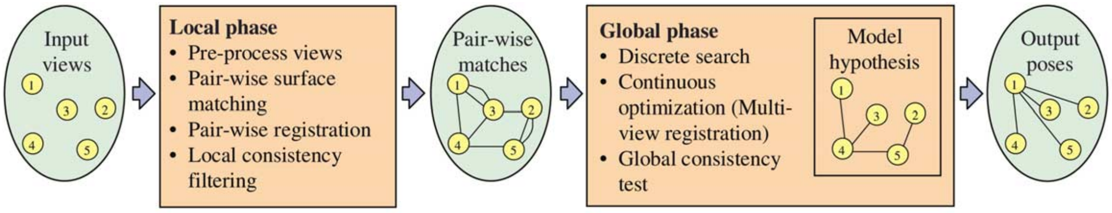


Fig. 2. A block diagram showing the two phases of our multi-view surface matching algorithm. The local phase takes an unordered set of input views and performs pair-wise surface matching, outputting a set of matches. The global phase searches this set of matches for a globally consistent solution, outputting the transforms that place the views in a common coordinate system (shown here with respect to view 1).

correspondences between the model and scene based on an invariant surface property such as curvature. Given a sufficient number of correspondences, the relative pose between model and scene can be estimated. If the registration error is small enough, the object is declared recognized. One approach to establishing model-scene correspondences is to use point signatures, which encode local surface properties in a data structure that facilitates efficient correspondence search and comparison. Proposed encodings include spin-images [1], splashes [3], point signatures [4], harmonic shape images [5], spherical attribute images [6], and the tripod-operator [7]. An alternative approach is to explicitly detect extended features, such as curves of maximal curvature [8], intersections of planar regions [9], or bitangent-curves [10], and match them between model and scene.

In our system, we use a modified version of Johnson's spin-image surface matching algorithm to perform initial pair-wise matching in the local registration phase [1,11,12]. The algorithm is fast, matching two views in 1.5 s, does not require any explicit feature detection, and is robust to holes in the surfaces, which frequently occur when using range image data.

Pair-wise registration algorithms improve upon an initial relative pose estimate by minimizing an objective measure of registration error. The initial relative pose can be provided by a surface matching algorithm, by manual registration, or by the data acquisition system. The dominant method in this category is the iterative closest point (ICP) algorithm, which repeatedly updates the relative pose by minimizing the sum of squared distances between closest points on the two surfaces (point-to-point matching) [13]. Chen and Medioni proposed a similar method in which the distance between points and tangent planes is minimized instead (point-to-plane matching) [14]. Rusinkiewicz' survey of ICP variants provides an elegant taxonomy and unifying framework for comparing the numerous extensions to the basic algorithm [15].

In our system, we use pair-wise registration in the local registration phase to improve pair-wise matches. We have experimented with point-to-point and point-to-plane matching and found that point-to-point matching tends to prevent surfaces from sliding relative to one another, leading to slower convergence in many cases. Therefore, we use the point-to-plane matching method. In practice, we use Neugebauer's multi-view registration algorithm for pair-wise registration (see below) [16]. In the two view case, his algorithm essentially reduces to Chen and Medioni's [14].

Multi-view registration algorithms minimize registration error over an entire network of overlapping views, optimizing the absolute pose parameters of all views. These algorithms require an initial estimate of either the absolute or relative poses. In 3D digitization, multi-view registration is normally used as a final step to improve the overall model quality or as a method for converting from relative poses to absolute poses. A wide variety of

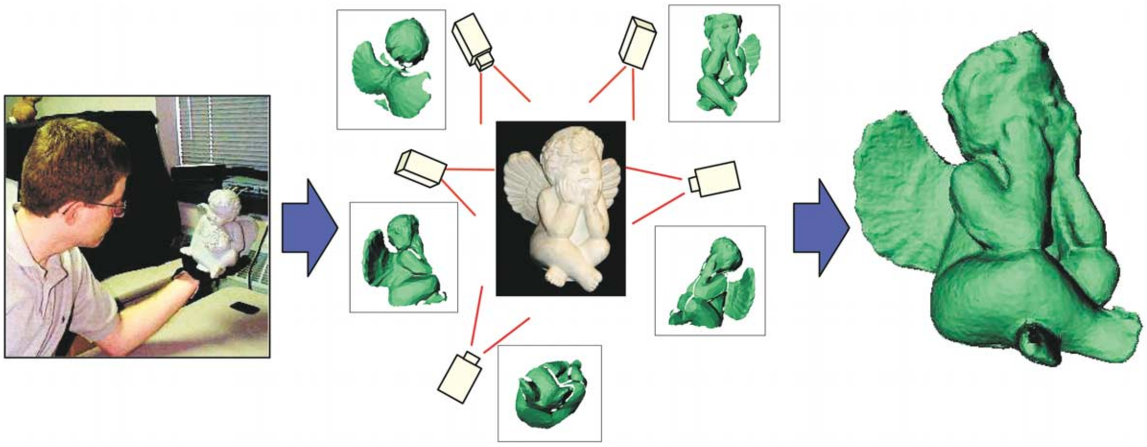


Fig. 3. Hand-held modeling, a 3D digitization application. Holding the object before a laser scanner (left), we obtain 3D data from various viewpoints (center), and automatically construct a digital version of the original object (right). The challenge lies in the uncontrolled poses and arbitrary order of views.

multi-view registration algorithms have been proposed. One approach is to update all the absolute poses simultaneously. In this vein, Benjemaa and Schmitt derived a multi-view extension of the ICP algorithm, and Neugebauer developed a multi-view version of Chen and Medioni's pair-wise algorithm [16,17]. Alternatively, the absolute poses can be updated sequentially. Bergevin et al. repeatedly perform pair-wise registration on pairs of overlapping views using a modified version of Chen and Medioni's algorithm [18]. Pulli uses a similar approach based on the ICP algorithm [19]. Another idea is to view the relative poses as constraints on the absolute poses. Lu and Milios derive uncertainty measures (covariance matrices) for the relative poses and solve for the absolute poses that minimize the Mahalanobis distance between the computed and measured relative poses [20]. 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 [21]. Eggert et al. also use a mechanical system analogy, but they include an inertia term and also update the correspondences over time [22]. Goldberger uses a model-based approach based on the EM algorithm [23]. At each iteration, every view is registered to the current model and then a new maximum likelihood model is created from the updated views.

Our system uses multi-view registration in the global registration phase. We have implemented Benjemaa and Schmitt's point-to-point matching algorithm as well as Neugebauer's point-to-plane matching algorithm. We found that the slower convergence of the pair-wise point-to-point method is exacerbated in the multi-view case, so we opt for Neugebauer's point-to-plane matching approach [16].

