# New Methods for Digital Modeling of Historic Sites

**Peter K. Allen, Alejandro Troccoli, Benjamin Smith, and Stephen Murray**
*Columbia University*

**Ioannis Stamos and Marius Leordeanu**
*The City University of New York*

**We discuss new methods for building 3D models of historic sites. Our algorithm automatically computes pairwise registrations between individual scans, builds a topological graph, and places the scans in the same frame of reference.**

Finding ways to preserve cultural heritage and historic sites is an important problem. These sites are subject to erosion and vandalism, and as long-lived artifacts, they have gone through many phases of construction, damage, and repair. It's important to keep an accurate record of these sites' current conditions by using 3D model building technology, so preservationists can track changes, foresee structural problems, and allow a wider audience to virtually see and tour these sites. Due to the complexity of these sites, building 3D models is time consuming and difficult, usually involving much manual effort. Recently, the advent of new 3D range scanning devices has provided means to preserve these sites digitally and preserve the historic record by building accurate geometric and photorealistic 3D models. This data provides some exciting possibilities for creating models, but at the cost of scaling up existing methods to handle the extremely large point sets these devices create. This reinforces the need for automatic methods of registering, merging, and abstracting the dense range data sets.

Other projects have addressed this and similar problems.[1-6] Each of these projects differs in the way they create models and in the amount of human interaction in the process. Our work centers on developing and automating new methods to recover complete geometric and photometric models of large sites. We're developing methods for data abstraction and compression through segmentation, 3D-to-3D registration (both coarse and fine), 2D-to-3D texture mapping of the models with imagery, and robotic automation of the sensing task. The methods we've developed are also suitable for a variety of other applications related to large-scale model building.

One of the test beds for our model-building methods is the Cathedral Saint-Pierre in Beauvais, France, which is an endangered structure on the World Monuments Fund's Most Endangered List (see the sidebar). We have a number of goals in building our models of the cathedral:

- establish a baseline model for the cathedral's current structural condition,
- create a geometrically accurate 3D model to examine weaknesses in the building and propose remedies, and
- visualize the building in previous contexts as an educational tool.

## Registration methods

To create data sets that we can turn into models, we use a time-of-flight laser scanner (Cyrax 2500) to measure the distance to points on an object. Data from the scanner comprises point clouds, with each point comprising four coordinates $(x, y, z)$ and a value representing the amplitude of the laser light reflected back to the scanner. This fourth coordinate, labeled reflectance strength value (RSV) is a function of the distance to the scanned surface, angle of the surface relative to the laser beam direction, and material properties of the surface. A scan of 1,000 × 1,000 points takes about 10 minutes.

To acquire data describing an entire structure such as the cathedral requires taking multiple range scans from different locations that we must register together correctly. Although we can register the point clouds manually, it's time consuming and error prone. Each range scan can provide up to 1 million data points (see Figure 1 on p. 34). However, manually visualizing millions of small points and matching them is quite imprecise and difficult as the number of scans increases. When possible, we use specially designed targets or fiducials to help during the registration phase. In many cases, such as with the cathedral, it's difficult to place targets. This problem led us to develop our automatic registration methods.

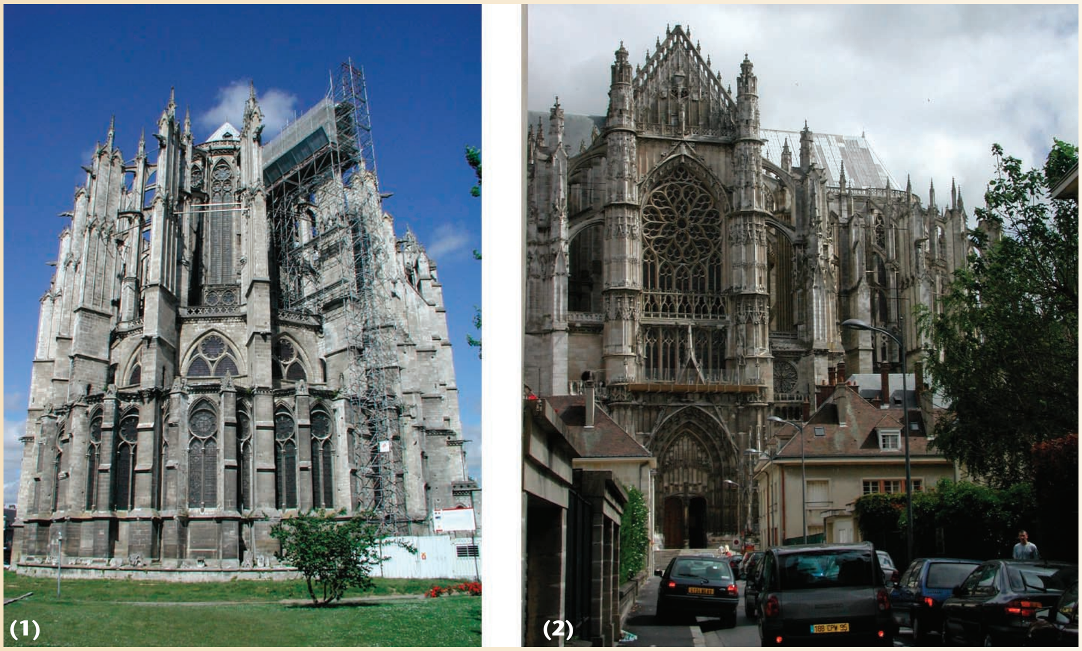
Our registration method is a three-step process.[7] The first step is an automatic pairwise registration between two overlapping scans. The pairwise registration match-

## The Cathedral Saint-Pierre

The Cathedral Saint-Pierre in Beauvais, France—commissioned in 1225 by Bishop Milon de Nanteuil—is an example of high Gothic architecture (see Figure A). Our team of architectural historians, computer scientists, and engineers at Columbia University has begun to study the fragile structure of the tallest medieval cathedral in France. This cathedral collapsed twice in the Middle Ages and continues to pose problems that threaten its long-term survival.

Although the cathedral survived the heavy incendiary bombing that destroyed much of Beauvais during World War II, the structure is as dangerous as it is glorious, being at risk from flaws in its original design, compounded by differential settlement and with stresses placed on its flying buttresses from gale force winds off the English Channel. The winds cause the buttresses to oscillate and have already weakened roof timbers. Between the 1950s and 1980s, numerous critical iron ties were removed from the choir buttresses in a damaging experiment. A temporary tie-and-brace system installed in the 1990s may have made the cathedral too rigid, increasing rather than decreasing stresses upon it. Although researchers have intensively studied the cathedral, there continues to be a lack of consensus on how to conserve the essential visual and structural integrity of this Gothic wonder.

