Automated Feature-Based Range Registration of Urban Scenes of Large Scale *

Ioannis Stamos and Marius Leordeanu
Department of Computer Science, Hunter College, City University of New York, NY, NY 10021
{istamos,mleordeanu}@hunter.cuny.edu

Abstract

We are building a system that can automatically acquire 3D range scans and 2D images to build geometrically and photometrically correct 3D models of urban environments. A major bottleneck in the process is the automated registration of a large number of geometrically complex 3D range scans in a common frame of reference. In this paper we provide a novel method for the accurate and efficient registration of a large number of complex range scans. The method utilizes range 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. We present results for building large scale 3D models of historic sites and urban structures.

1 Introduction

The recovery and representation of 3D geometric and photometric information of the real world is one of the most challenging and well studied problems in Computer Vision and Robotics research. There is a clear need for highly realistic geometric models of the world for applications related to Virtual Reality, Tele-presence, Digital Cinematography, Digital Archeology, Journalism, and Urban Planning. Recently, there has been a large interest in reconstructing models of outdoor urban environments. The areas of interest include geometric and photorealistic reconstruction of individual buildings or large urban areas using a variety of acquisition methods and interpretation techniques, such as ground-base laser sensing, air-borne laser sensing, ground and air-borne image sensing.

A typical 3D modeling system involves the phases of 1) Individual range image acquisition from different viewpoints, 2) Registration of all images into a common frame of reference, 3) Transformation of each range image into an intermediate surface-based or volumetric-based representation, and 4) Merging of all range images into a common representation. Typical 3D modeling systems include the work by Allen [12], Levoy [3], the VIT group [16], and Bernardini [1]. In this paper we are presenting a novel range registration algorithm between N pairs of range images (second task of a 3D modeling system) that is automated, it does not assume any a-priori knowledge regarding the position of the sensors, it does not assume that the views are spatially close with respect to each other, and is suitable for large scale data sets. Our algorithm produces accurate results and is computationally efficient, as shown in the last section of this paper. The algorithm is integrated in a larger system that generates photorealistic geometrically correct 3D models [14].

Registering two range images when two sets of corresponding 3D points have been identified can be done using a quaternion-based non-linear optimization method as described in [6]. The automation of the process of establishing point correspondences was presented in [2]. This is the widely used Iterative Closest Point (ICP) algorithm, where the rigid transformation between two views is iteratively refined, while larger sets of corresponding points between views can be extracted after each refinement step. Variants of this method include the work by Turk [15] and Rusinkiewicz [13]. All ICP-type methods require the meshes to be spatially close with respect to each other in order for an initial set of closest point correspondence to be established. Global ICP-type methods that compute registrations between all acquired scans include the work of Pulli [11] and Nishino [10]. We can utilize those methods as a post-processing step after the execution of our algorithm, since we do not constrain the scans to be spatially close. Hebert [7] introduced the idea of spin-images, where the initial list of corresponding points is extracted by using a pose-invariant representation for the range images. Integration of both range and intensity information in the 3D registration process is described in [17, 8]. Finally, Lucchese [9] proposes a method based on 3D Fourier transform for the registration of 3D solids (texture information is also used).

Our method starts with a segmentation algorithm that extracts planar and linear features from the raw range scans (section 2). The reduction of the geometric complexity due to segmentation makes our method suitable for large scale registration problems [14]. Our novel pairwise registration algorithm is applied in order to register overlapping scans (section 3). The pairwise registration algorithm is exploring the joined feature-space of the two scans for pairs of features that generate a robust match, and the optimal such match is kept. Finally, the pairwise registrations generate a graph; the nodes are the individual scans and the edges are the transformations between the scans. A graph algorithm is used in order to compute the transformation between every scan in

*Supported in part by NSF MRI/RUI EIA-0215962 award, CUNY Institute of Software Design and Development award, and PSC-CUNY award. We would like to thank Columbia's University Robotics Lab and Prof. P. K. Allen for using their Cathedral data sets.

this graph and a scan which is called the pivot (section 4). Results of applying the algorithm for the reconstruction of a historic building and of an urban structure are presented in section 5.