Multi-view surface matching algorithms fall into the final category of our taxonomy. To the best of our knowledge, our algorithm is the first in this category [24]. However, since we use multi-view surface matching to automate the process of 3D digitization, it is reasonable to ask how 3D digitization is normally accomplished and to what degree is the process automated. Existing systems use one or more of the following techniques to register the data: a calibrated data acquisition system, manual registration/verification, or environmental modification. With calibrated data acquisition systems, the type of hardware used depends on the size of scene. For smaller objects, absolute poses can be obtained by mounting the sensor on a robot arm [26] or by keeping the sensor fixed and moving the object on a calibrated platform, such as a turntable [27]. For larger scenes, such as the statues scanned in the Digital Michelangelo project, accurate measurement of camera pose becomes considerably more complex, requiring a custom-made gantry system [28]. With manual registration, the user may specify corresponding feature points in pairs of range images, from which relative poses can be estimated [16]. In some systems, corresponding feature points are automatically detected and then manually verified for correctness [29]. Alternatively, the 3D data can be aligned directly through an interactive method [19]. In more advanced approaches, a person indicates only which views to register (i.e. specifies the overlaps), and performs surface matching, manually verifying the results [9,30]. If the motion between successive scans is small, a pair-wise registration algorithm (e.g. ICP) can be used instead, as was done with the Great Buddha digitization project [31] and with Rusinkiewicz' real-time modeling system [32]. In the Pieta project, the environment was augmented with markers, which served as feature points to aid in manual registration [33].

Our algorithm offers several advantages over existing approaches. First, it does not require any specialized data acquisition system, which simplifies the hardware and reduces cost. Second, the algorithm can be applied to 3D data at any scale, so it is not necessary to design one system to model desktop-sized objects and a separate one for modeling buildings. Third, manual registration is time-consuming and tedious—qualities that oppose the widespread acceptance of 3D digitization at the consumer level. Finally, our algorithm enables new applications, such as hand-held modeling, and greatly simplifies data collection.

3. The model graph

Before getting into the details of our algorithm, we introduce the model graph, a concept which simplifies the description of our algorithm and provides a visual interpretation of the registration process. A model graph is an undirected graph G that encodes the topological relationship between views (Fig. 4). It contains a node n_i for each input view V_i and an edge $e_{i,j}$ for each pair of overlapping views V_i and V_j. Associated with each node is an absolute pose T_i and with each edge is a relative pose $T_{i,j}$ as well as additional information, such as registration quality. The relative pose between two connected views V_i and V_j can be computed by composing the relative poses along any path from V_i to V_j in G.

A connected model graph specifies a *complete model* and a potential solution to the multi-view surface matching problem, since every view can be transformed into a common coordinate system by composing relative poses. If G contains several connected components, each component is called a *partial model*. A spanning tree of G is

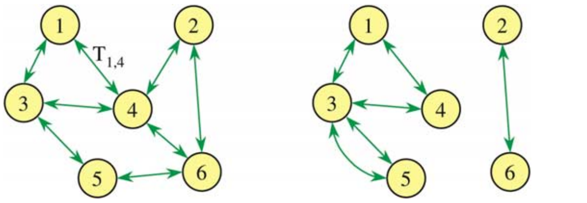


Fig. 4. Example model graphs. A complete model (left) and two partial models (right).

the minimum specification of a complete model. Additional edges will create cycles in G, which can lead to conflicts because composing transforms along different paths between two views may give different results. A model is *pose consistent* if the relative pose of two views is independent of the path in G used for the calculation. In practice, pose inconsistencies arise from the accumulation of small errors in relative pose estimates along a path.

We can now view the registration algorithms used in our system in terms of operations on a model graph G. Surface matching adds one or more edges between two nodes in G. Pair-wise registration updates a single edge (relative pose), while multi-view registration updates all the edges (relative poses) and nodes (absolute poses). Our multi-view surface matching algorithm uses pair-wise surface matching and registration to construct a model graph and then searches this graph for a complete model that is pose consistent and globally surface consistent. In Section 4, we define the term 'surface consistency' and compare several surface consistency measures.

4. Surface consistency

The multi-view surface matching problem would be greatly simplified if we could know with absolute certainty which matches from pair-wise surface matching were correct. This goal cannot be achieved just by looking at pairs of views because two data sets could have zero registration error but still be an incorrect match. The region that would indicate an incorrect match may be far from the overlapping area, and the error may be detectable only indirectly through other matches. To solve this problem, we must look at the consistency of the data at the global level as well.

Surface consistency is a measure of the degree to which the overlapping data from two (or more) surfaces could represent the same physical surface. For any consistency measure, we can define a classifier, which is a thresholded version of the measure. We use surface consistency measures in three ways: (1) to rank the results of pair-wise surface matching; (2) as a classifier that filters out the worst pair-wise matches; and (3) as a basis for a global consistency classifier for verifying entire models.

4.1. Local surface consistency

We have implemented three local surface consistency measures: overlap distance, a general measure that applies to any pair of surfaces; and two measures based on visibility consistency that are tailored to surfaces derived from range images. The measures are defined as error measures—smaller values represent more consistent surfaces. The input surfaces are assumed to be represented in a common coordinate system (i.e. one view is already transformed by the relative pose).

4.1.1. Overlap distance

One way to judge the consistency of two surfaces is to directly measure the distance between the surfaces in overlapping regions. We begin with the following definition of overlap:

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

Given two surfaces represented as meshes, we can estimate the average overlap distance of surface S_i with respect to S_j :

$$O_{\mathrm{D}}(S_i, S_j) = \frac{\sum_{f \in F_{\mathrm{O}}} w_f A(f)}{\sum_{f \in F_{\mathrm{O}}} A(f)} \tag{1}$$

where w_f is the average of the distances between the corners of face f on S_i and the surface S_j, $A(f)$ is the surface area of f, and F_{O} is the set of faces on S_i for which all three corners overlap S_j according to the overlap definition above. Partially overlapping faces are handled as special cases.

We also compute the proportion of S_i that overlaps S_j:

$$O_{\mathrm{P}}(S_i, S_j) = \frac{\sum_{f \in F_{\mathrm{O}}} A(f)}{\sum_{f \in S_i} A(f)} \tag{2}$$

Similarly, O_{D} and O_{P} can be computed for S_j with respect to S_i. Since larger overlapping proportions give a more stable estimate of overlap distance, we define our first local consistency measure to be the weighted average of the two non-symmetric distances:

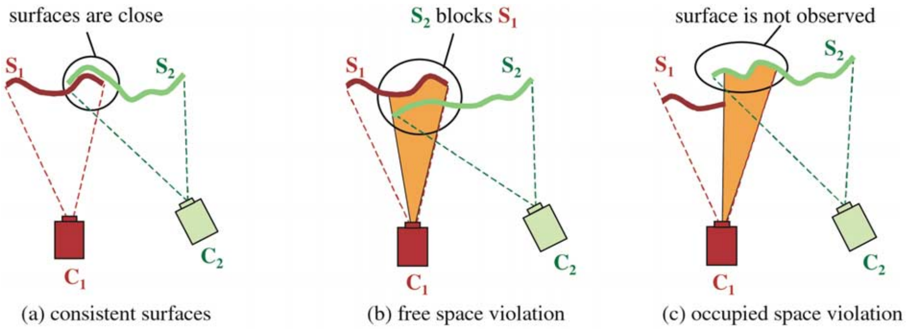
$$\overline{O}_{\mathrm{D}}(S_i, S_j) = \frac{O_{\mathrm{P},i,j} O_{\mathrm{D},i,j} + O_{\mathrm{P},j,i} O_{\mathrm{D},j,i}}{O_{\mathrm{P},i,j} + O_{\mathrm{P},j,i}} \tag{3}$$

where $O_{\mathrm{P},i,j}$ is shorthand for $O_{\mathrm{P}}(S_i, S_j)$ and similarly with $O_{\mathrm{D},i,j}$, $O_{\mathrm{P},j,i}$, and $O_{\mathrm{D},j,i}$.