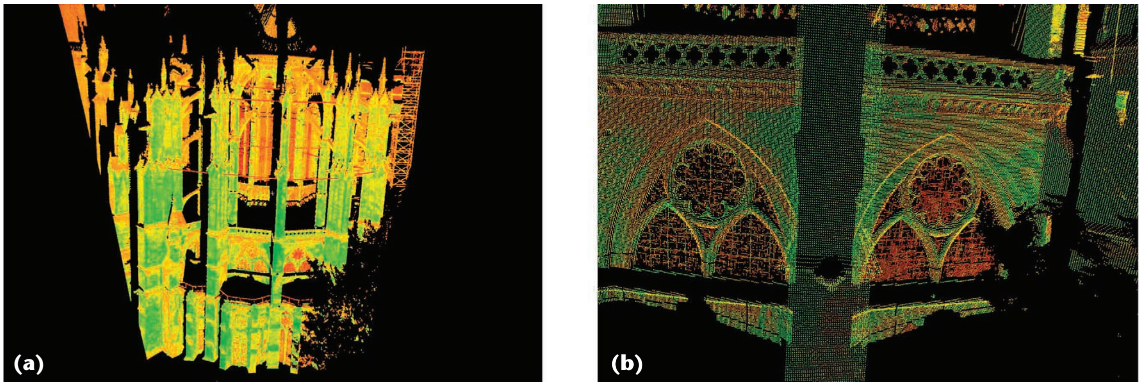
With its five-aisled choir intersected by a towered transept and its great height (keystone 152.5 feet above the pavement), the cathedral provides an extreme expression of the Gothic enterprise. Only the magnificent choir and transepts were completed; the area where the nave and facade would be is still occupied by the previous church constructed just before 1000. The choir was completed by 1272, but in 1284 part of the central vault collapsed, necessitating extensive work of consolidation that continued until the mid-14th century. Closed in with a wooden wall to the west, the choir remained without a transept or nave until work resumed in 1499. Completed by the mid-16th century, the transept was crowned by an ambitious central spire that allowed the Gothic cathedral to rival its counterpart in Rome. The tower collapsed on Ascension Day in 1573. Repairs were completed rapidly, but soon after 1600, attempts to complete the nave were abandoned and the unfinished cathedral closed off with the provisional west wall that has remained until today.



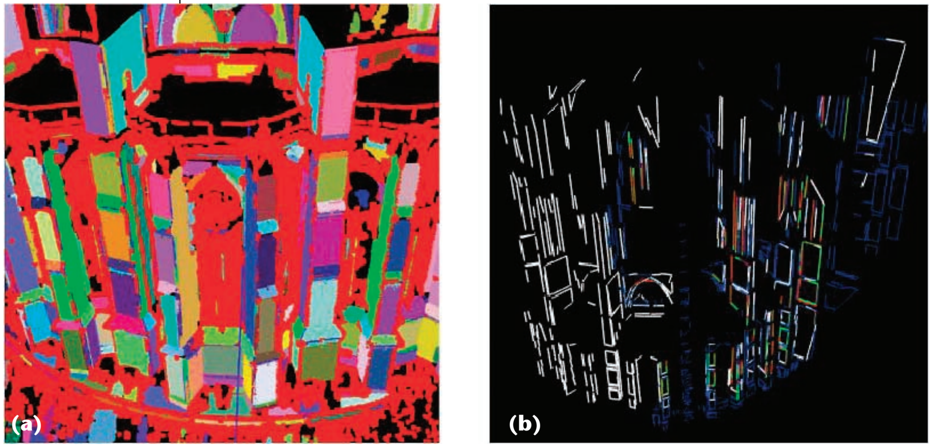
**A Cathedral Saint-Pierre viewed from (1) the east and (2) the south.**

es 3D line segments extracted from overlapping range scans to compute the correct transformation. Next is a global registration step that tries to align all the scans using overlapping pairs. The third step is applying a multi-image simultaneous iterative closest point (ICP) algorithm[8] that does the final fine registration of the entire data set.

With this data set, we use a range segmentation algorithm that we developed[9] to automatically extract planar regions from the point clouds. Once we have these planar features, we can create a set of linear 3D features at the borders of each planar segment. Thus, we convert a 3D range scan into a set of bounded planes and a set of finite lines (see Figure 2). By matching these lines we



**1** **(a) Single dense range scan of the cathedral (approximately 1 million points). (b) Detail of the scan.**



**2** **(a) Segmentation detail in a scan of the cathedral data set. Different colors represent different planes. The red color corresponds to the unsegmented (nonplanar) parts of the scene. (b) Registration of lines between a pair of overlapping scans of the cathedral: left lines (white), right lines (blue), and matched lines (red and green). Due to the overlap of the matched lines, sometimes only the red or the green component dominates.**

can find the correct registration between scans.

After the segmentation step, we extract the following elements from the point clouds: planar regions P, outer and inner borders of those planar regions $B_{out}$ and $B_{in}$, outer and inner 3D border lines $L_{in}$ and $L_{out}$ (defining the borders of the planar regions), and 3D lines of intersection I at the Boolean intersection of the planar regions. Border lines are represented by their two endpoints ($\mathbf{p}_{start}$, $\mathbf{p}_{end}$), and by the plane $\prod$ on which they lie. That is, each border line has an associated line direction and an associated supporting plane $\prod$. In more detail, we represent each border line as a tuple ($\mathbf{p}_{start}$, $\mathbf{p}_{end}$, $p_{id}$, $\mathbf{n}$, $p_{size}$), where $p_{id}$ is a unique identifier of its supporting plane $\prod$, $\mathbf{n}$ is the normal of $\prod$, and $p_{size}$ is the size of $\prod$. We estimate the planes' size by using the number of range points on the plane, the plane's computed distance from the coordinate system's origin, and the plane normal. The additional information associated with each line greatly helps the automated registration.

### *Pairwise registration*

To automatically register a pair of overlapping range scans $S_1$ and $S_2$, we need to solve for rotation matrix $\boldsymbol{R}$ and translation vector $\mathbf{T} = [T_x, T_y, T_z]^T$ that place the two scans in the same coordinate system. Figure 3 shows the algorithm's flowchart. The features that the segmentation algorithm extracted are automatically matched and verified to compute the best rigid transformation between the two scans. The input to the algorithm is a set of lines with associated planes from the segmentation step. Let's call scan $S_1$ the left scan and scan $S_2$ the right scan. Each left line l is represented with the tuple ($\mathbf{p}_{start}$, $\mathbf{p}_{end}$, $p_{id}$, $\mathbf{n}$, $p_{size}$), and each right line r with the tuple ($\mathbf{p'}_{start}$, $\mathbf{p'}_{end}$, $p'_{id}$, $\mathbf{n'}$, $p'_{size}$). The algorithm has four stages.

