New Methods for Digital Modeling of Historic Sites Using Range and Image Data *

Peter Allen, Ioannis Stamos†, Alejandro Troccoli, Benjamin Smith
Marius Leordeanu†and Stephen Murray‡
Dept. of Computer Science, Columbia University
†Dept. of Computer Science, Hunter College, CUNY
‡Dept. of Art History and Archaeology, Columbia University

Abstract

Preserving cultural heritage and historic sites is an important problem. These sites are subject to erosion, vandalism, and as long-lived artifacts, they have gone through many phases of construction, damage and repair. It is important to keep an accurate record of these sites using 3-D model building technology as they currently are, 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 3-D models is time consuming and difficult, usually involving much manual effort. This paper discusses new methods that can reduce the time to build a model using automatic methods. The method utilizes range image segmentation and feature extraction algorithms. Our algorithm automatically computes pairwise registrations between individual scans, builds a topological graph, and places the scans in the same frame of reference. The methods can be extended to automate the texture mapping process as well, to create both geometric and photometric realistic models. Examples of these methods are shown in reconstructing a model of the Cathedral Saint-Pierre in Beauvais, France.

1 Introduction

Preserving cultural heritage and historic sites is an important problem. These sites are subject to erosion, vandalism, and as long-lived artifacts, they have gone through many phases of construction, damage and repair. It is important to keep an accurate record of these sites using 3-D model building technology as they currently are, 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 3-D models is time consuming and difficult, usually involving much manual effort. Recently, the advent of new 3–D range scanning devices has provided new means to preserve these sites digitally, and to preserve the historic record by building accurate geometric and photorealistic 3–D models of these sites. 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 created by these devices. This reinforces the need for automatic methods of registering, merging and abstracting the dense range data sets.

A number of other projects have addressed this and similar problems including [1, 7, 3, 6, 9, 5]. Each of these projects differs in the way models are created and the amount of human interaction in the process. Our work is centered on developing new methods to recover complete geometric and photometric models of large sites and to automate this process. We are developing new methods for data abstraction and compression through segmentation, 3–D to 3–D registration (both coarse and fine), 2–D to 3–D texture mapping of the models with imagery, and robotic automation of the sensing task. The methods we have developed are also suitable for a variety of other applications related to large-scale model building.

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

*This work was supported in part by NSF grants IIS-01-21239, EIA-0215962, and IIS-0237878, The Samuel Kress Foundation, CUNY Institute of Software Design and Development, and PSC-CUNY Research Award.

- Establish a baseline model for the current structural condition of the Cathedral
- Create a geometrically accurate 3-D model to examine weaknesses in the building and propose remedies
- Visualize the building in previous contexts as an educational tool

Sidebar I: The Cathedral Saint-Pierre

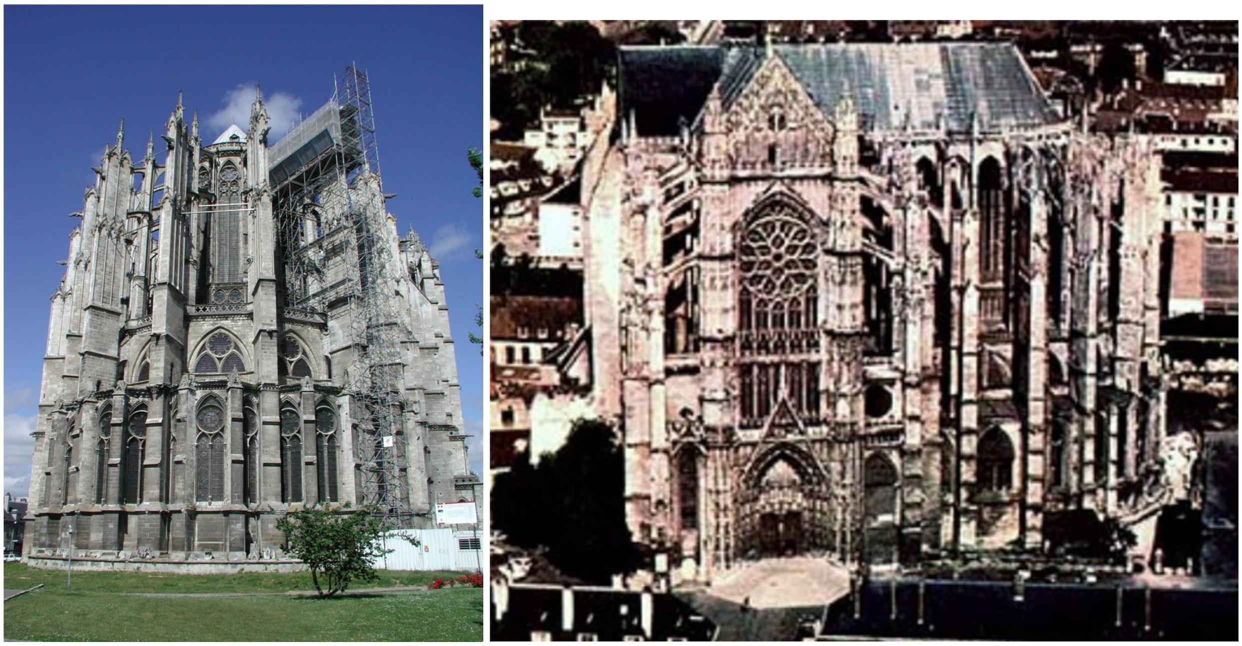

Figure 1: The Cathedral Saint-Pierre.

The Cathedral Saint-Pierre in Beauvais, France is a unique and still used church which is a prime example of high gothic architecture. A team of Columbia University architectural historians, computer scientists, and engineers has begun to study the fragile structure of the tallest medieval cathedral in France. This thirteenth-century Gothic cathedral collapsed twice in the Middle Ages and continues to pose problems for 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 already weakened roof timbers to shift. 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 the cathedral has been intensively studied, 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), Beauvais Cathedral, commissioned in 1225 by Bishop Milon de Nanteuil, provides an extreme expression of the Gothic enterprise. Only the magnificent choir and transepts were completed; the area where the nave and façade 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-fourteenth 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-sixteenth 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 remarkably 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.