The disadvantage of overlap distance is that it only takes into account the space close to the two surfaces. In some cases, obviously incorrect matches will have a small overlap distance simply because the overlapping regions have similar shapes. For example, in Fig. 6b, the overlapping region on the angel's head matches well because it is roughly spherical.

4.1.2. Visibility consistency

For range sensors with a single point of projection, we can develop more powerful measures that take advantage of the sensor's entire viewing volume by looking at the consistency of the two surfaces along the line of sight from each of the sensor viewpoints. We call this concept *visibility consistency*. For example, consider the surfaces in Fig. 5



(a) consistent surfaces (b) free space violation (c) occupied space violation

Fig. 5. Visibility consistency from the perspective of sensor position C_1. (a) Consistent surfaces are close together wherever they overlap; (b) a free space violation (FSV) occurs when surface S_2 blocks the view of the observed surface S_1 (highlighted region); (c) an OSV occurs when a portion of S_2 is not observed from C_1 even though it is expected to be visible (highlighted region).

viewed from the sensor position C_1. For a correct registration, the two surfaces have similar range values wherever they overlap (Fig. 5a). For an incorrect registration, two types of visibility inconsistencies can arise. A *free space violation* (FSV) occurs when a region of S_2 blocks the visibility of S_1 from C_1 (Fig. 5b), while an *occupied space violation* (OSV) occurs when a region of S_2 is not observed by C_1, even though it should be (Fig. 5c). Free space violations are so named because the blocking surface violates the assumption that the space is clear along the line of sight from the sensor to the sensed surface. Similarly, OSV surfaces violate the assumption that the range sensor detects occupied space. Here, we focus on FSVs, but the potential of OSVs is discussed briefly in Section 8. Visibility consistency has been used previously in other 3D vision contexts, including hypothesis verification [34], surface registration [22], range shadow detection [35], and multiview integration [36,37].

We can detect FSVs with respect to sensor position C_i by projecting a ray from the center of projection of C_i through a point p on S_i. If the ray passes through S_j at a point q which is significantly closer to C_i than p, then q is an inconsistent point. We must test whether q is *significantly* closer because even for correctly registered surfaces, p and q will not have precisely the same range.

We can efficiently implement FSV detection using two z-buffers [38] to construct synthetic range images (Fig. 6). To compute FSVs for surfaces S_i and S_j with respect to C_i, the surfaces are projected into separate z-buffers and converted into range images (R_i and R_j) using the coordinate system and parameters of C_i (e.g. focal length, viewing frustum). The range difference

$$D_{i,j}(k) = R_i(k) - R_j(k) \tag{4}$$

is then computed for each pixel $x(k)$ where both range images are defined.

We have developed two local consistency measures based on the FSV concept. The first one, which we call the FSV odds, is a statistical measure based on Bayesian decision theory [39]. Given the two possible hypotheses, H^+ (correct match) and H^- (incorrect match), and the set of range difference measurements $D = \{D_{i,j}(1), \ldots, D_{i,j}(K)\}$, we estimate the ratio

$$L(S_i, S_j) = \frac{\Pr(H^+|D)}{1 - \Pr(H^+|D)} = \frac{\Pr(H^+|D)}{\Pr(H^-|D)} = \frac{\Pr(D|H^+)\Pr(H^+)}{\Pr(D|H^-)\Pr(H^-)} \tag{5}$$

Assuming samples of D are independent and taking the logarithm, we have

$$\ln(L(S_i, S_j)) = \sum_{k=1}^{K} \ln \Pr(D_{i,j}(k)|H^+) - \sum_{k=1}^{K} \ln \Pr(D_{i,j}(k)|H^-) + C \tag{6}$$

where $C = \ln \Pr(H^+) - \ln \Pr(H^-)$, which is independent of the data and can be dropped.

An independent measure $L(S_j, S_i)$ can be computed with respect to sensor viewpoint C_j. Frequently, an incorrect match will be detectable from only one viewpoint, so we conservatively combine $L(S_i, S_j)$ and $L(S_j, S_i)$ to form our second consistency measure, the FSV odds:

$$\bar{L}(S_i, S_j) = -\min(\ln(L(S_i, S_j)), \ln(L(S_j, S_i)) \tag{7}$$

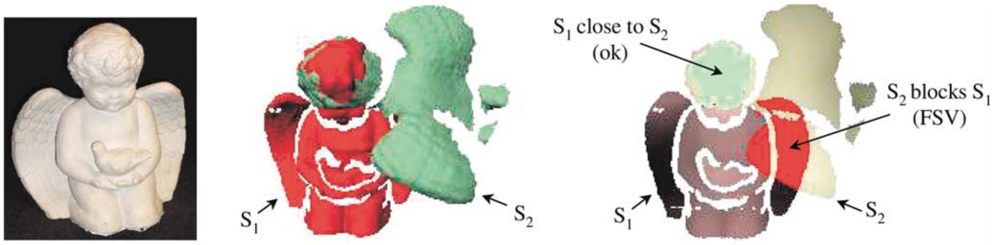
The smaller the value of $\bar{L}(S_i, S_j)$, the more likely it is a correct match[2]. The corresponding FSV odds classifier is

$$\bar{L}_C(S_i, S_j) = \begin{cases} \text{true}, & \text{if } \bar{L}(S_i, S_j) < t_o \\ \text{false}, & \text{otherwise} \end{cases} \tag{8}$$

The distributions $\Pr(D_{i,j}(k)|H^+)$ and $\Pr(D_{i,j}(k)|H^-)$ in Eq. (6) can be estimated from labeled training data. We use a set of hand-labeled matches obtained from exhaustive pair-wise surface matching of the views of a typical object.

First, we compute the range differences for the set of correct matches (Fig. 7, left) and for the set of incorrect matches (Fig. 7, center). We then model $\Pr(D_{i,j}(k)|H^+)$ as a mixture of two Gaussians, one for outliers and one for inliers. The parameters are determined by maximum likelihood estimation. The process is repeated for

[2] The sign change is introduced to make this measure consistent with our other measures such as overlap distance.



a) original object

b) pair-wise match from view 1

c) synthetic range images from view 1

Fig. 6. An example of visibility consistency for an incorrect match. The synthetic range images show the classification of pixels according to Eq. (9). The large number of FSV pixels indicate an incorrect match.

the incorrect matches to estimate $\Pr(D_{i,j}(k)|H^-)$. Mixtures of two Gaussians are necessary because correct matches will contain some outliers, primarily due to small registration errors, and incorrect matches will contain inliers in the region that was matched during surface matching.