At a preprocessing step, the algorithm filters out pairs whose ratios of lengths and plane sizes $p_{size}$, $p'_{size}$ are smaller than a threshold. Even though the scanner doesn't identically acquire the overlapping parts of the two scans (because of occlusion and noise), the data is accurate enough for the extracted matching lines to have similar lengths and positions and the matching planes similar sizes. After some experimentation we found thresholds that worked on all pairs of scans, giving results of similar quality. Next, the algorithm sorts all possible pairs of left and right lines (l, r) in lexicographic order.

At Stage 1, the algorithm gets the next not visited pair of lines ($l_1$, $r_1$). It computes the rotation matrix $\boldsymbol{R}$ and estimates the translation $\mathbf{T}_{est}$ assuming the match ($l_1$, $r_1$). Each line is a boundary segment of an associated plane. Hence, we can also use the information about the planes (that is, their normals) to create enough constraints to fully determine the rotation matrix.

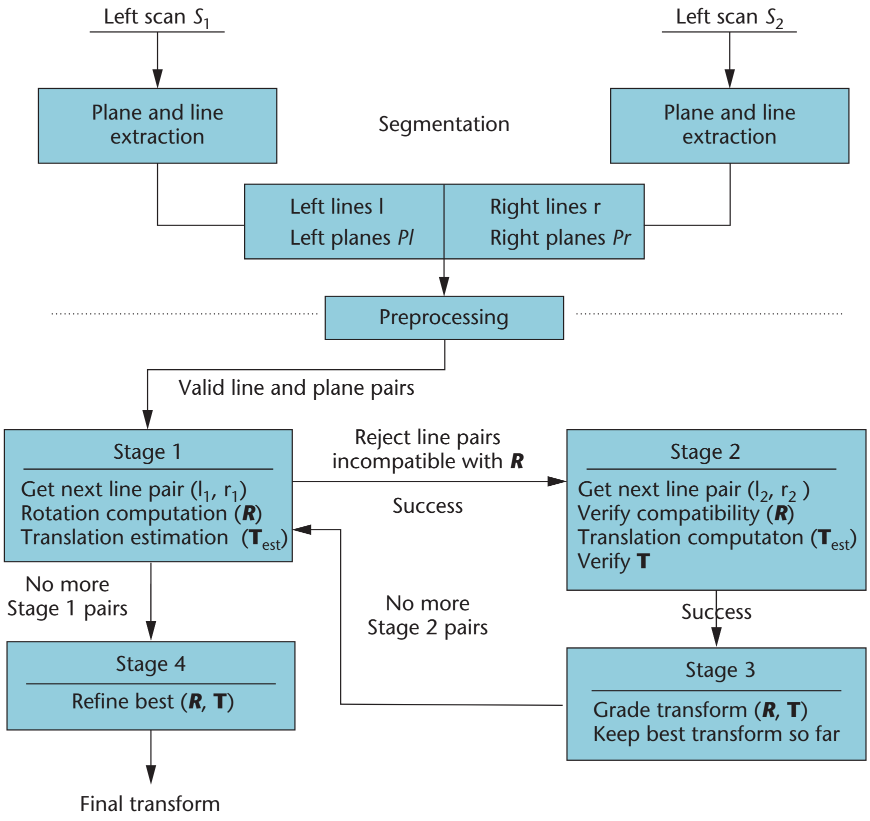
Next, the algorithm applies the computed rotation $\boldsymbol{R}$ to all pairs (l, r) with $l > l_1$. It rejects all line pairs whose directions and normals don't match after the rotation. If the number of remaining matches is less than the current maximum number of matches, then Stage 1 is repeated. Otherwise, accept the match between lines ($l_1$, $r_1$) and their associated planes.

Stage 2 gets the next pair ($l_2$, $r_2$) from the remaining pairs of lines. It rejects the match if it isn't compatible with the estimated translation $\mathbf{T}_{est}$. Otherwise, it computes an exact translation $\mathbf{T}$ from the two pairs ($l_1$, $r_1$) and ($l_2$, $r_2$). Next, it verifies that the two line pairs and their associated plane normals correspond after applying the translation **T**. The algorithm then accepts ($l_2$, $r_2$) as the second match.

Stage 3 grades the computed transformation (***R***, **T**) by transforming all valid left lines to the coordinate system of the right scan and counting the absolute number of valid pairs that correspond. The algorithm then moves to Stage 1.

At Stage 4, after all possible combinations of valid pairs have been exhausted, the algorithm recomputes the best transform (***R***, **T**) by using all matched lines.

The pairwise registration algorithm efficiently computes the best rigid transformation (***R***, **T**) between a pair of overlapping scans $S_1$ and $S_2$. This transformation has an associated grade $g(\boldsymbol{R}, \mathbf{T})$ that equals the total number of line matches after applying the transformation. Note that the grade is small if no overlap exists between the scans.


Left scan $S_1$
Left scan $S_2$
Plane and line extraction
Segmentation
Plane and line extraction
Left lines l
Left planes *Pl*
Right lines r
Right planes *Pr*
Preprocessing
Valid line and plane pairs
Stage 1
Get next line pair ($l_1$, $r_1$)
Rotation computation (***R***)
Translation estimation (**T**$_{est}$)
Reject line pairs incompatible with ***R***
Success
Stage 2
Get next line pair ($l_2$, $r_2$ )
Verify compatibility (***R***)
Translation computaton (**T**$_{est}$)
Verify **T**
No more Stage 1 pairs
No more Stage 2 pairs
Success
Stage 4
Refine best (***R***, **T**)
Stage 3
Grade transform (***R***, **T**)
Keep best transform so far
Final transform

**3 Flowchart for automatic registration between a pair of overlapping range scans.**

Generally, a solution to the problem is possible if the algorithm finds two pairs of matched lines between the two scans $S_1$ and $S_2$. Using these two matched pairs, a closed-form formula[10] provides the desired transformation (***R***, **T**). That means that a blind hypothesis-and-test approach would have to consider all possible

$$\binom{N}{2} \times \binom{M}{2} = O(M^2N^2)$$

pairs of lines, where $N$ and $M$ are the number of lines from scans $S_1$ and $S_2$, respectively. Such an approach is impractical due to the size of the search space to be explored. For each pair of lines, we'd need to compute the transformation (***R***, **T**) and then verify the transformation by transforming all lines from scan $S_1$ to the coordinate system of scan $S_2$. The verification step is an expensive $O(MN)$ operation. With our method, we need only to consider a fraction of this search space. For more information see Stamos and Leordeanu.[7]