<u>End of Sidebar I</u>

2 Registration Methods

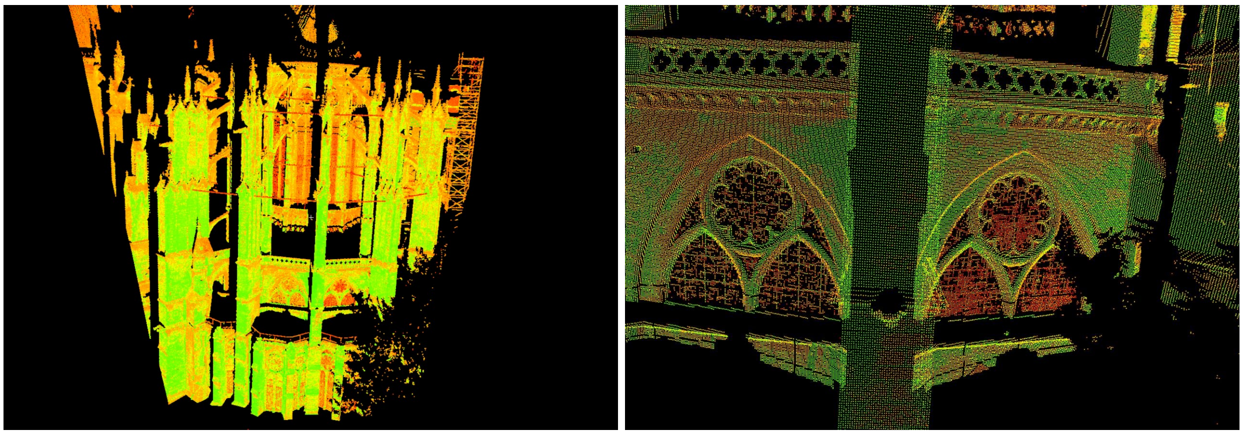

Figure 2: Left: Single dense range scan of Cathedral, approx. 1 million points. Right: detail of scan

To create data-sets that can be turned into models, we are using a time-of-flight laser scanner (Cyrax 2500) to measure the distance to points on an object. Data from the scanner comprises point clouds, 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 RSV (Reflectance Strength Value) is a function of distance to the scanned surface, angle of the surface relative to the laser beam direction, and material properties of the surface. A scan of 1000×1000 points takes about 10 minutes.

In order to acquire data describing an entire structure such as the Cathedral, multiple range scans must be taken from different locations which must be registered together correctly. Although the point clouds may be registered manually, it is very time consuming and error-prone. Each range scan can provide up to 1 million data points (see Fig. 2), and manually visualizing millions of small points and matching them is quite imprecise and difficult as the number of scans increases. When possible, it is a common practice to use specially designed targets/fiducials to help during the registration phase. In many cases, such as with the Cathedral, it is difficult to place targets, requiring us to develop our automatic registration methods.

Our registration method is a three step process [11]. The first step is an automatic pairwise registration between two overlapping scans. The pairwise registration matches 3-D line segments extracted from overlapping range scans to compute the correct transformation. The second step is a global registration step that tries to align all the scans together using overlapping pairs. The third step is a multi-image simultaneous ICP (Iterative Closest Point) algorithm [2] that does the final fine registration of the entire data-set. We describe each of these processes below.

We first use a range segmentation algorithm we have developed [10] to automatically extract planar regions from the point clouds. Once we have these planar features, we can create a set of linear 3–D features at the borders of each planar segment. Thus, a 3–D range scan is converted to a set of bounded planes and a set of finite lines (see Fig. 3). By matching these lines we can find the correct registration between scans.

After the segmentation step, the following elements are extracted from the point clouds: planar regions $\mathbf{P}$, outer and inner borders of those planar regions $\mathbf{B_{out}}$ and $\mathbf{B_{in}}$, outer and inner 3–D border lines $\mathbf{L_{in}}$ and $\mathbf{L_{out}}$ (defining the borders of the planar regions), and 3–D lines of intersection $\mathbf{I}$ 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