For pair-wise matches, the FSV odds is a good measure of surface consistency, but for non-adjacent views in a model graph, the accumulation of error when computing the relative transforms reduces the quality of this measure. For this situation, we use an alternative method for estimating surface consistency, the FSV fraction. To compute the FSV fraction, we apply a threshold, t_{SS}, to the range difference measurements Eq. (4) to classify the overlapping pixels into one of three categories (Fig. 6c): points on the same surface (X_{SS}), points that are FSVs (X_{FSV}), and 'don't care' points where S_j is behind S_i (X_{DC}) :

$$x(k) \in \begin{cases} X_{\mathrm{SS}}(i,j), & \text{if } |D_{i,j}(k)| \leq t_{\mathrm{ss}} \\ X_{\mathrm{FSV}}(i,j), & \text{if } D_{i,j}(k) > t_{\mathrm{ss}} \\ X_{\mathrm{DC}}(i,j), & \text{if } D_{i,j}(k) < -t_{\mathrm{ss}} \end{cases} \tag{9}$$

We then compute the fraction of points that are FSVs, ignoring 'don't care' points (class X_{DC}):

$$F(S_i,S_j) = \frac{|X_{\mathrm{FSV}}(i,j)|}{|X_{\mathrm{FSV}}(i,j)| + |X_{\mathrm{SS}}(i,j)|} \tag{10}$$

As with the FSV odds measure, we can perform the computation with respect to sensor viewpoint S_j to get $F(S_j,S_i)$. Conservatively combining the results gives our third consistency measure, the FSV fraction:

$$\bar{F}(S_i,S_j) = \max(F(S_i,S_j),F(S_j,S_i)) \tag{11}$$

The corresponding FSV fraction classifier is:

$$\bar{F}_{\mathrm{C}}(S_i,S_j) = \begin{cases} \text{true}, & \text{if } \bar{F}(S_i,S_j) < t_{\mathrm{f}} \\ \text{false}, & \text{otherwise} \end{cases} \tag{12}$$

4.1.3. Comparison of local consistency measures

We compare the three consistency measures by evaluating their performance on the task of classifying matches from a test object. By varying the threshold for each classifier and computing the false positive and false negative rates, we can observe how each measure trades off between the two types of errors. The resulting ROC curves (Fig. 7, right) indicate that the two visibility consistency measures are an improvement over the overlap distance measure. This is because they can detect inconsistencies throughout the sensor's entire viewing volume.

4.2. Global surface consistency

Global surface consistency is the extension of local surface consistency to an entire model. A model is globally surface consistent if every pair of views is locally surface

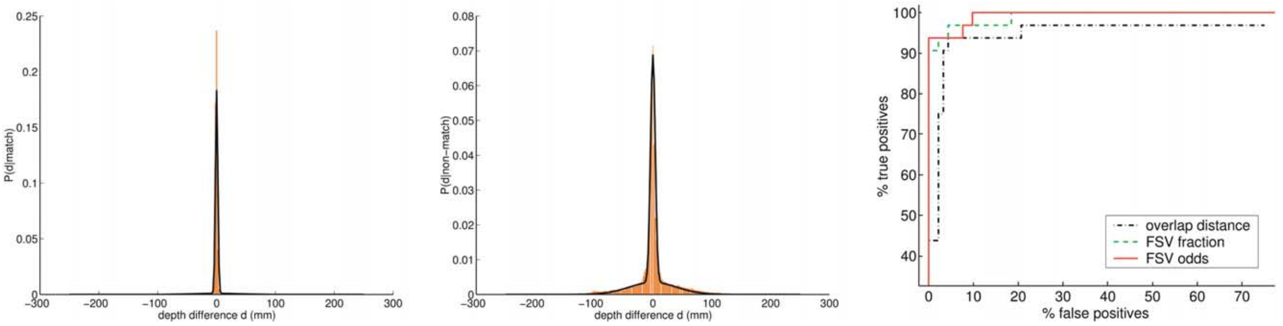


Fig. 7. The distribution of range difference measurements D over a large set of correct matches (left) and incorrect matches (center) from a test object. The predicted distributions, mixtures of two Gaussians learned from a separate training set, are shown overlaid (thin black line). The ROC curves (right) compare the classification accuracy of the three consistency measures.

consistent according to the FSV fraction classifier (Eq. (12)):

$$C_{\mathrm{M}}(G) = \begin{cases} \text{true}, & \text{if } \forall_{(i,j)\in V_{\mathrm{C}}} \bar{F}_{\mathrm{C}}(S_i, T_{i,j}S_j) = \text{true} \\ \text{false}, & \text{otherwise} \end{cases} \tag{13}$$

where V_{C} the set of connected (not necessarily adjacent) view pairs in G, and $T_{i,j}$ is the relative pose computed by composing transforms along a path between nodes n_i and n_j in G. The fact that global consistency is computed between non-adjacent views in G is important. These nonadjacent comparisons produce new, non-local consistency constraints, which makes the global consistency test much more powerful than local consistency tests.

5. Local registration phase

Now that we have defined the model graph and the surface consistency measures, we can fully explain our multi-view surface matching algorithm. The process will be demonstrated on the angel2 test object (Fig. 11d). In the local registration phase, we attempt to register all pairs of views using a surface matching algorithm. For small numbers of views (≈50 or less), this exhaustive registration strategy is reasonable. For larger scenes, the combinatorics make this approach infeasible, and view pairs must be selectively registered (Section 7.2).

In preparation for surface matching, the views are preprocessed in the following manner. The input range images are converted to triangular surface meshes by projecting into 3D coordinates and connecting adjacent range image pixels. Mesh faces within range shadows (which occur at occluding boundaries in the range image) are removed by thresholding the angle between the viewing direction and the surface normal. For computational efficiency, the meshes are simplified using Garland's quadric algorithm [40]. The simplified meshes are used only for pair-wise surface matching; all other operations use the full resolution meshes.

The pair-wise surface matching algorithm registers two surfaces based on their shape. We treat this process as a black box that takes two meshes as input and outputs a list of relative pose estimates. Details can be found in Ref. [30]. If the two views overlap, the algorithm often finds the correct relative pose, but it may fail for a number of data-dependent reasons (e.g. not enough overlap or insufficient complexity of the surfaces). Even if the views do not contain overlapping scene regions, the algorithm may nevertheless find a plausible, but incorrect, match. Furthermore, symmetries in the data may result in multiple matches between a single pair. The model graph of pair-wise matches for the angel2 data set is shown in Fig. 8a. For illustration, the matches have been hand-classified, but these labels are not known by our algorithm. Next, the alignment of each match is improved using pair-wise registration. Finally, we perform a local surface consistency test by applying the FSV odds classifier to the matches Eq. (8). We classify the matches using a conservative threshold chosen with the intention of eliminating obviously incorrect matches without removing any correct ones. The resulting model graph G_{LR} (LR for local registration) is shown in Fig. 8b.

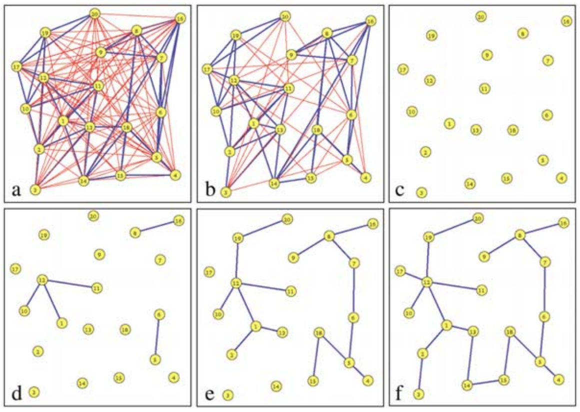


Fig. 8. Model graphs from the local registration phase (a and b) and global registration phase (c–f) for the angel2 test object. Matches were hand-labeled for illustration: thick (blue) edges are correct matches, and thin (red) edges are incorrect matches. (a) The model graph from exhaustive pair-wise registration; (b) after filtering the worst matches (G_{LR}); (c) empty model graph (G_0); (d) after 5 steps of the `full` algorithm; (e) after 15 steps; (f) the final model graph.