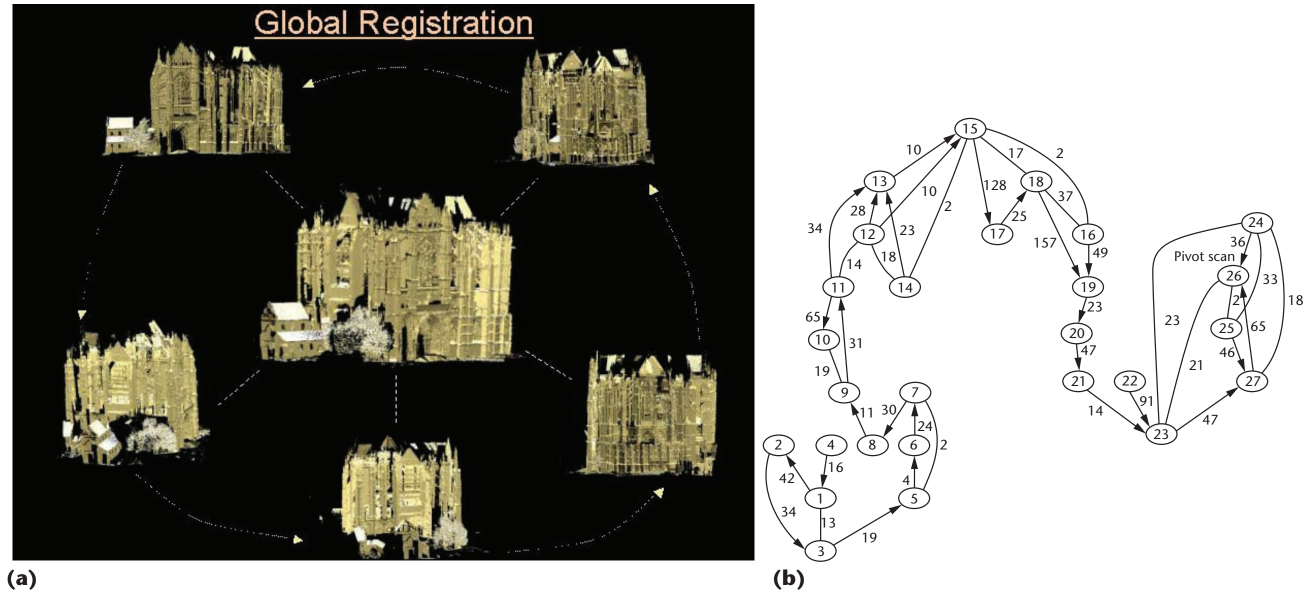
## *Global registration*

In a typical scanning session, tens or hundreds of range scans must be registered. Calculating all possible pairwise registrations is impractical because it leads to a combinatorial explosion (see Figure 4, next page). In our system, the user provides a list of overlapping pairs of scans, which reduces the number of possible pairings. From this reduced list we can compute all pairwise transformations. Then, the system selects the rightmost scan from the graph (a better approach would be to select a node at the center of the graph) to be the anchor scan $S_a$. Finally, all other scans $S$ are registered with respect to the anchor $S_a$. In the final step, we can reject paths of pairwise transformations that contain registrations of lower confidence.

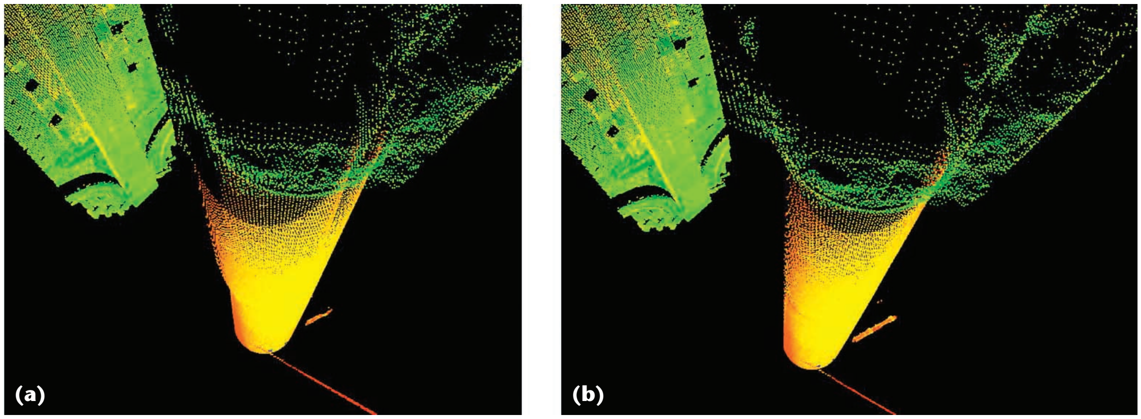
In more detail, our system computes the rigid transformations $(\boldsymbol{R}_i, \mathbf{T}_i)$ and their associated grades $g(\boldsymbol{R}_i, \mathbf{T}_i)$ between each pair of overlapping scans. In this manner, the system constructs a weighted undirected graph. The graph's nodes are the individual scans, and the edges are the transformations between scans. Finally, the grades $g(\boldsymbol{R}_i, \mathbf{T}_i)$ are the weights associated with each edge. More than one path of pairwise transformations can exist between a scan $S$ and the anchor $S_a$. Our system uses a Dijkstra-type algorithm to compute the most robust transformation path from $S$ to $S_a$. If $p_1$ and $p_2$ are two different paths from $S$ to $S_a$, then $p_1$ is more robust than $p_2$ if the cheapest edge on $p_1$ has a larger weight than the cheapest edge of $p_2$. This is the case because the cheapest edge on the path corresponds to the pairwise transformation of lowest confidence (the smaller the weight the smaller the overlap between scans). In this manner, our algorithm uses all possible paths of pairwise registrations between $S$ and $S_a$ to find the path of maximum confidence. This strategy can reject weak overlaps between scans that could affect the quality of the global registration between scans.