2 Feature Extraction

The individual range-images which the range-sensor provides are the result of dense sampling of visible surfaces in large urban scenes. Using a Cyrax laser scanner [4], we get 1K by 1K range samples ($\sim$ one million range samples) with a spatial resolution of a few centimeters. Our previously developed range segmentation algorithm [14] automatically extracts planar regions and linear features at the areas of intersection of neighboring planar structures. Thus, a 3D range scan is converted to a set of bounded planes and a set of finite lines (see Fig. 6 for an example). In this paper we extend our initial line extraction algorithm, by extracting linear features at the internal and external boundaries of the recovered planar surfaces.

After the segmentation and line extraction step, the following elements are very accurately generated from the point clouds: planar regions $\mathbf{P}$, outer and inner borders of those planar regions $\mathbf{B_{out}}$ and $\mathbf{B_{in}}$, and outer and inner 3D border lines $\mathbf{L_{in}}$ and $\mathbf{L_{out}}$ (defining the borders of the planar regions). Border lines are represented by their two endpoints $(\mathbf{p_{start}}, \mathbf{p_{end}})$, and by the plane Π on which they lie. That is, each border line has an associated line direction, and an associated supporting plane Π (Fig. 2). In more detail, we represent each border line as a 5-tuple $(\mathbf{p_{start}}, \mathbf{p_{end}}, \mathbf{p_{id}}, \mathbf{n}, \mathbf{p_{size}})$, where $\mathbf{p_{id}}$ is a unique identifier of the line's 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 the number of range points on the plane, the computed distance of the plane from the origin of the coordinate system and the normal of the plane. The additional information associated with each line greatly helps the automated registration algorithm described in the following section.

3 Pairwise Registration

In this section we describe the automated registration between a pair of overlapping range scans S_1 and S_2. We are solving for the 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 [1]. 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 border lines with associated planes (see section 2). The flowchart of the algorithm is shown in Fig. 1, and it is explained in more detail below.

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

[1] If a point p is expressed in the coordinate system of scan S_1, then $p' = Rp + \mathbf{T}$ is the same point expressed in the coordinate system of scan S_2.

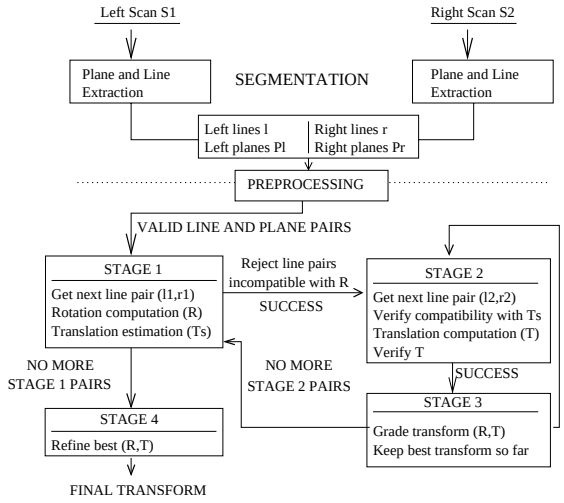

Figure 1: **Flowchart of the automated registration between a pair of overlapping scans.**

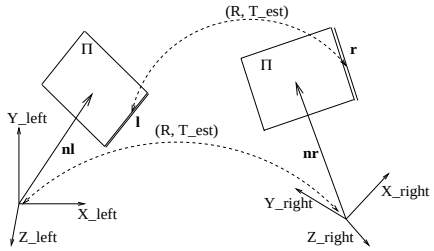

Figure 2: **Two matched *infinite* lines $(\mathbf{l}, \mathbf{r})$ provide an exact solution for the rotation R between the two coordinate systems. In general, one pair of lines is not enough for the rotation computation. We, however, take advantage of the fact that each line has an associated plane normal on which it lies. Also, by utilizing the actual endpoints of the lines we can compute an estimated translation T_{est}.**

these two matched pairs, a closed-form formula [5] provides the desired transformation $(R, \mathbf{T})$. 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 would need to compute the transformation $(R, \mathbf{T})$ 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. In order to avoid the combinatorial explosion caused by the *blind exhaustive search* we designed an algorithm that efficiently searches for a robust pair of line matches. For each 3D line the additional information of the supporting planes (section 2) on which it lies facilitates the search. Also, the length of the lines, and the size of the planes is used in order to discard invalid pairs at a preprocessing step, and in order to verify the quality of the match at later steps.