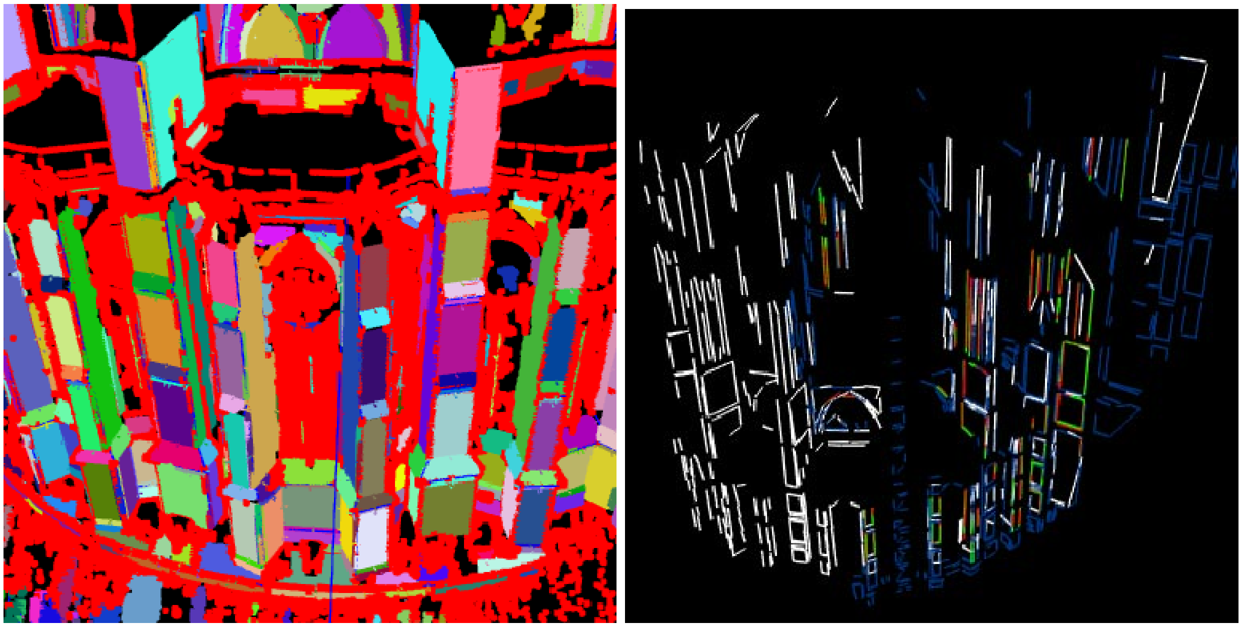

Figure 3: Left: Segmentation detail in a scan of the Cathedral data-set. Different planes are represented with different colors. The red color corresponds to the unsegmented (non-planar) parts of the scene. Right: Registration of lines between a pair of overlapping scans (Cathedral). left lines (white), right lines (blue), matched lines (red and green). Viewing note: due to the overlap of the matched lines some times only the red or the green component dominates.

Π on which they lie. That is, each border line has an associated line direction, and an associated supporting plane Π. In more detail, we represent each border line as a tuple $(\mathbf{p_{start}}, \mathbf{p_{end}}, \mathbf{p_{id}}, \mathbf{n}, \mathbf{p_{size}})$, where $\mathbf{p_{id}}$ is a unique identifier of its supporting plane Π, $\mathbf{n}$ is the normal of Π, and $\mathbf{p_{size}}$ is the size of Π. We estimate the size of the planes by using their number of range points on the plane, the computed distance of the plane from the origin of the coordinate system and by the plane normal. The additional information associated with each line greatly helps the automated registration.

2.1 Pairwise Registration

To automatically register a pair of overlapping range scans S_1 and S_2 we need to solve for rotation matrix R and translation vector $\mathbf{T} = [T_x, T_y, T_z]^T$ that place the two scans in the same coordinate system. The flowchart of the algorithm is shown in fig. 4. The features extracted by the segmentation algorithm are automatically matched and verified in order 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 $\mathbf{l}$ is represented with the tuple $(\mathbf{p_{start}}, \mathbf{p_{end}}, \mathbf{p_{id}}, \mathbf{n}, \mathbf{p_{size}})$, and each right line $\mathbf{r}$ with the tuple $(\mathbf{p'_{start}}, \mathbf{p'_{end}}, \mathbf{p'_{id}}, \mathbf{n}', \mathbf{p'_{size}})$. The algorithm works as follows.

1. At a preprocessing step pairs whose ratios of lengths and plane sizes $\mathbf{p_{size}}, \mathbf{p'_{size}}$ is smaller than a threshold are filtered out. Even though the overlapping parts of the two scans are not acquired identically by the scanner (because of occlusion and noise), the data was accurate enough for the extracted matching lines to have similar lengths and positions and matching planes similar sizes. After some experimentation we were able to find thresholds that worked on all pairs of scans, giving results of similar quality.

2. Sort all possible pairs of left and right lines $(\mathbf{l}, \mathbf{r})$ in lexicographic order.

3. *STAGE 1:* Get the next not visited pair of lines $(\mathbf{l_1}, \mathbf{r_1})$. Compute the rotation matrix R, and an estimation of the translation T_{est} assuming the match $(\mathbf{l_1}, \mathbf{r_1})$. Each line is a boundary segment of an

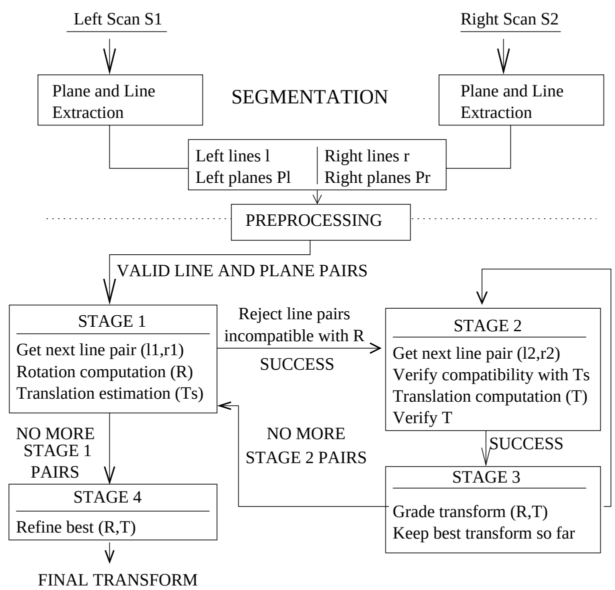

Figure 4: Flowchart for automatic registration between a pair of overlapping range scans.

associated plane. Hence, we can also use the information about the planes (i.e. their normals) to create enough constraints to fully determine the rotation matrix.

4. Apply the computed rotation R to all pairs $(\mathbf{l}, \mathbf{r})$ with $\mathbf{l} > \mathbf{l_1}$. Reject all line pairs whose directions and associated plane normals do not match after the rotation. If the number of remaining matches is less than the current maximum number of matches, go to *STAGE 1*. Otherwise accept the match between lines $(\mathbf{l_1}, \mathbf{r_1})$ and their associated planes.

5. *STAGE 2:* Get the next pair $(\mathbf{l_2}, \mathbf{r_2})$ from the remaining pairs of lines. Reject the match if it is not compatible with the estimated translation T_{est}. Compute an exact translation T from the two pairs $(\mathbf{l_1}, \mathbf{r_1})$ and $(\mathbf{l_2}, \mathbf{r_2})$. Verify that the two line pairs and their associated plane normals are in correspondence after the translation T is applied. Accept $(\mathbf{l_2}, \mathbf{r_2})$ as the second match.

6. *STAGE 3:* Grade 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 are in correspondence. Go to *STAGE 1*.

7. *STAGE 4:* After all possible combinations of valid pairs have been exhausted, the best computed transform (R, T) is recomputed 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 $g(R, T)$ that equals the total number of line matches after the transformation is applied. Note that the grade is small if there is no overlap between the scans.