## *Simultaneous registration of multiple range images*

Once the system registers the range images using the automatic method previously discussed, it uses a refined ICP algorithm to simultaneously register multiple range



4 (a) Global registration problem. We need to correctly register all scans of the cathedral. (b) Graph of 27 registered scans of the cathedral data set. The nodes correspond to the individual range scans. The edges show pairwise registrations. The weights on the edges show the number of matched lines that the pairwise registration algorithm provides. The directed edges show the paths from each scan to the pivot scan that's used as an anchor.



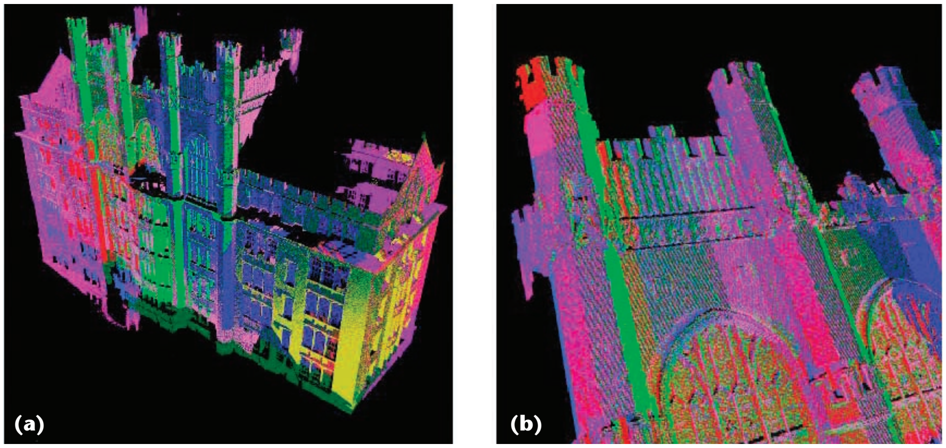
5 (a) Before and (b) after fine registration. Note column misalignment in (a) has been corrected.

images to provide the final registration. This is an extended method of the one proposed by Nishino and Ikeuchi.[11] Their work extends the basic pairwise ICP algorithm to simultaneously register multiple range images. In our approach, an error function is minimized globally against all range images. The error function uses an estimator that robustly rejects the outliers and can be minimized efficiently using a conjugate gradient search framework. To speed up the registration process, the algorithm uses a K-D tree structure to reduce the search time for the matched point. ICP type algorithms work by matching the closest point in one scan to another scan. If the matches are predicated only on point-to-point geometric distance, the algorithm can sometimes cause misalignment. Additional information to suggest better matches is required. For this purpose, the algorithm uses the laser RSV. The idea is that points that are close will have similar RSV values. To find a best match point of a model point, the algorithm searches for multiple closest points in the K-D tree. Then it evaluates the RSV distance (to the model point) for each of them to get the closest point. Once correspondences are made, we can iteratively find the correct transformation matrices for the data points.

The data set for the cathedral contains more than 100 scans, requiring significant computational resources and time to register these scans with full resolution. Therefore, we subsample those scans down to 1/25 of their original resolution to speed up the registration process. Figure 5 shows the results of applying the algorithm on two coarsely registered scans. The column, which is misaligned initially, is correctly aligned after the procedure (see Figure 5b).

## Registration results

We first tested our methods on scans from the Thomas Hunter building at Hunter College in New York (referred to as the campus data set) and then on the cathedral data set. The scans have a nominal accuracy of 6 millimeters along the laser-beam direction at a distance of 50 meters from the scanner. We first segmented each scan and extracted major planes and lines from the scans. We executed the pairwise registration algorithm on pairs of overlapping scans. In the final step, we used the global registration algorithm. Figure 6 shows the results from the campus building: our system automatically registered 10 range scans. Figure 7 shows registration for 27 scans of the cathedral. Figure 7a shows the registration, the relationship of each scanner position in the registered model, and colored segments used in the registration process. Figure 7b is a closeup of the registration around the central window, showing the accuracy of the alignment.



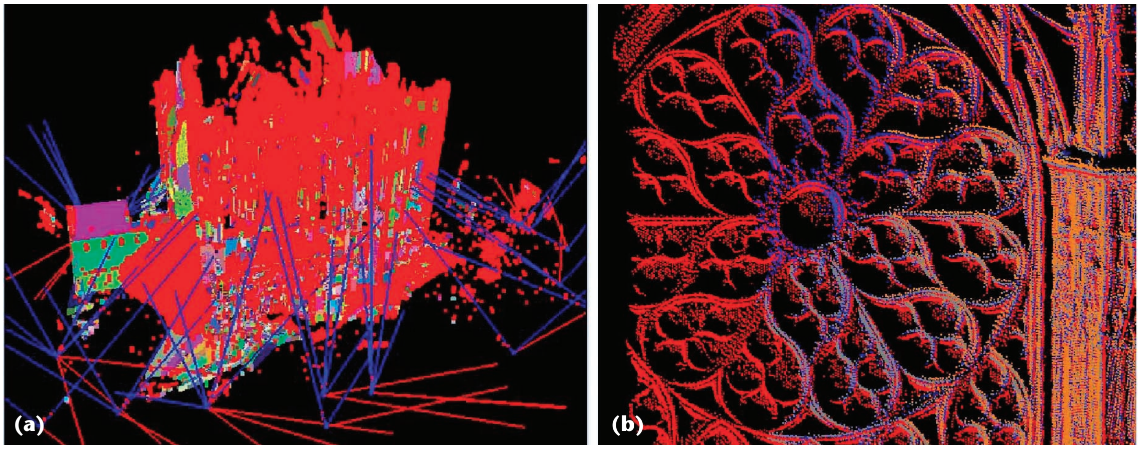
6 Campus site. (a) Ten automatically registered scans. Each scan is represented with a different color. (b) Registration detail.

Tables 1 and 2 (next page) provide an extensive evaluation of our algorithm's efficiency and accuracy. The percentage of line pairs that survive after preprocessing and reach algorithm Stages 2 and 3 demonstrate the algorithm's efficiency. Few lines need to be considered at the expensive Stage 3. The running times range from 3 to 52 seconds (on a 2-GHz Linux machine) per pair, depending on the input size and on the amount of overlap. The tables also show the measured pairwise registration error. This error is the average distance between matched planes lying on the surface of the scans. The error ranges from 1.36 mm to 14.96 mm for the campus data set (see Table 1) and from 5.34 mm to 56.08 mm for the cathedral (see Table 2). The average error over all 10 scans of the campus data set is 7.4 mm and over all 27 scans of the cathedral data set is 17.3 mm.