The rotational component of the transformation between the two views can be computed using the orientations $\mathbf{o_l^i}$ and $\mathbf{o_r^i}$ of the matched 3-D lines $(\mathbf{l_i}, \mathbf{r_i})$[5]. This is done via the minimization of the error function $Err(N) = \sum_{i=1}^{N} ||\mathbf{o_r^i} - R\mathbf{o_l^i}||^2$ where R is the unknown rotational matrix. The minimization of the above function has a closed-form solution when the rotation is expressed as a quaternion. The minimum number of correspondences for the computation of the rotation is two, i. e. $N = 2$. In our case only **one** pair of lines $(\mathbf{l}, \mathbf{r})$ from the left and right scan can produce a unique estimation for the rotation R, due to the utilization of the normals $(\mathbf{n_l}, \mathbf{n_r})$ of the two supporting planes (Fig. 2). Also, by using the endpoints of the lines we can produce an estimate $\mathbf{T_{est}}$ of the translational vector[2] as the vector that connects the mid-points of the two lines. It can be proved that this is the best approximation of the translation if we assume uniform errors in the locations of the endpoints of the lines. **Two pairs** of lines $(\mathbf{l_1}, \mathbf{r_1})$ and $(\mathbf{l_2}, \mathbf{r_2})$ on the other hand, determine uniquely not only the rotational but also the translational part of the transformation $(R, \mathbf{T})$, provided that such a transformation exists.

3.1 Algorithm

In more detail the algorithm works as follows (flowchart in Fig. 1). Let us call scan S_1, the left scan, and scan S_2 the right scan. Each left line $\mathbf{l}$ is represented by the 5-tuple $(\mathbf{p_{start}}, \mathbf{p_{end}}, \mathbf{p_{id}}, \mathbf{n}, \mathbf{p_{size}})$, and each right line $\mathbf{r}$ by the 5-tuple $(\mathbf{p'_{start}}, \mathbf{p'_{end}}, \mathbf{p'_{id}}, \mathbf{n'}, \mathbf{p'_{size}})$ (section 2). The central idea is the selection of a robust pair of lines in *STAGE 1* (computation of $(R, \mathbf{T_{est}})$) and of a second pair of lines in *STAGE 2* (computation of $(R, \mathbf{T})$) in order to evaluate the computed transformation in *STAGE 3* and to refine it in *STAGE 4*. The efficiency of the algorithm lies on the fact that a small number of pairs survive to *STAGE 3*. The necessary steps follow:

- Filter out, at a preprocessing step, pairs whose ratios of lengths and plane sizes $\mathbf{p_{size}}, \mathbf{p'_{size}}$ is smaller than a threshold. The overlapping parts of the two scans are not acquired identically by the scanner, due to occlusion and noise. The data though was accurate enough for the extracted matching lines to have similar lengths and positions and for the 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.

- Consider all remaining pairs of left and right lines $(\mathbf{l}, \mathbf{r})$ in lexicographical order. Initialize the current maximum number of line matches to $N_{max} = 0$ and the best transformation so far $(R_{best}, \mathbf{T_{best}})$ to identity.

- *STAGE 1:* Get the next pair of lines $(\mathbf{l_1}, \mathbf{r_1})$. Compute the rotation matrix R, and an estimation of the translation $\mathbf{T_{est}}$ assuming the match $(\mathbf{l_1}, \mathbf{r_1})$. A-priori information regarding the rotation R can be applied here in order to reject the match. Let M_{planes} be the set of pairs of matched planes. Make $M_{planes} = \emptyset$.