Generally, a solution to the problem is possible if two pairs of matched lines are found between the two scans S_1 and S_2. Only the orientation and position of the lines is used due to the fact the endpoints can never be exactly localized (this is an inherent problem of all line detectors). Using these two matched pairs, a closed-form formula [4] provides the desired transformation (R,T). That means that a blind hypothesis-and-test approach would have to consider all possible $\begin{pmatrix} N \\ 2 \end{pmatrix} \times \begin{pmatrix} M \\ 2 \end{pmatrix} = O(M^2 N^2)$ 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 would 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 only a fraction of this search space needs to be considered. Results are presented in section 3.

2.2 Global Registration

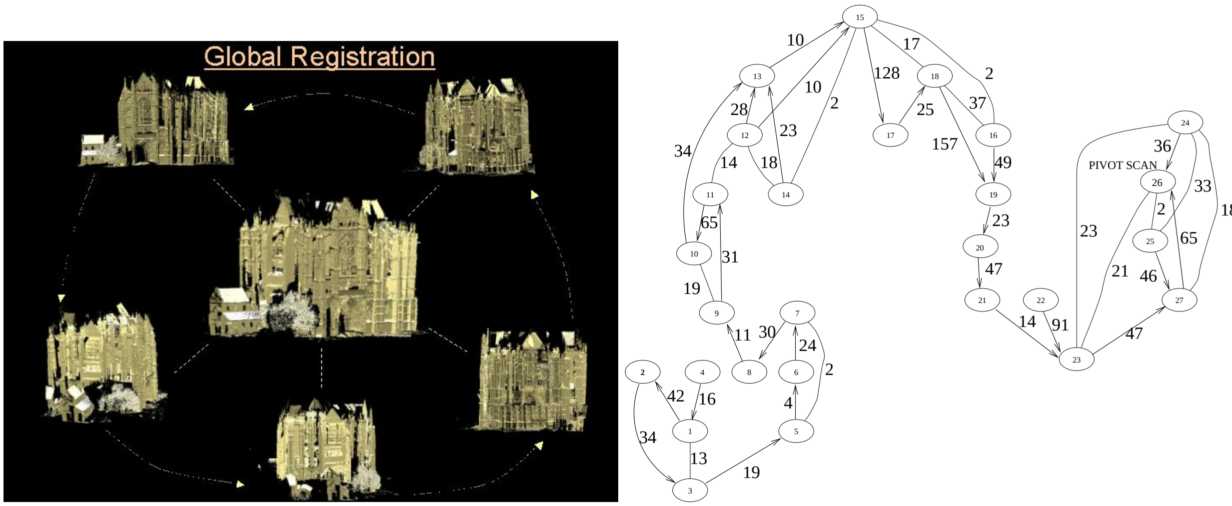

Figure 5: Left: Global registration problem. We need to correctly register all scans of the Cathedral. Right: Graph of 27 registered scans of the Cathedral data-set. The nodes correspond to the individual range scans. The edges show pairwise registrations. The weights on the edges show the number of matched lines that the pairwise registration algorithm provides. The directed edges show the paths from each scan to the pivot scan that is used as an anchor.

In a typical scanning session tens or hundreds of range scans need to be registered. Calculating all possible pairwise registrations is impractical because it leads to a combinatorial explosion. In our system, the user provides a list of overlapping pairs of scans. This reduces the number of possible pairings and from this reduced list all pairwise transformations are computed. Then, one of the scans is chosen 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 have the ability to reject paths of pairwise transformation that contain registrations of lower confidence.

In more detail, the rigid transformations (R_i, T_i) and their associated grades $g(R_i, T_i)$ are computed between each pair of overlapping scans. In this manner a weighted undirected graph is constructed. The nodes of the graph are the individual scans, and the edges are the transformations between scans. Finally the grades $g(R_i, T_i)$ 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 in order 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 utilizes all possible paths of pairwise registrations between S and S_a in order 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.

2.3 Simultaneous Registration of Multiple Range Images

Once the range images are registered using the automatic method above, a refinement of the basic ICP algorithm to simultaneous registration of multiple range images is used to provide the final registration. This method is an extension of the method proposed by Nishino and Ikeuchi [8]. Their work extends the basic pair-wise ICP algorithm to simultaneous registration of multiple range images. An error function is designed to be minimized globally against all range images. The error function utilizes an estimator that robustly rejects the outliers and can be minimized efficiently using a conjugate gradient search framework. To speed up the registration process, a *K-D* tree structure is employed so that the search time for the matched point is reduced. 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 laser reflectance strength value (RSV) is used. The idea is that points that are close will have similar RSV values. To find a best match point of a model point, multiple closest points in th K-D tree are searched. Then the RSV distance (to the model point) for each of them is evaluated to get the closest point. Once correspondences are made, we can iteratively find the the correct transformation matrices for the data points.