6. Global registration phase

The global registration phase uses the locally consistent pair-wise matches (G_{LR}) to construct a pose consistent and globally surface consistent model from which the absolute poses can be read directly. The connected sub-graphs of G_{LR} represent the set of all possible model hypotheses for the given pair-wise matches. To succeed, the global registration must find a sub-graph containing only correct matches. Combinatorics prevent us from exhaustively searching the sub-graphs of G_{LR}. If we restrict our search to acyclic sub-graphs (i.e. spanning trees and their subgraphs), then the problem becomes simpler for two reasons. First, acyclic graphs are guaranteed to be pose consistent, since there is at most one path between any two views. Second, a spanning tree uses the minimal number of matches to specify a complete model, so the chances of including an incorrect match are reduced. In practice, even a single incorrect match results in a dramatically incorrect solution (Fig. 11h). This follows from the fact that our pair-wise registration algorithm converges to a local minimum in registration error over a fairly wide range of initial pose parameters. Therefore, the pair-wise matches are typically either approximately correct or quite wrong. Surprisingly, restricting ourselves to acyclic graphs does not limit the expressiveness of our representation. To see why this is so, consider a graph with cycles and a spanning tree of that graph, which will necessarily be missing some edges from

```
1: $G \leftarrow G_0$
2: for all edges $e_{i,j} \in G_{LR}$, sorted in increasing order using Equation 7 do
3:   if $n_i$ and $n_j$ are not connected in $G$ then
4:     $G' \leftarrow G \cup e_{i,j}$
5:     multi_view_register(G')
6:     if $G'$ is globally surface consistent (Equation 13) then
7:       $G \leftarrow G'$
```

Fig. 9. Pseudo-code for the full algorithm for the global registration phase.

the original graph. The relative pose for any omitted correct match can be inferred by traversing the path in the spanning tree that connects the two nodes.

The global registration can be posed as a mixed discrete and continuous optimization problem over the discrete space of acyclic sub-graphs of G_{LR} and the associated continuous valued absolute pose parameters. We decompose the problem into two nested sub-problems: an inner continuous optimization over absolute poses for a fixed model graph and an outer discrete optimization over model graphs for fixed poses. For the discrete optimization, we sequentially construct a spanning tree from the edges in GLR using a modified version of Kruskal's minimum spanning tree algorithm [41]. Using a spanning tree allows us to directly compute absolute pose estimates for initializing the continuous optimization step. For a fixed graph structure, the continuous optimization is just the multiview registration problem. Our implementation, based on Neugebauer's [16], minimizes the point-to-plane correspondence error. Correspondences are established between *all* overlapping view pairs, not just the edges from the current model graph. At the end of each step, the graph is checked for global surface consistency, which ensures the final solution will be surface consistent and reduces the chances that the algorithm will fall into a local minimum.

The pseudo-code for this algorithm is shown in Fig. 9. Initially, G represents N partial models, each containing one view (line 1). The edges of G_{LR} are sorted by their FSV odds value and tested one at a time. In each iteration through the loop, the best untested edge from G_{LR} is selected, and if it connects two components, a temporary model graph G' is formed, thereby joining two partial models (line 4). The alignment of the views in G' is then improved by multi-view registration (line 5). If the resulting partial model is globally surface consistent (line 6), the new edge is accepted, and G' becomes the starting point for the next iteration (line 7). Eventually, the algorithm either finds a spanning tree of G_{LR}, resulting in a complete model, or the list of edges is exhausted, resulting in a set of partial models. Fig. 8 shows the model graph G at several stages. Once the views are registered, they are merged into a single surface using Curless' VRIP algorithm [42]. The final model, corresponding to the graph in Fig. 8f, is shown in Fig. 10.

In addition to the full algorithm described above (full hereafter), we tested two simpler versions of our algorithm to analyze the effects of the continuous optimization and the global consistency check. The discrete_only algorithm omits the continuous optimization step (line 5), and the min_span algorithm skips the global consistency check (line 6) as well. The min_span algorithm, which finds the minimum spanning tree of G_{LR}, has the advantage that it is simple, fast, and always finds a solution, but the result may not be globally surface consistent. The global consistency test can be performed at the end as a verification, but it is not possible to correct the inconsistency. The discrete_only algorithm integrates the global surface consistency check into min_span, effectively allowing a single step of backtracking. However, the buildup of small pairwise errors can lead to large discontinuities between some overlapping surfaces, which may cause a model to be globally inconsistent even though it contains only correct matches. By incorporating multi-view registration at each step, the errors are distributed over the entire model, allowing the full algorithm to find the correct solution in some cases where discrete_only fails (e.g. angel1).

7. Results

We tested our multi-view surface matching algorithms by digitizing a collection of ten test objects, ranging in size from 190 to 375 mm (Fig. 11). Using a Minolta Vivid 700 laser scanner, we obtained 15 to 20 views of each object, scanning with the hand-held data collection method described in the introduction. A black background and glove allow simple, automatic segmentation of the background by thresholding the intensity image. We compared the performance of the three global registration algorithms described in Section 6 (Table 1). The parameters used in

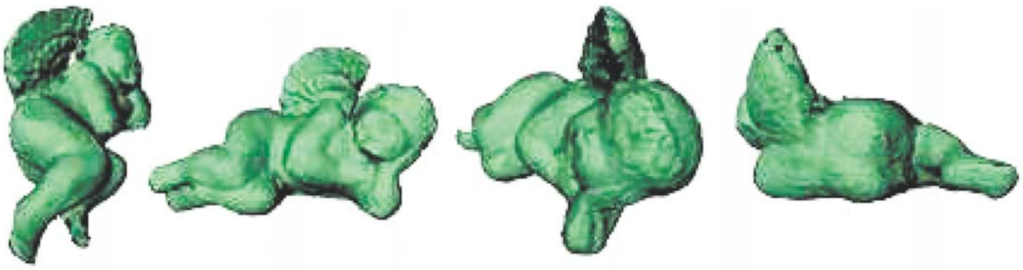


Fig. 10. Four renderings of the automatically modeled angel2 test object.