- Apply the computed rotation R to all pairs $(\mathbf{l}, \mathbf{r})$ with $\mathbf{l} > \mathbf{l_1}$ (note that the pairs are visited in lexicographic order). Reject all line pairs whose directions and associated plane normals do not match after the rotation, by using a fixed threshold $angle\_thresh$. If the number of remaining pairs (after the rejection) 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 of their associated planes. Update the set M_{planes} by including the two associated planes.

- *STAGE 2:* Get the next pair $(\mathbf{l_2}, \mathbf{r_2})$ from the remaining pairs of lines (a number of pairs were rejected in the previous step). Reject the match if it is not compatible with the estimated translation $\mathbf{T_{est}}$, using fixed thresholds $dist\_thresh$ and $endpts\_thresh$, and go back to *STAGE 2*. Otherwise, compute an exact translation $\mathbf{T}$ from the two pairs $(\mathbf{l_1}, \mathbf{r_1})$ and $(\mathbf{l_2}, \mathbf{r_2})$. Verify that the two line pairs are in correspondence, using fixed thresholds $dist\_thresh$ and $endpts\_thresh$, after the translation $\mathbf{T}$ is applied, and accept $(\mathbf{l_2}, \mathbf{r_2})$ as the second match. Update the set M_{planes} by including the planes of $(\mathbf{l_2}, \mathbf{r_2})$.

- *STAGE 3:* Grade the computed transformation $(R, \mathbf{T})$, by transforming all left lines to the coordinate system of the right scan, and by counting the number of valid pairs that are in correspondence, using fixed thresholds $dist\_thresh$ and $endpts\_thresh$. Each time a correspondence $(\mathbf{l}, \mathbf{r})$ is found, update the set M_{planes} by including the two associated planes. If the current number of line matches is greater than N_{max}, update N_{max} and $(R_{best}, \mathbf{T_{best}})$. Go to *STAGE 2* if there are more pairs to consider, or to *STAGE 1* if no more stage 2 pairs can be considered.

- *STAGE 4:* In this stage we transform all S_1 lines to the coordinate system of scan S_2 using $(R_{best}, \mathbf{T_{best}})$. We can thus find all possible pairs of line matches between S_1 and S_2, and recompute $(R_{best}, \mathbf{T_{best}})$ based on all these matches (a weighted least-squares algorithm is used [5, 14]). The thresholds used for matching lines are discussed in the next section. Now, we reiterate STAGE 4 using refined thresholds for a fixed number of times (see next section) in order to refine the transformation $(R_{best}, \mathbf{T_{best}})$.

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

[2] This is not an exact solution since the endpoints of 3D lines can not be exactly localized.

small if there is no overlap between the scans. In this case the grade is the result of spurious matches. The efficiency of the algorithm stems from the fact that a large number of line pairs is rejected before *STAGE 3* is reached. Evaluation of the performance of the algorithm is shown in section 5.

3.2 Algorithmic Details & Thresholds

Each time a pair of lines $(\mathbf{l}, \mathbf{r})$ is matched in *STAGE 1*, *STAGE 2*, and *STAGE 3* the associated planes are matched as well. A list of matched planes M_{planes} is maintained. The efficiency of the algorithm is increased by never allowing line matches if there is an inconsistency with the matching of their planes in previous stages [3].

A number of thresholds (for actual values see section 5) has to be used in order to decide if two lines or planes can be potential matches: (1) max_pl_ratio and max_ln_ratio: maximum ratio between the sizes of two matched planes and lines, (2) $dist\_thresh$: the maximum perpendicular distance allowed between two matched lines **l** and **r** after the transformation $(R, \mathbf{T})$ is applied, (3) $endpts\_thresh$: the maximum distance allowed between the endpoints of two matched lines **l** and **r** after the transformation $(R, \mathbf{T})$ is applied, and (4) $angle\_thresh$: the maximum angle between matched lines after the rotation R transformation has been applied. Thresholds (1) are used in the preprocessing stage, and thresholds (2) and (3) for the verification of the translation (STAGES 2, 3, and 4). Finally threshold (4) is used for the verification of the rotation (STAGES 1, 2, 3, and 4).