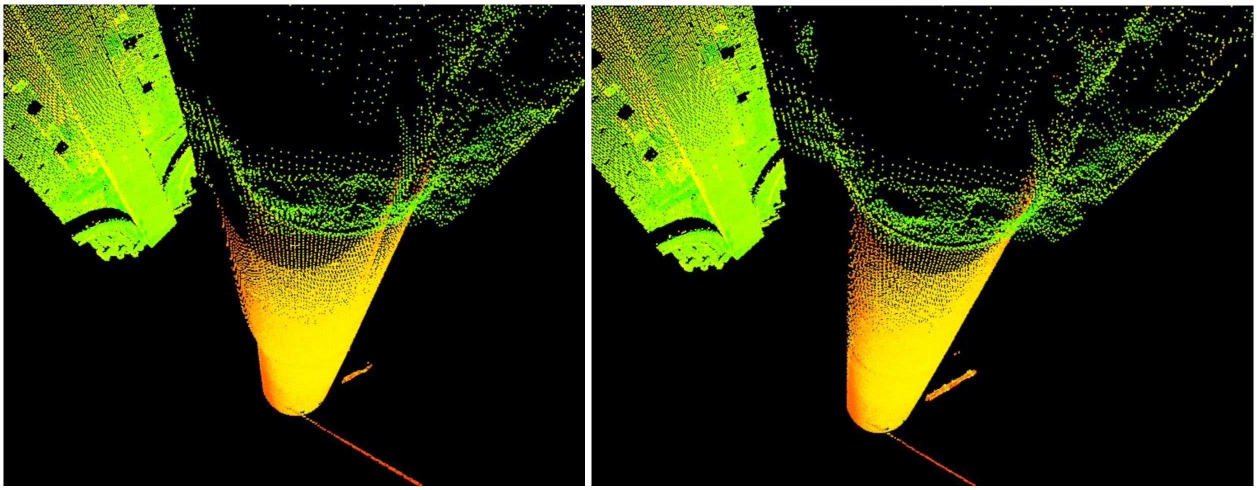

Figure 6: Before and after fine registration. Note column misalignment in left image has been corrected.

The data-set for the Cathedral contains over 100 scans, and it requires significant computational resources and time to register these scans with full resolution; therefore, those scans are sub-sampled down to 1/25 of their original resolution to speed up the registration process. Figure 6 shows the results of applying the algorithm on 2 scans that have been coarsely registered. The column, which is misaligned initially, is correctly aligned after the procedure.

3 Registration Results

Our methods were first tested 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 6mm along the laser-beam direction at a distance of 50m from the scanner. Each scan was first segmented and major planes and lines were extracted from it. The pairwise registration algorithm was executed on pairs of overlapping scans. The final step was the global registration algorithm. Results from the campus building are shown in figure 7. In this figure, 10 range scans have been automatically registered. Figure 8 shows registration for 27 scans of the Cathedral. The left image shows the registration, the relationship of each scanner position in the registered model, and segments that were used in the registration process which are colored. The right image is a closeup of the registration around the central window, showing the accuracy of the alignment.

Table 1 provides an extensive evaluation of the efficiency and accuracy of our algorithm. The efficiency of the algorithm is demonstrated by the percentage of line pairs that survive after preprocessing, and that reach *STAGE 2*, and *STAGE 3* of the algorithm. Very few lines need to be considered at the expensive *STAGE 3*. The running times range from 3 to 52 seconds (2GHz Linux machine) per pair, depending on the input size and on the amount of overlap. The measured pairwise registration error is also shown. This error is the average distance between matched planes lying on the surface of the scans. The error ranges from 1.36mm to 14.96mm for the Campus data-set and from 5.34mm to 56.08mm for the Cathedral. The average error over all ten scans of the Campus data-set is 7.4mm and over all twenty-seven scans of the Cathedral data-set 17.3mm.

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 are due to the lower spatial resolution of the scans (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 entirely adequate for our modeling task, and finer registration can be accomplished through ICP.

Campus Building - Results (average error 7.4mm)							
Pair	Line Pairs	Pre (%)	S2 % (#)	S3 % (#)	Matches	t (sec)	Error
1	301×303	16	1.7 (1555)	0.38 (346)	35	15	10.99mm
2	303×290	17	2.8 (2429)	0.84 (735)	25	29	6.28mm
3	290×317	21	2.8 (2572)	1.88 (1728)	36	52	2.77mm
4	317×180	19	3.4 (1955)	1.15 (656)	28	21	14.96mm
5	211×180	21	4.6 (1759)	2.1 (802)	31	19	9.26mm
6	180×274	17	2.6 (1306)	0.34 (168)	22	9	11.42mm
7	114×274	19	1.6 (507)	2.2 (894)	33	6	5.61mm
8	274×138	16	1.8 (667)	1.5 (557)	31	5	3.08mm
9	114×138	18	2.7 (423)	3.8 (593)	32	4	3.94mm
10	138×247	18	2.3 (791)	1.3 (429)	20	5	1.36mm
Cathedral - Results (average error 17.3mm)							
1	406×464	7	0.9 (1650)	0.3 (615)	42	39	9.37mm
2	464×269	7	0.7 (888)	0.3 (443)	34	16	16.9mm
3	406×269	11	0.7 (794)	0.1 (104)	13	9	56.08mm
4	151×406	21	1.1 (668)	0.8 (480)	16	7	5.34mm
5	269×387	11	0.7 (702)	0.4 (369)	19	9	15.8mm
6	326×197	10	0.9 (597)	0.1 (49)	24	4	11.68mm
7	197×143	15	1.0 (290)	0.3 (82)	30	3	6.44mm
8	143×194	16	1.9 (520)	0.1 (31)	11	3	29.24mm
9	194×356	15	2.0 (1429)	0.1 (93)	19	11	30.82mm

Table 1: Evaluation of the performance of the algorithm. A set of pairwise registrations is shown. Each row represents one registered pair of scans. The second column displays the number of line pairs. Column *Pre* shows the % (over all possible pairs) of line pairs that need to be considered after the preprocessing step of the algorithm. Column *S2* shows the % (over all possible combinations) and total number of pairs that reach *STAGE 2* and column *S3* the same number for *STAGE 3*, the most expensive stage (in *S3* the reduction is computed over all possible pairs of matches $((\mathbf{l_1}, \mathbf{r_1})$ and $(\mathbf{l_2}, \mathbf{r_2}))$). The efficiency of our algorithm lies on the great reduction of the pairs that need to be considered in this stage. Column *Matches* presents the number of matched pairs that the algorithm establishes. The running *Time* of the algorithm is shown for every pair (2GHz Linux machine). Finally, the pairwise registration *Error* is displayed. This *Error* is the average distance between matched planar region between the two scans. The error ranges from 1.36mm to 14.96mm for the first data-set and from 5.34mm to 56.08mm for the second.

The algorithm will fail in some cases. If very few linear features exist in the scene, matches are not possible. However, this is something we do not find in urban settings, which contain rich sets of linear features. Scene symmetry can also introduce false matches. This is not an inherent limitation of this particular algorithm, but a problem with all registration algorithms. Extra-constraints should be given in this case (i.e. the user can specify that the rotation - translation should be within certain limits, or the user can add extra point constraints).