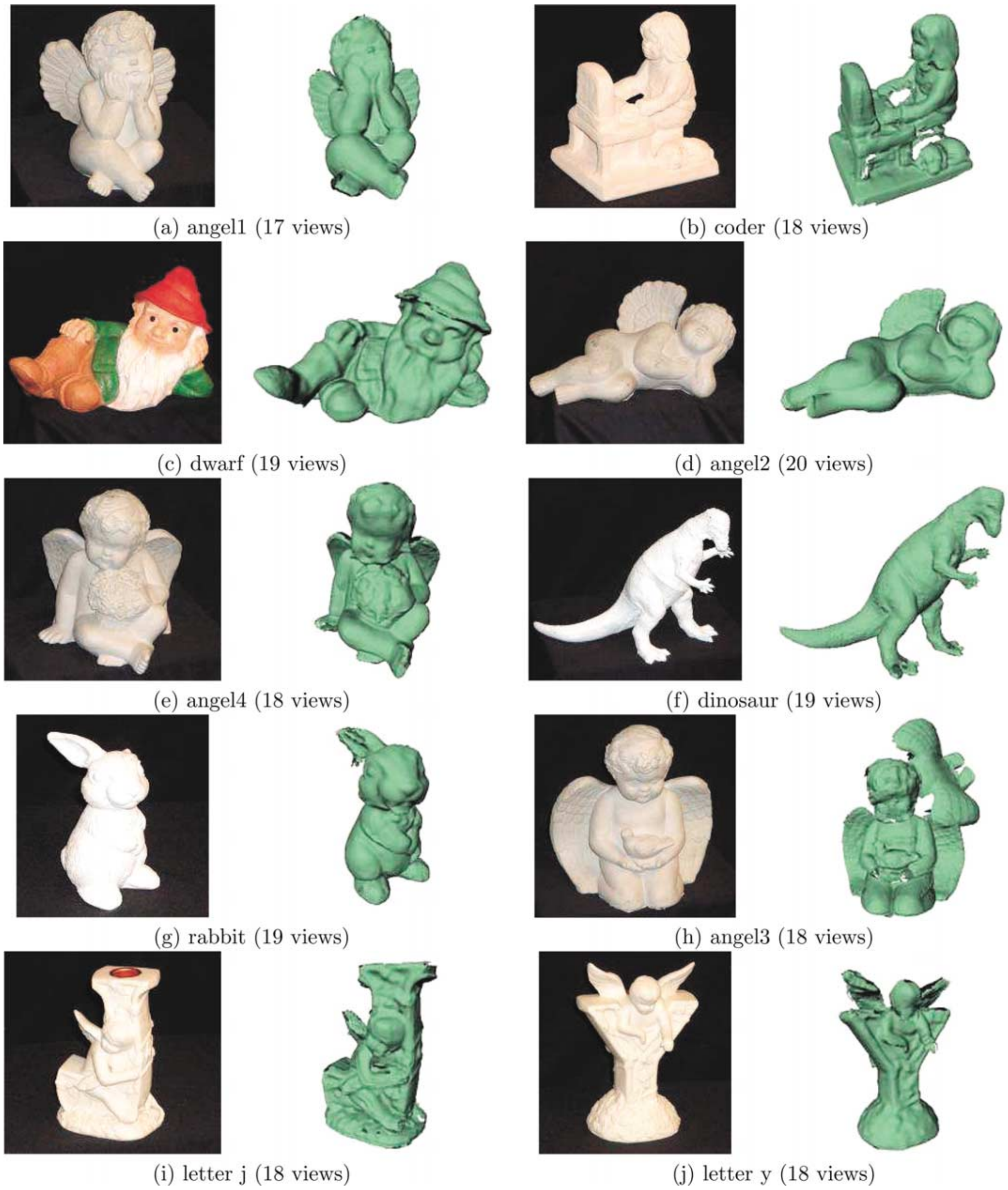

(a) angel1 (17 views)
(b) coder (18 views)
(c) dwarf (19 views)
(d) angel2 (20 views)
(e) angel4 (18 views)
(f) dinosaur (19 views)
(g) rabbit (19 views)
(h) angel3 (18 views)
(i) letter j (18 views)
(j) letter y (18 views)

Fig. 11. Photographs of the 10 test objects (first and third columns) and the resulting 3D models (second and fourth columns). All of the objects were correctly reconstructed except angel3, which contains one error.

the algorithm remained unchanged throughout the experiment (Table 2).

The full algorithm found the correct model (i.e. a model containing only correct matches) in nine of the ten test cases. For two test sets (angel2 and letter y), min_span failed where discrete_only succeeded. This is because one of the most consistent matches was actually an incorrect match, and the discrete_only algorithm correctly detected the incorrect match using the global consistency test. For the angel1 test set, discrete_only failed but full succeeded. In this case, the accumulation of pair-wise error prevented discrete_only from merging the last two components into a consistent model. The components output by discrete_only represented the left and right sides of the object.

Table 1
Performance of the three algorithms on the test models. + indicates a correct result (i.e. a complete model with no incorrect matches); × indicates an incorrect result (i.e. partial models or a graph with incorrect matches)

Object	min_span	discrete_only	full
angel1	+	× (2 partial models)	+
angel2	× (1 err.)	+	+
angel3	× (1 err.)	× (1 err.)	× (1 err.)
angel4	+	+	+
coder	+	+	+
rabbit	+	+	+
dwarf	+	+	+
letter_j	+	+	+
letter_y	× (1 err.)	+	+
dinosaur	+	+	+

Table 2
Parameter settings used in the automatic modeling experiments

Parameter description	Name	Value
Range shadow angle threshold (Section 5)	n/a	80°
Overlap normal angle threshold (Section 4.1.1)	t_θ	45°
Overlap distance threshold (Section 4.1.1)	t_D	20 mm
FSV odds classifier threshold Eq. (8)	t_o	0
Same surface threshold Eq. (9)	t_{ss}	20 mm
FSV fraction classifier threshold Eq. (12)	t_f	0.15

In one case (angel3), none of the algorithms succeeded. This is because pair-wise surface matching did not find any correct matches between two clusters of views: one set representing the front and sides of the object and the other containing views of the back. By searching for a spanning tree of G_{LR} containing only correct matches, we are assuming that such a tree exists. When this assumption is violated, the global registration process cannot succeed. Ideally, the algorithm should recognize that there is no solution and output a set of correct partial models. We could then pass the partial models back into the local registration phase, treating each partial model as a single view. The greater surface area of each partial model can result in matches that would not have been found in the initial matching phase. Another solution is to acquire more data sets that span the boundary region between the two clusters.

7.1. Registration accuracy

While our algorithm outputs models that are qualitatively correct, it is important to establish quantitative methods for evaluating an algorithm's success. We use a method proposed by Wheeler to measure the error in absolute poses [26]. The maximum correspondence error (MCE) for surface S_i is the maximum displacement of any point on S_i from its ground truth position. Note that once the poses reach a certain level of accuracy, this measure actually evaluates the accuracy of the multi-view registration algorithm used within our algorithm.

Since we do not have ground truth poses for the 10 test objects in Fig. 11, we generated synthetic range images of digital objects, corrupting the synthetic range values with Gaussian noise (Fig. 12). The noise level ($\sigma = 1$ mm) and sensor parameters (e.g. focal length, image size, and field of view) were chosen to simulate the worst-case operation of the Vivid 700—scanning a dark-colored object from a distance of 1 m. The synthetic objects were scaled to a size of 200 mm. The MCE for the absolute poses of the object in Fig. 12 was less than 0.135 mm (0.068%) for all 32 input views. For the entire set of 16 automatically modeled synthetic objects, the MCE did not exceed 1.11 mm (0.56%) for any view.