The selection of correct thresholds is an important issue. We attack this problem by allowing *large fixed* thresholds on the first three stages of the algorithm (*STAGE 1* to *STAGE 3*). That means that the search for the best transform $(R_{best}, \mathbf{T_{best}})$ becomes less vulnerable to the quality of the extracted lines sets. As a result, the experiments performed on the campus building and on the Cathedral used the same set of thresholds for all pairwise registered scans belonging to the same building. In STAGE 4 of the algorithm, where the best transformation is being refined, we adapt the $angle\_thresh$ to each pair of matches, we iteratively reduce the angle and distance thresholds by a constant factor c for a fixed number of steps, and we introduce weights for each pair of matches (see next paragraph).

Our experiments have shown us that there is a predictable bias in the computation of the orientation of the 3D lines depending on the orientation of the planes where the lines lie. The bigger the angle α_i between the plane Π_i and the scanning direction, the more probable is that the lines that lie on plane Π_i are biased (see Fig. 3). That means that these lines will bias the rotation estimation. Therefore, for every pair of lines checked for matching we adapt the angle threshold to be proportional to $\cos(\alpha_i) * \cos(\alpha_j)$, where angles α_i and α_j correspond to planes on which the two

[3] A line **l** with corresponding plane Π_i can not be matched with a line **r** and corresponding plane Π_j if the plane Π_i has been previously matched with a plane Π_k and plane Π_j has been previously matched with a plane Π_w, with $k \neq w$.

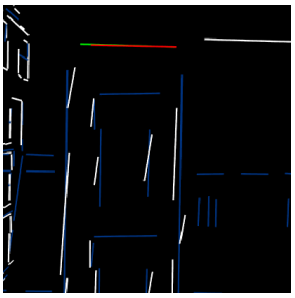

Figure 3: **Bias in line estimation. See the skewed orientation of the white lines (they belong to a slanted plane with respect the scanning direction) and the vertical orientation of the blue lines (they belong to a plane almost perpendicular to the viewing direction).**

matched lines lie. Also, as we mentioned before, the angle and distance thresholds are reduced at every iterative step of STAGE 4 by a constant factor c. So, the angle thresholds used for matching line pairs are adapted to each individual pair according to the formula $angle\_thresh(n) = angle\_thresh(0) * c^n * \cos(\alpha_i) * \cos(\alpha_j)$, where n is the refinement iteration in STAGE 4 and c is a constant factor lying in the interval $(0, 1)$. The distance thresholds are just reduced and are not adapted to each individual pair: $dist\_thresh(n) = dist\_thresh(0) * c^n$. Finally in the rotation computation, we adapt the weight of matched lines to $W\_lines(i, j) = K * \cos(\alpha_i) * \cos(\alpha_j) * (length_i + length_j)$, where K is a constant, in order to reduce the effect of the bias. We consider the weight of the plane normals of matched planes (Π_i, Π_j) (planes are not biased) to be $W\_plane(i, j) = L * (size_i + size_j)$, where L is a constant. These heuristic optimizations increase the quality of the output transformations by almost an order of magnitude (from cm to mm accuracy).

4 Global Registration between Pairs of Scans

In a typical scanning session tens or hundreds of range scans need to be registered. Our efficient pairwise registration algorithm (section 3) is executed for pairs of overlapping scans. In our system, the user is providing a list of overlapping pairs[4]. After all pairwise transformations from the list of overlapping pairs are computed 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 $(R_i, \mathbf{T_i})$ 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 (see Fig. 4 for an example). The nodes of the graph are the individual scans, and the edges are the transformations between scans. Finally the grades

[4] The list does not have to be a complete enumeration of all possible overlaps.