A comprehensive model of the Cathedral has been constructed made up of data from all the scans. The resulting model is very large, and visualizing the entire model can be difficult. Fig. 9 show the model from a number of views. For these models, 120 scans were registered on the inside of the cathedral, and 47 on the outside. The outside scans were registered automatically except for a small number (7) where a single extra point constraint were added manually 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 3-D video fly-through animation of the model is available at the website: www.cs.columbia.edu/~allen/BEAUVAIS.

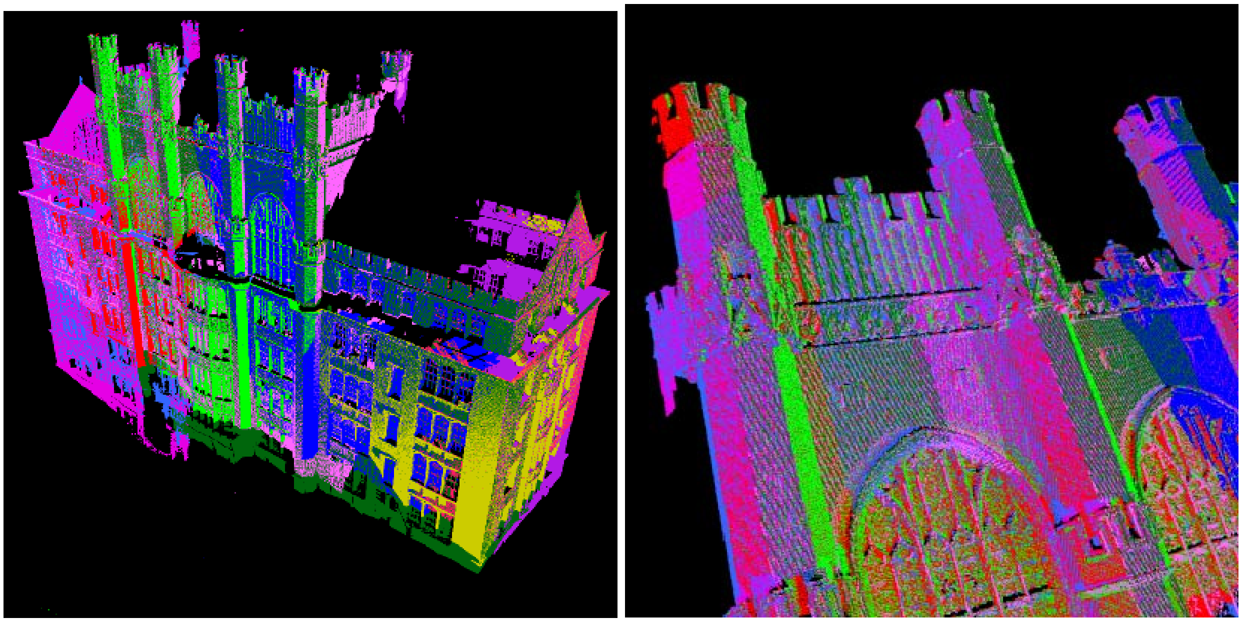

Figure 7: Campus site: a) (Left) Ten automatically registered scans. Each scan is represented with different color. b) Registration detail.

4 Texture Mapping

The range data allowed us to 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 2–D image. The triangular mesh is generated using the topology of the points acquired by the scanner, which provides a 2-D grid for each set of scanned 3–D points. Due to the size of the point sets we decimate them before building the mesh.

Once the mesh is built, a user manually selects a set of corresponding points from the point cloud and the 2–D image which are used to compute a projection matrix P that transforms world coordinates to image coordinates. Let $(\mathbf{X}_i, \mathbf{x}_i)$ be a pair of 3–D and 2–D 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,

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

where each P^i is a row of P. By stacking up the equations derived from a set of n pairs, a $2n \times 12$ matrix A is obtained. The solution vector p of the set of equations $Ap = 0$ contains the entries of the matrix P. At least 6 point correspondences are required to obtain a unique solution. In practice, an overdetermined system is used, which we solve using the SVD decomposition of matrix A. Prior to solving the system of equations, both 3–D and 2–D points are normalized to improve numerical stability. This normalization consists of a translation and scaling step; both 2–D and 3–D points are translated so that their centroid is at the origin and then scaled so that the RMS (root-mean-squared) distance to the new origin of the point sets is $\sqrt{2}$ and $\sqrt{3}$ respectively.