7.2. Complexity

For models with approximately 20 views, the entire modeling process takes 6 to 8 min once data collection is complete, depending on the algorithm used in the global registration phase (Table 3). Three aspects of the process are potential bottlenecks when scaling to large numbers of views. First, the number of view pairs matched during pair-wise surface matching is $O(N^2)$, where N is the number of input views. Even for 19 views, the majority of processing time (4:13 min) is spent performing this operation. This issue can be addressed by using heuristics to selectively register views, exploiting information inherent in each view to sort the views based on the likelihood of a successful match or to partition them into groups that are likely to match with each other. For example, views containing similarly shaped regions should be attempted first. The global consistency test is another $O(N^2)$ operation because every pair of views in a given hypothesis is tested for consistency. However, only view pairs that are physically close to one another actually need to be tested. Geometric data structures can be used to quickly determine which views are close enough to interact. For example, interval trees can be used to compute the intersection of the viewing volume bounding boxes for all view pairs in time $O(N \times \log N + s)$, where s is the number of intersecting pairs [43], and only those that intersect need to be tested for consistency. Finally, multi-view registration involves solving a linear system with $6(N - 1)$ unknowns (the absolute pose parameters), an $O(N^3)$ operation. Fortunately, the system of equations is sparse (the sparseness is related to the number of non-overlapping view pairs), and a substantial

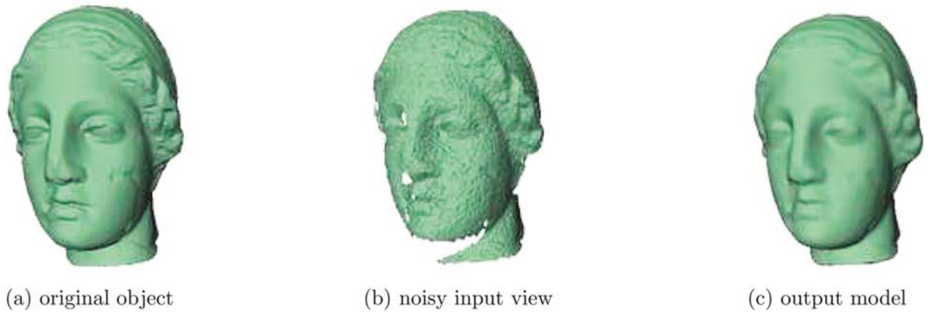

(a) original object (b) noisy input view (c) output model

Fig. 12. (a) An example input object used in the registration accuracy experiment. (b) Synthetic range images were corrupted with noise and then converted to surface meshes. (c) A visualization of the automatically modeled object.

Table 3
Processing time for the dwarf object (19 views) on a Linux PC (1.5 GHz Athlon processor)

Processing stage	Execution time
Local registration phase	6 min. 2 s total
Preprocessing	62 s (3.3 s/view)
Pair-wise surface matching (171 pairs)	253 s (1.5 s/match)
Pair-wise registration (105 matches)	41 s (0.4 s/match)
Local consistency filtering	6 s (0.052 s/match)
Global registration phase	19–87 s
`min_span` algorithm	19 s
`discrete_only` algorithm	27 s
`full` algorithm	87 s

speedup could be achieved through sparse matrix techniques.

7.3. Generality

To test the generality of our method, we constructed models of some larger scale environments. These experiments also demonstrate alternative data collection techniques and modeling with a different sensor. Fig. 13a shows an interior model constructed from data collected by moving the Vivid scanner around the lab. Fig. 13b shows a terrain model constructed using data from a different sensor, the Zoller + Fröhlich (Z + F) LARA 25200 laser scanner. For these models, different settings were used for some of the parameters in Table 2 (e.g. t_{ss} and t_D). In our more recent work, we have eliminated the need to manually set such parameters.

8. Summary and future work

We have presented a method for automatically registering a set of 3D views of a scene. The procedure uses a combination of discrete and continuous optimization methods to construct a globally consistent solution from a set of pair-wise registration results. We defined three measures of surface consistency, one based on overlap distance and two based on visibility consistency. We showed that visibility consistency gives a more accurate prediction of match correctness because it takes into account the entire space between the sensor and the sensed surface. Using these visibility consistency measures, we tested three versions of our multi-view surface matching algorithm, demonstrating their utility by automatically constructing 3D models of a collection of test objects. We also showed that the algorithm accurately registers the input views and that the method can be applied to different data collection scenarios and sensors.

We have identified several aspects of our multi-view surface matching method to be further developed. One problem that can arise in the global registration phase occurs when an incorrect match is added to the graph and the resulting partial model is still globally consistent. Once the algorithm proceeds to the next iteration, there is no hope of finding the correct solution. This problem can be addressed in several ways. One solution is to use a more sophisticated search algorithm. For example, we could incorporate backtracking, turning it into a depth first search of the space of all spanning trees and their subgraphs, or we could use beam search to track multiple hypotheses at each iteration. A second solution is to consider different search heuristics. For example, we could bias the search towards matches that have the most overlap or that would overlap with the most views. A third approach is to use randomized graph search. We have investigated using a RANSAC algorithm, in which spanning trees are randomly sampled from G_{LR} and then evaluated using the global consistency test. Unfortunately, depending on the number and arrangement of incorrect matches in G_{LR}, an unacceptably large number of trials may be required. Currently, we are experimenting with other stochastic methods, such as simulated annealing. The `min_span` algorithm could be used to generate a starting solution for such methods.

Fig. 11h shows an example of a model that contains a single incorrect match. Although obviously wrong, the

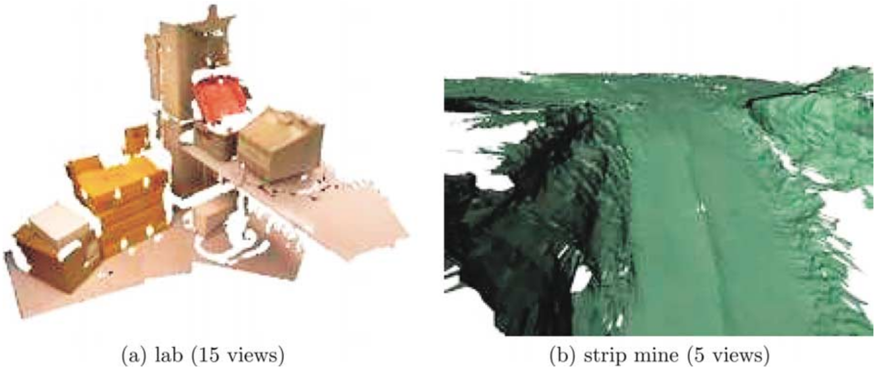

(a) lab (15 views)
(b) strip mine (5 views)

Fig. 13. Models constructed using alternate data collection methods and sensors. (a) For the lab model (model size = 3.25 m), the sensor was moved around the room (texture maps from the original views have been applied); (b) The longer range Z + F LARA 25200 was used for generating the terrain model of a strip mine (model size = 44.6 m).

model is actually consistent according to our global consistency test. This situation could be avoided with an enhanced consistency test that considers OSVs as well as FSVs. Detecting OSVs requires a more sophisticated sensor model than FSVs because surfaces may go undetected for a number of reasons (e.g. the surface is out of sensor range or the normal is too oblique to the viewing direction).

Finally, we must address the issue of scaling to large numbers of views. Although 20 to 50 views are sufficient for modeling small objects and environments, our algorithm's success at this task ensures that eventually we will want to model larger and more complex scenes. To accomplish this, we need to implement and evaluate selective registration and the other optimizations we proposed in Section 7.2.

Acknowledgements

This research has been supported by a fellowship from the Eastman Kodak Company and by the National Science Foundation under Grant 0102272.

References

[1] A. Johnson, M. Hebert, Using spin images for efficient object recognition in cluttered 3D scenes, IEEE Transactions on Pattern Analysis and Machine Intelligence 21 (5) (1999) 433–449.

[2] R. Campbell, P. Flynn, A survey of free-form object representation and recognition techniques, Computer Vision and Image Understanding 81 (2) (2001) 166–210.