Each table shows a set of pairwise registrations. The rows represent one registered pair of scans. The second column displays the number of line pairs. The preprocessing column shows the percentage (over all possible pairs) of line pairs that must be considered after the preprocessing step of the algorithm. The percent reaching Stage 2 and 3, respectively, show the percentage (over all possible combinations) and total number of pairs that reach Stage 2 and 3. Matches present the number of matched pairs that the algorithm establishes. The run time of the algorithm is shown for every pair (on a 2-GHz Linux machine). Finally, the pairwise registration error is the average distance between matched planar region between the two scans.

Note that the errors are small if we consider the spatial extent of the 3D data sets. The larger errors in the cathedral data set result from the scans' lower spatial resolution (larger distance between scan lines). The error also increases with the grazing angle between the scan direction and the scanned surface. This level of initial registration is adequate for our modeling task, and we can accomplish finer registration through ICP.

However, the algorithm will fail in some cases. If few linear features exist in the scene, matches aren't possible. We don't find this in urban settings, though, which contain rich sets of linear features. Scene symmetry can also introduce false matches. This isn't an inherent limitation of this particular algorithm, but is a problem with all registration algorithms. We should give extra



7 Cathedral site. (a) Three-dimensional mesh after placing 27 scans in the same coordinate system. Segments used in the registration are colored. The local coordinate system of each individual scan is shown. The *z*-axis of each scan points toward the cathedral. (b) Registration detail.

Table 1. Performance of our algorithm on the campus building.

| | | Survive | Reach | Reach | | | |
|---|---|---|---|---|---|---|---|
| | Line | Preprocessing | Stage 2 | Stage 3 | | Runtime | Error |
| Pair | Pairs | (%) | (%/number) | (%/number) | Matches | (Seconds) | (mm) |
| 1 | 301 × 303 | 16 | 1.7/1555 | 0.38/346 | 35 | 15 | 10.99 |
| 2 | 303 × 290 | 17 | 2.8/2429 | 0.84/735 | 25 | 29 | 6.28 |
| 3 | 290 × 317 | 21 | 2.8/2572 | 1.88/1728 | 36 | 52 | 2.77 |
| 4 | 317 × 180 | 19 | 3.4/1955 | 1.15/656 | 28 | 21 | 14.96 |
| 5 | 211 × 180 | 21 | 4.6/1759 | 2.1/802 | 31 | 19 | 9.26 |
| 6 | 180 × 274 | 17 | 2.6/1306 | 0.34/168 | 22 | 9 | 11.42 |
| 7 | 114 × 274 | 19 | 1.6/507 | 2.2/894 | 33 | 6 | 5.61 |
| 8 | 274 × 138 | 16 | 1.8/667 | 1.5/557 | 31 | 5 | 3.08 |
| 9 | 114 × 138 | 18 | 2.7/423 | 3.8/593 | 32 | 4 | 3.94 |
| 10 | 138 × 247 | 18 | 2.3/791 | 1.3/429 | 20 | 5 | 1.36 |

Table 2. Performance of our algorithm on the cathedral.

| | | Survive | Reach | Reach | | | |
|---|---|---|---|---|---|---|---|
| | Line | Preprocessing | Stage 2 | Stage 3 | | Runtime | Error |
| Pair | Pairs | (%) | (%/number) | (%/number) | Matches | (Seconds) | (mm) |
| 1 | 406 × 464 | 7 | 0.9/1650 | 0.3/615 | 42 | 39 | 9.37 |
| 2 | 464 × 269 | 7 | 0.7/888 | 0.3/443 | 34 | 16 | 16.9 |
| 3 | 406 × 269 | 11 | 0.7/794 | 0.1/104 | 13 | 9 | 56.08 |
| 4 | 151 × 406 | 21 | 1.1/668 | 0.8/480 | 16 | 7 | 5.34 |
| 5 | 269 × 387 | 11 | 0.7/702 | 0.4/369 | 19 | 9 | 15.8 |
| 6 | 326 × 197 | 10 | 0.9/597 | 0.1/49 | 24 | 4 | 11.68 |
| 7 | 197 × 143 | 15 | 1.0/290 | 0.3/82 | 30 | 3 | 6.44 |
| 8 | 143 × 194 | 16 | 1.9/520 | 0.1/31 | 11 | 3 | 29.24 |
| 9 | 194 × 356 | 15 | 2.0/1429 | 0.1/93 | 19 | 11 | 30.82 |

constraints in this case (that is, the user can specify that the rotation and translation should be within certain limits, or the user can add extra point constraints).

We constructed a comprehensive model of the cathedral made up of data from all the scans. The resulting model is large and visualizing the entire model can be difficult. Figure 8 shows the model from different views. For these models, we registered 120 scans on the inside of the cathedral and 47 on the outside. The outside scans were registered automatically except for seven, where we manually added a single extra point constraint to assist the automatic procedure in overcoming symmetries. The inside scans were first coarsely registered manually, before we developed our automatic methods, and were quite time consuming. We then ran our simultaneous ICP algorithm to substantially improve the registration. A 3D video fly-through animation of the model is available at http://www.cs.columbia.edu/~allen/BEAUVAIS.