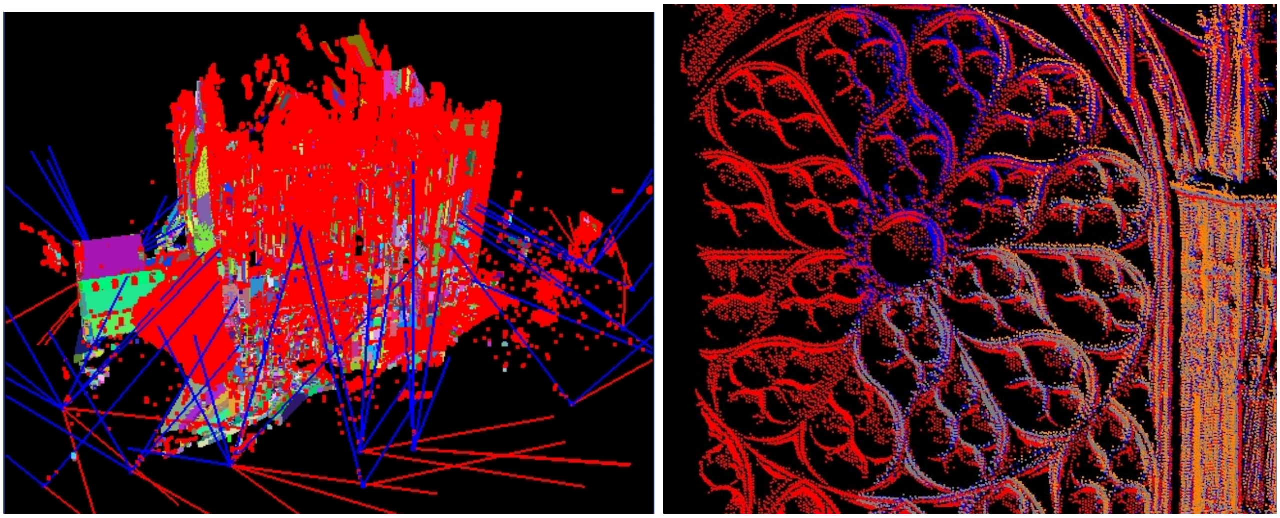

Figure 8: Cathedral site. Left: 3–D mesh after 27 scans has been placed in the same coordinate system. Segments used in the registration are shown colored. The local coordinate system of each individual scan is shown. The z-axis of each scan points towards the Cathedral. Right: Registration detail. Each scan is shown with different color. Note the accuracy of the alignment of points.

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

The matrix P can also be computed from 3–D and 2–D line correspondence or a mixture of both, points and lines. We are currently working on computing 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 10 shows a textured mesh model of the cathedral from a number of views.

5 Conclusions

As the cost of range scanning devices continues to decrease, we can expect to see more and more large-scale models of the type shown here being created. The tools and methods we have developed show promise in being able to automate the registration task, both for 3–D to 3–D and 2–D to 3–D. 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 allows us to 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. ICP can then be used to create a final registration. Using these registered data-sets, we can then create meshes for texture mapping and photorealistic viewing.

However, there are still open problems that need to be solved. 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 very rich research area, with application to Virtual Reality, Tele-presence, Digital Cinematography, Digital Archaeology, Journalism, and Urban Planning.

References

[1] F. Bernardini, I. Martin, J. Mittleman, H. Rushmeier, and G. Taubin. Building a digital model of Michelangelo's Florentine Pieta. *IEEE Computer Graphics and Applications*, 22(1):59–67, January 2002.

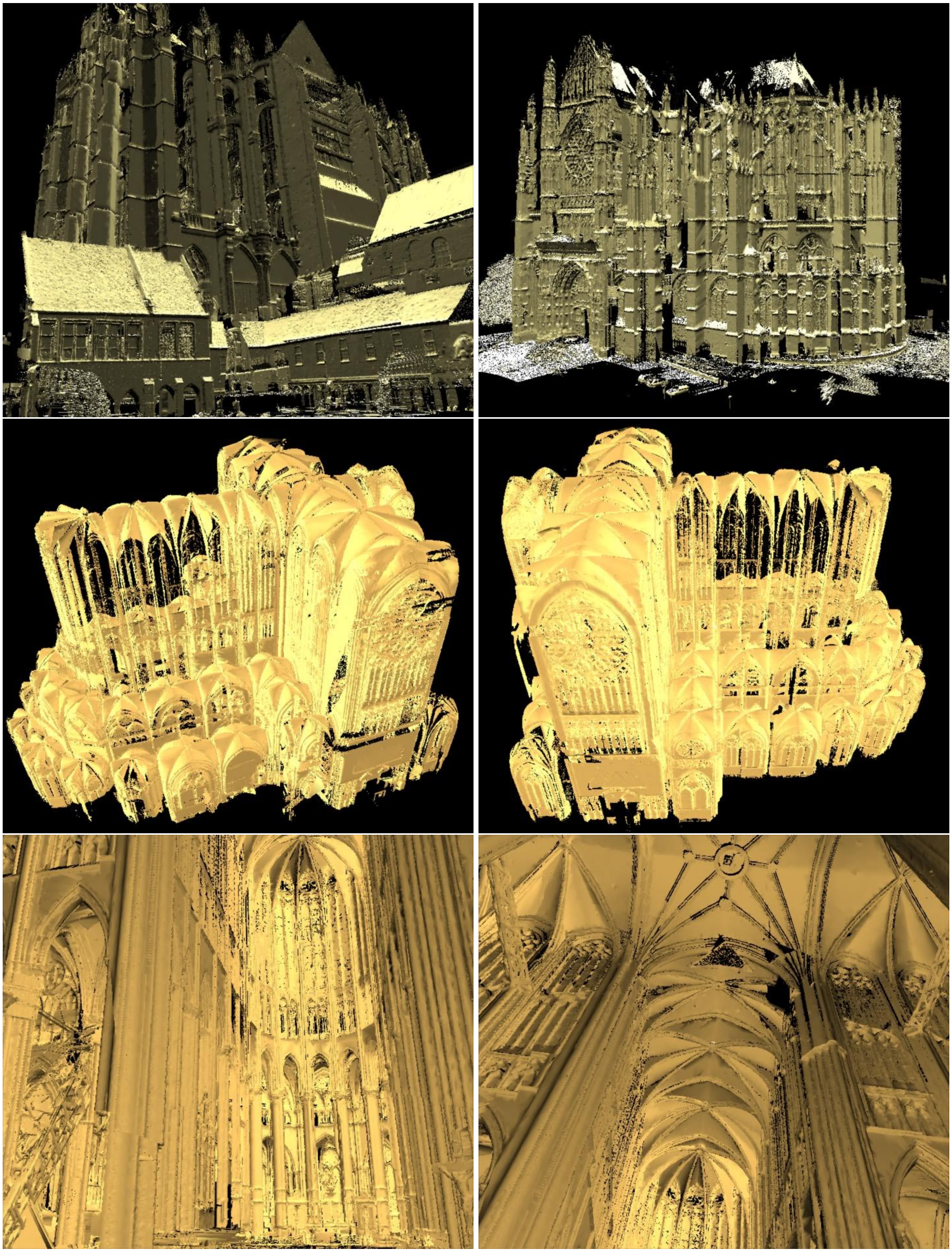

Figure 9: Row 1: Exterior model, 47 registered scans. Row 2-3: Interior model (viewed from outside and inside), 120 registered scans.