[3] F. Stein, G. Medioni, Structural indexing: efficient 3D object recognition, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (2) (1992) 125–145.

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

[5] D. Zhang, M. Hebert, Harmonic maps and their applications in surface matching, Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR) 2 (1999) 524–530.

[6] K. Higuchi, M. Hebert, K. Ikeuchi, Building 3D models from unregistered range images, Graphical Models and Image Processing 57 (4) (1995) 315–333.

[7] F. Pipitone, W. Adams, Tripod operators for recognizing objects in range images: rapid rejection of library objects, Proceedings of the IEEE Conference on Robotics and Automation 2 (1992) 1596–1601.

[8] F.P. Ferrie, M.D. Levine, Integrating information form multiple views, Proceedings of the IEEE Computer Society Workshop on Computer Vision November (1987) 117–122.

[9] P. Allen, I. Stamos, A. Gueorguiev, E. Gold, P. Blaer, AVENUE: Automated site modeling in urban environments, Proceedings of the International Conference on 3D Digital Imaging and Modeling (3DIM) May (2001) 357–364.

[10] J.V. Wyngaerd, L. Van Gool, R. Kock, M. Proesmans, Invariant-based registration of surface patches, Proceedings of the Seventh IEEE International Conference on Computer Vision September (1999) 301–306.

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

[12] O. Carmichael, D.F. Huber, M. Hebert, Large data sets and confusing scenes in 3D surface matching and recognition, Proceedings of the International Conference on 3D Digital Imaging and Modeling (3DIM) October (1999) 358–367.

[13] P. Besl, N. McKay, A method of registration of 3D shapes, IEEE Transactions on Pattern Analysis and Machine Intelligence 14 (2) (1992) 239–256.

[14] Y. Chen, G. Medioni, Object modelling by registration of multiple range images, Image and Vision Computing 10 (3) (1992) 145–155.

[15] S. Rusinkiewicz, M. Levoy, Efficient variants of the icp algorithm, Proceedings of the International Conference on 3D Digital Imaging and Modeling (3DIM) May (2001) 145–152.

[16] P. Neugebauer, Reconstruction of real-world objects via simultaneous registration and robust combination of multiple range images, International Journal of Shape Modeling 3 (1/2) (1997) 71–90.

[17] R. Benjemaa, F. Schmitt, A solution for the registration of multiple 3D point sets using unit quaternions, Proceedings of the Fifth European Conference on Computer Vision (ECCV '98) June (1998) 34–50.

[18] R. Bergevin, M. Soucy, H. Gagnon, D. Laurendeau, Towards a general multi-view registration technique, IEEE Transactions on Pattern Analysis and Machine Intelligence 18 (5) (1996) 540–547.

[19] K. Pulli, Multiview registration for large data sets, Proceedings of the International Conference on 3D Digital Imaging and Modeling (3DIM) October (1999) 160–168.

[20] F. Lu, E. Milios, Globally consistent range scan alignment for environment mapping, Autonomous Robots 4 (4) (1997) 333–349.

[21] A.J. Stoddart, A. Hilton, Registration of multiple point sets, Proceedings of the 13th International Conference on Pattern Recognition August (1996) 40–44.

[22] D. Eggert, Fitzgibbon, R. Fisher, Simultaneous registration of multiple range views for use in reverse engineering of CAD models, Computer Vision and Image Understanding 69 (3) (1998) 253–272.

[23] J. Goldberger, Registration of multiple point sets using the EM algorithm, Proceedings of the Seventh IEEE International Conference on Computer Vision September (1999) 730–736.

[24] D.F. Huber, Automatic 3D modeling using range images obtained from unknown viewpoints, Proceedings of the International Conference on 3D Digital Imaging and Modeling (3DIM), IEEE Computer Society, 2001, pp. 153–160.

[25] D.F. Huber, M. Hebert, Fully automatic registration of multiple 3D data sets, IEEE Computer Society Workshop on Computer Vision Beyond the Visible Spectrum (CVBVS 2001) December (2001) 2001.

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

[27] G. Turk, M. Levoy, Zippered polygon meshes from range images, Proceedings of SIGGRAPH 94 July (1994) 311–318.

[28] M. Levoy, K. Pulli, B. Curless, S. Rusinkiewicz, et al., The digital michelangelo project: 3D scanning of large statues, Proceedings of SIGGRAPH 00 (2000) 131–144.

[29] S.F. El-Hakim, P. Boulanger, F. Blais, J.-A. Beraldin, A system for indoor 3D mapping and virtual environments, Proceedings of Videometrics V (SPIE) 3174 (1997) 21–35.

[30] A. Johnson, Spin Images: A Representation for 3D Surface Matching, Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, Aug. 1997

[31] K. Nishino, K. Ikeuchi, Robust simultaneous registration of multiple range images, Proceedings of the Fifth Asian Conference on Computer Vision January (2002) 454–461.

[32] S. Rusinkiewicz, Real-time Acquisition and Rendering of Large 3D Models, PhD thesis, Stanford University, Stanford, California, August 2001

[33] F. Bernardini, H. Rushmeier, Strategies for registering range images from unknown camera positions, Proceedings of Three-Dimensional Image Capture and Applications III January (2000) 200–206.

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

[35] L. Nyland, D.K. McAllister, V. Popescu, C. McCue, A. Lastra, Interactive exploration of acquired 3D data, Proceedings of the 28th AIPR Workshop: 3D Visualization for Data Exploration and Decision Making October (1999) 46–57.

[36] P. Robert, D. Minaud, in: R. Koch, L. van Gool (Eds.), Integration of multiple range maps through consistency processing, 3D Structure from Multiple Images of Large-Scale Environments, European Workshop, SMILE'98, Springer, Berlin, 1998, pp. 253–265.

[37] M. Soucy, 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) (1995) 344–358.

[38] J. Foley, A. van Dam, S. Feiner, J. Hughes, Computer Graphics Principles and Practice, Addison-Wesley, Reading, MA, 1987.

[39] R. Duda, P. Hart, D. Stork, Pattern Classification, second ed., Wiley, London, 2001.

[40] M. Garland, P. Heckbert, Surface simplification using quadric error metrics, Proceedings of SIGGRAPH 97 (1997) 209–216.

[41] T. Cormen, C. Leiserson, R. Rivest, Introduction to Algorithms, MIT Press, 1990.

[42] B. Curless, M. Levoy, A volumetric method for building complex models from range images, Proceedings of SIGGRAPH '96 (1996) 303–312.

[43] F.P. Preparata, M.I. Shamos, Computational Geometry—An Introduction, Springer, Berlin, 1985.

Daniel Huber received the BS degree in electrical engineering from the University of Texas in 1992. He received the MS degree in computer science from Stanford University in 1996. In 2002, he received the PhD degree in robotics from Carnegie Mellon University. Currently, he is a Research Associate in the Robotics Institute at Carnegie Mellon. His research interests include three-dimensional modeling and recognition, computer vision, and visual navigation for mobile robots.

Martial Hebert is a professor at the Robotics Institute, Carnegie Mellon University. His interests include the development of techniques for representing, comparing, and matching 3D data from sensors, with application to object recognition and model building. His group has developed techniques for fully automatic modeling of 3D scenes from many views and for recognizing objects in complex scenes. His interests also include the development of perception techniques for autonomous mobility for mobile robots navigation and techniques for object recognition in images.