## Texture mapping

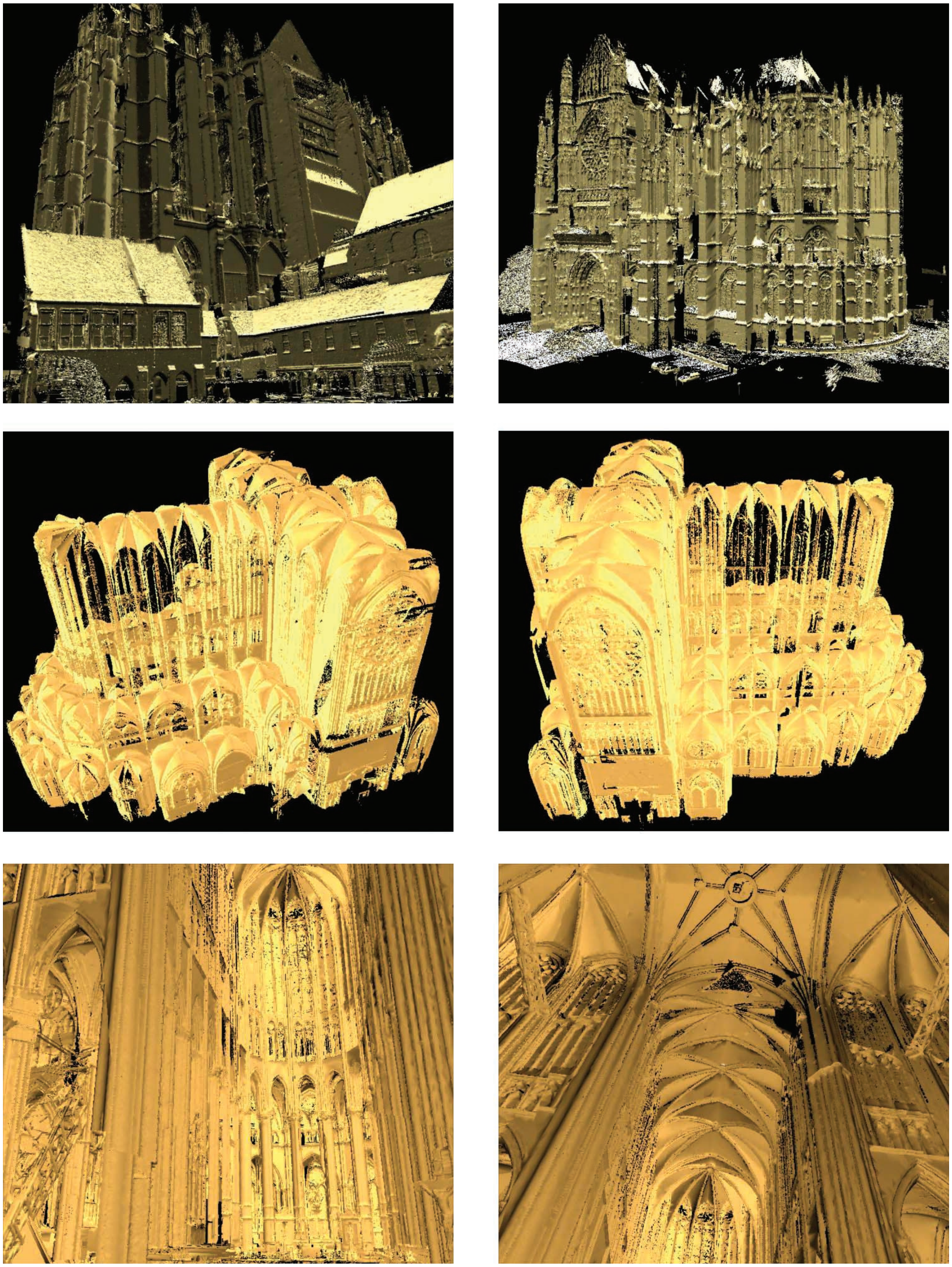
The range data let us build a geometrically correct model. For photorealistic results we mapped intensity images to the polygon meshes. The input to the texture mapping stage is a point cloud, a triangular mesh corresponding to the point cloud, and a 2D image. We generated the triangular mesh using the topology of the points the scanner acquired, providing a 2D grid for each set of scanned 3D points. Due to the size of the point sets, we decimate them before building the mesh.

Once we build the mesh, a user manually selects a set of corresponding points from the point cloud and the 2D image (we use the points to compute a projection matrix $\boldsymbol{P}$ that transforms world coordinates to image coordinates). Let $(\mathbf{X}_i, \mathbf{x}_i)$ be a pair of 3D and 2D homogeneous point correspondences, with $\mathbf{X}_i$ and $\mathbf{x}_i$ of the form $(X_i, Y_i, Z_i, W_i)$ and $(x_i, y_i, w_i)$ respectively. Each pair provides the following two equations:

$$\begin{bmatrix} \mathbf{0}^T & -w_i\mathbf{X}_i^T & y_i\mathbf{X}_i^T \\ w_i\mathbf{X}_i^T & \mathbf{0}^T & -x_i\mathbf{X}_i^T \end{bmatrix} \begin{pmatrix} \mathbf{P}^1 \\ \mathbf{P}^2 \\ \mathbf{P}^3 \end{pmatrix} = 0$$

where each $\mathbf{P}^i$ is a row of $\mathbf{P}$. By stacking up the equations derived from a set of $n$ pairs, we obtain a $2n \times 12$ matrix $\boldsymbol{A}$. The solution vector $\mathbf{p}$ of the set of equations $\boldsymbol{A}\mathbf{p} = 0$ contains the entries of the matrix $\boldsymbol{P}$. We need at least six point correspondences to obtain a unique solution. In practice, we use an over-determined system, which we solve using the singular value decomposition of matrix $\boldsymbol{A}$. Prior to solving the system of equations, we normalize both 3D and 2D points to improve numerical stability. This normalization consists of a translation and scaling step; both 2D and 3D points are translated so that their centroid is at the origin and then scaled so that the root-mean-squared distance to the new origin of the point sets is $\sqrt{2}$ and $\sqrt{3}$, respectively.

Once we obtain the projection matrix $\boldsymbol{P}$, we compute



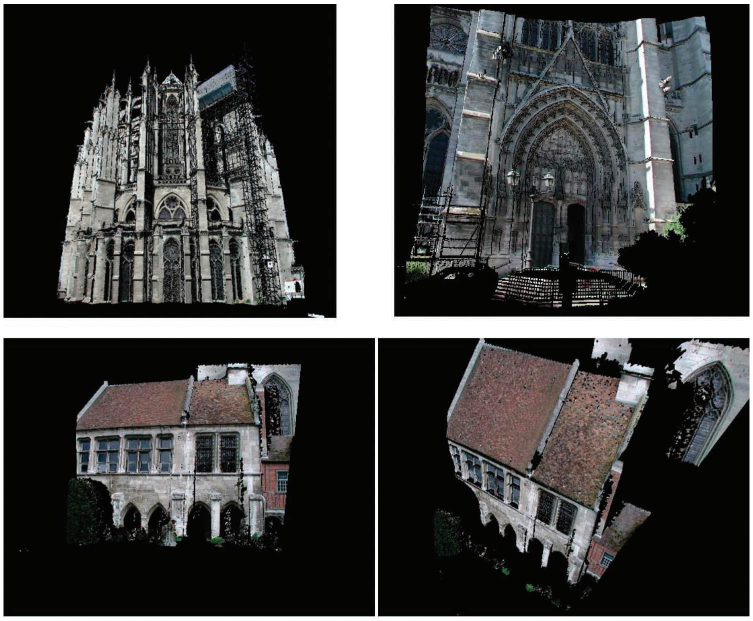
**8** Top row: Exterior model with 47 registered scans. Middle and bottom rows: Interior model (viewed from outside and inside) with 120 registered scans.

a visibility function $V(\boldsymbol{P}, T_i) \rightarrow 0, 1$ for each mesh triangle $T_i$ in the model. The function evaluates to 1 when all three vertices of $T_i$ are visible from the point of view of the camera described by $\boldsymbol{P}$ and 0 otherwise.