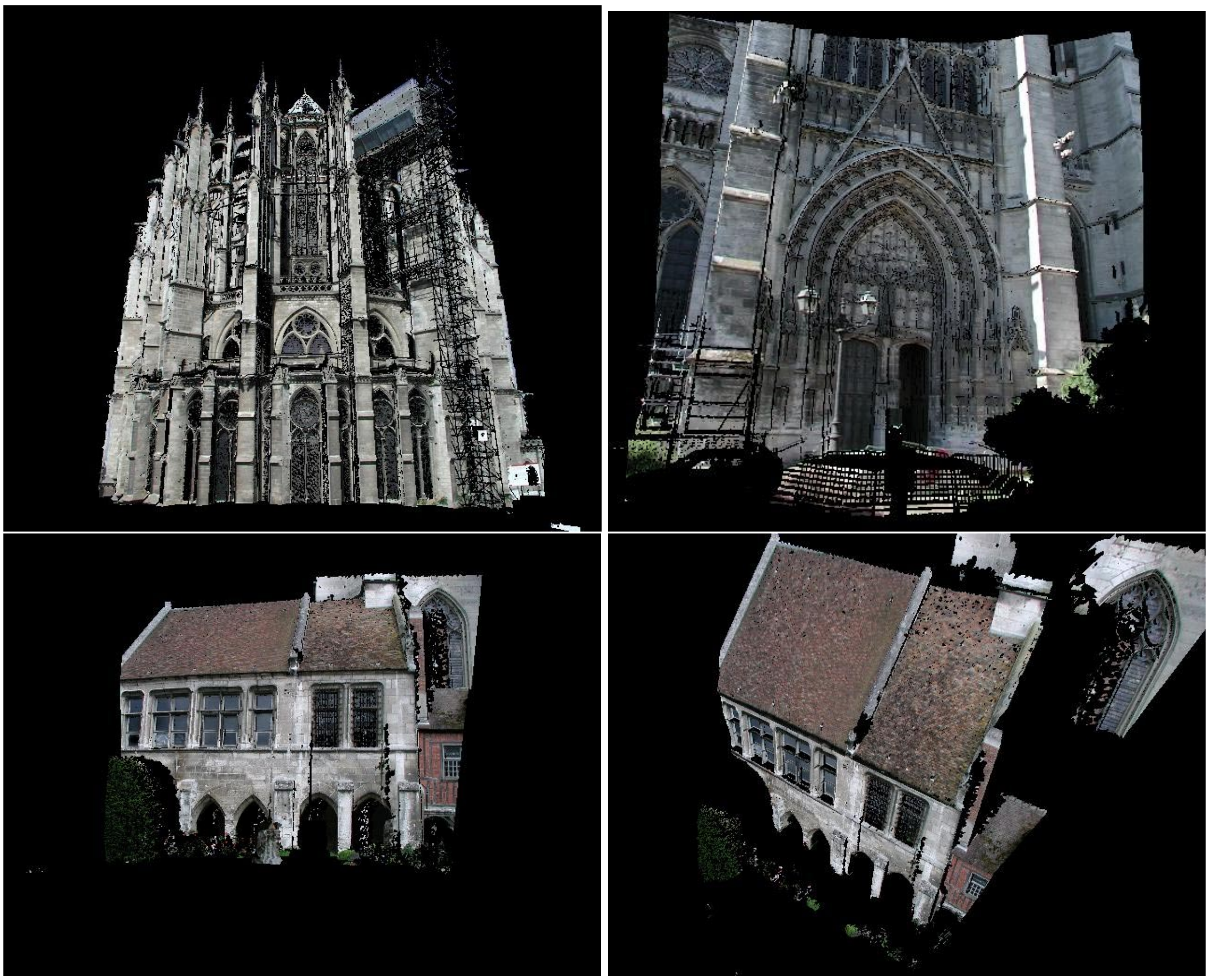

Figure 10: Texture mapped views of the model.

[2] P. J. Besl and N. D. Mckay. A method for registration of 3–D shapes. *IEEE Transactions on PAMI*, 14(2):239–256, February 1992.

[3] Paul E. Debevec, Camillo J. Taylor, and Jitendra Malik. Modeling and rendering architecure from photographs: A hybrid geometry-based and image-based approach. In *SIGGRAPH*, pages 11–20, 1996.

[4] Olivier Faugeras. *Three–Dimensional Computer Vision*. The MIT Press, 1996.

[5] J. Gregor and R. Whitaker. Indoor scene reconstruction from sets of noisy images. *Graphical Models*, (63):304–332, 2001.

[6] Katsushi Ikeuchi and Yoichi Sato. *Modeling from reality*. Kluwer Academic Press, 2001.

[7] Marc Levoy, Kari Pulli, Brian Curless, Szymon Rusinkiewicz, David Koller, Lucas Pereira, Matt Ginzton, Sean Anderson, James Davis, Jeremy Ginsberg, Jonathan Shade, and Duane Fulk. The Digital Michelangelo Project: 3D scanning of large statues. In *SIGGRAPH,* pages 131–144, 2000.

[8] K. Nishino. and K. Ikeuchi. Robust simultaneous registration of multiple range images. In *The 5th Asian Conference on Computer Vision, 2002*, pages 454–461, January 2002.

[9] Kris Nuyts, Jean-Pierre Kruth, Bert Lauwers, Herman Neuckermans, Marc Pollefeys, Li Qiongyan, Joris Schouteden, Pierre Smars, Koen Van Balen, Luc Van Gool, and Maarten Vergauwen. Vision on conservation. In Bob Fischer, Kenneth Dawson-Howe, and Carol O'Sullivan, editors, *International Symposium on Virtual and Augmented Architecture (VAA 01)*, pages 125–132. Springer, June 2001.

[10] Ioannis Stamos and Peter K. Allen. Geometry and texture recovery of scenes of large scale. *Computer Vision and Image Understanding (CVIU),* 88(2):94–118, November 2002.

[11] Ioannis Stamos and Marius Leordeanu. Automated feature-based range registration of urban scenes of large scale. In *IEEE International Conference of Computer Vision and Pattern Recognition*, volume II, pages 555–561, Madison, WI, June 2003.