We can also compute the matrix $\boldsymbol{P}$ from a 3D and 2D line correspondence or a mixture of both, points and lines. We're currently working on computing $\boldsymbol{P}$ using line correspondences so that we can later make this process automatic following the ideas of the range-to-range registration described earlier. Figure 9 shows a textured mesh model of the cathedral from a number of views.

## Conclusions

As the cost of range scanning devices continues to decrease, we can expect to see the creation of more large-scale models. The tools and methods we've developed show promise in automating the registration task,



**9** **Texture-mapped views of the model.**

## Web Extras

Visit http://csdl.computer.org/comp/mags/cg/2003/06/g6toc.htm to view a video fly-through animation of the cathedral model mentioned in this article.

both for 3D to 3D and 2D to 3D. The major challenges are the extremely large size of each data set and the need to build complete models that integrate multiple views. The methods described here are helpful in a number of ways. First, the segmentation algorithm[9] lets us reduce the data set size and create linear features that can be efficiently matched for initial registration. Once these data sets are pairwise registered, we can find a globally consistent registration using a topological graph that minimizes error. We can then use the ICP algorithm to create a final registration. Using these registered data sets, we can create meshes for texture mapping and photorealistic viewing.

However, we still have some problems to solve. These include automating the data acquisition task, view planning to select the best viewpoints, and real-time model creation and visualization. We believe this is a rich research area, with application to virtual reality, telepresence, digital cinematography, digital archaeology, journalism, and urban planning. ■

## Acknowledgments

This work was supported in part by National Science Foundation grants IIS-01-21239, EIA-0215962, and Career IIS-0237878, the Samuel Kress Foundation, CUNY Institute of Software Design and Development, the Professional Staff Congress CUNY research award, and the Institution of Ioannis S. Latsis.

## References

1. F. Bernardini et al., "Building a Digital Model of Michelangelo's Florentine Pieta," *IEEE Computer Graphics and Applications*, vol. 22, no. 1, Jan./Feb. 2002, pp. 59-67.
2. M. Levoy et al., "The Digital Michelangelo Project: 3D Scanning of Large Statues," *Proc. Siggraph*, ACM Press, 2000, pp. 131-144.
3. P.E. Debevec, C.J. Taylor, and J. Malik, "Modeling and Rendering Architecture from Photographs: A Hybrid Geometry-Based and Image-Based Approach," *Proc. Siggraph*, ACM Press, 1996, pp. 11-20.
4. K. Ikeuchi and Y. Sato, *Modeling from Reality*, Kluwer Academic Press, 2001.
5. K. Nuyts et al., "Vision on Conservation," *Proc. Int'l Symp. Virtual and Augmented Architecture* (VAA 01), B. Fischer, K. Dawson-Howe, and C. O'Sullivan, eds., Springer, 2001, pp. 125-132.
6. J. Gregor and R. Whitaker, "Indoor Scene Reconstruction from Sets of Noisy Images," *Graphical Models*, vol. 63, no. 5, 2002, pp. 304-332.
7. I. Stamos and M. Leordeanu, "Automated Feature-Based Range Registration of Urban Scenes of Large Scale," *Proc. IEEE Int'l Conf. Computer Vision and Pattern Recognition*, vol. II, IEEE CS Press 2003, pp. 555-561.
8. P.J. Besl and N.D. Mckay, "A Method for Registration of 3-D Shapes," *IEEE Trans. Pattern Analysis and Machine Intelligence*, vol. 14, no. 2, Feb. 1992, pp. 239-256.
9. I. Stamos and P.K. Allen, "Geometry and Texture Recovery of Scenes of Large Scale," *Computer Vision and Image Understanding* (CVIU), vol. 88, no. 2, Nov. 2002, pp. 94-118.
10. O. Faugeras, *Three-Dimensional Computer Vision*, MIT Press, 1996.
11. K. Nishino and K. Ikeuchi, "Robust Simultaneous Registration of Multiple Range Images," *Proc. 5th Asian Conf. Computer Vision*, 2002, pp. 454-461.

***Peter K. Allen*** *is a professor of computer science and director of the Robotics Laboratory at Columbia University. His research interests include real-time computer vision, dexterous robotic hands, 3D modeling, and sensor planning. Allen received a PhD in computer science from the University of Pennsylvania, an MS in computer science from the University of Oregon, and an AB from Brown University in mathematics-economics. The National Science Foundation has named Allen a Presidential Young Investigator.*

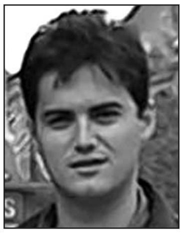
***Ioannis Stamos*** *is an assistant professor in the computer science departments of Hunter College and Graduate Center of the City University of New York. His main research interest is the photorealistic 3D model acquisition of large-scale scenes. Stamos received an MS, M.Phil, and PhD from Columbia Uni-*

*versity in computer science, and a Diploma of Engineering from the University of Patras, Greece in computer engineering and informatics. He is a recipient of the Faculty Early Career Development Award from the National Science Foundation.*

***Alejandro Troccoli*** *is a PhD student in the computer science department at Columbia University. His research interests include 3D modeling, computer vision, and computer graphics. Troccoli received a BS and a Licentiate degree (MS) in computer science from the Universidad de Buenos Aires, Argentina.*

***Benjamin Smith*** *is a master's student in the computer science department at Columbia University. His research interests include 3D modeling, editing, and visualization. Smith received a BS in computer science from Columbia University.*

For further information on this or any other computing topic, please visit our Digital Library at http://computer.org/publications/dlib.



***Marius Leordeanu*** *is a PhD student at the Robotics Institute of Carnegie Mellon University. His research interests include computer vision and graphics and artificial intelligence. Leordeanu received a BA in computer science and mathematics from Hunter College, New York.*

***Stephen Murray*** *is director of the Media Center for Art History, Archaeology, and Historic Preservation at Columbia University. His research interests include medieval sermons, storytelling in Gothic, and the Romanesque architecture of the Bourbonnais. He is currently engaged in projecting his cathedral studies through the electronic media using a combination of 3D simulation, digital imaging, and video. Murray was educated at Oxford and the Courtauld Institute of Art, University of London, where he received a PhD in art history.*

*Readers may contact Peter Allen at the Dept. of Computer Science, Columbia Univ., 500 W. 120th St., New York, NY 10027; allen@cs.columbia.edu.*