$g(R_i, T_i)$ are the weights associated with each edge (Fig. 4). 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. Let p_1 and p_2 be two different paths from S to S_a. We call p_1 to be 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 global registration between scans. Note that our method is not a global optimization technique that solves the correspondence problem for all scans simultaneously.

5 Results and Conclusions

The algorithm was tested on real data gathered from the famous Gothic Cathedral of Ste. Piérre in Beauvais (twenty-seven scans used, each consisting of one million 3D points) and from the Thomas Hunter building in our campus (ten scans used, each consisting of one million 3D points). The scans were gathered by a Cyrax 2500 laser range-scanner [4]. The accuracy of a point is 6mm along the laser-beam direction at a distance of 50m from the scanner. Each scan was first segmented (see section 2) and major planes and lines were extracted from it. The pairwise registration algorithm was executed (section 3) on pairs of overlapping scans. The final step is the global registration algorithm (section 4). A segmentation and pairwise registration between line-sets is shown in Figs. 5f and 6. The thresholds used (section 3.2) for all pairs of the Cathedral (campus) data set are: $max\_pl\_ratio = 2.0$ (3.0), $max\_ln\_ratio = 2.0$ (2.0), $dist\_thresh = 40cm$ (40cm), $endpts\_thresh = 40cm$ (60cm), $angle\_thresh = 10^o(10^o)$. Note the large distance and endpoints thresholds. As was described in section 3.2 those thresholds are iteratively and adaptively refined in *STAGE 4* of the algorithm.

Results from the Cathedral and campus experiments are shown in Fig. 5. Note the registration accuracy. 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 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 Intel 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). We believe that this is an excellent initial alignment between scans. The accuracy of our method can be further increased by using a point-based global ICP algorithm like [10].

We have described a fully automated method for the registration of large number of geometrically complex range data sets. Our algorithm is based on a segmentation and feature extraction step that reduces the geometric complexity and produces features that are used for registration purposes. Our pairwise registration algorithm does not require the scans to be spatially close with respect to each other. It is based on rejection of large number of line-pairs before an expensive verification step. Finally, a graph algorithm is utilized in order to register each individual scan with a central pivot scan. Scene symmetry is a problem that may lead our algorithm to produce inaccurate results. In this case the incorporation of a-priori constraints is needed. Also, our pairwise registration is not a symmetric operation. Finally, our metric, the absolute number of matched features, while commonly used in object recognition literature, could be replaced by the actual registration error in mm. Our method can be very efficiently applied to geometrically complex scans of large scale urban scenes. Due to its efficiency it can produce the desired result in a matter of minutes on a regular Intel workstation.

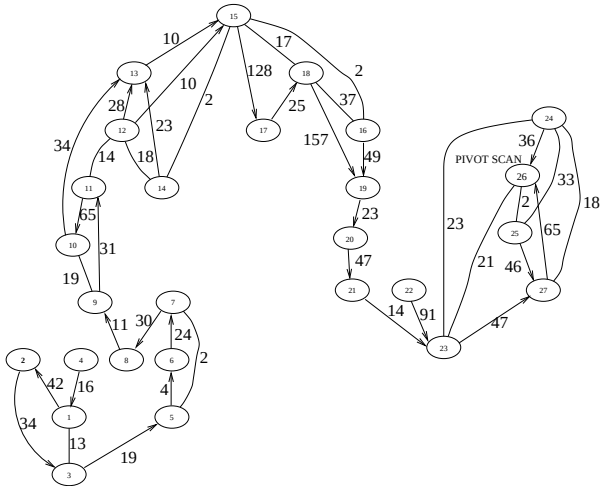

Figure 4: **Graph of the twenty-seven registered scans of the Cathedral data set. The nodes correspond to the individual range scans. The edges show pairwise registrations. The weights on the edges show the number of matched lines that the pairwise registration algorithm provides. The directed edges show the paths from each scan to the pivot scan that is used as an anchor.**

References

[1] F. Bernardini and H. Rushmeier. The 3D model acquisition pipeline. In *Eurographics 2000 (STAR)*, 2000.

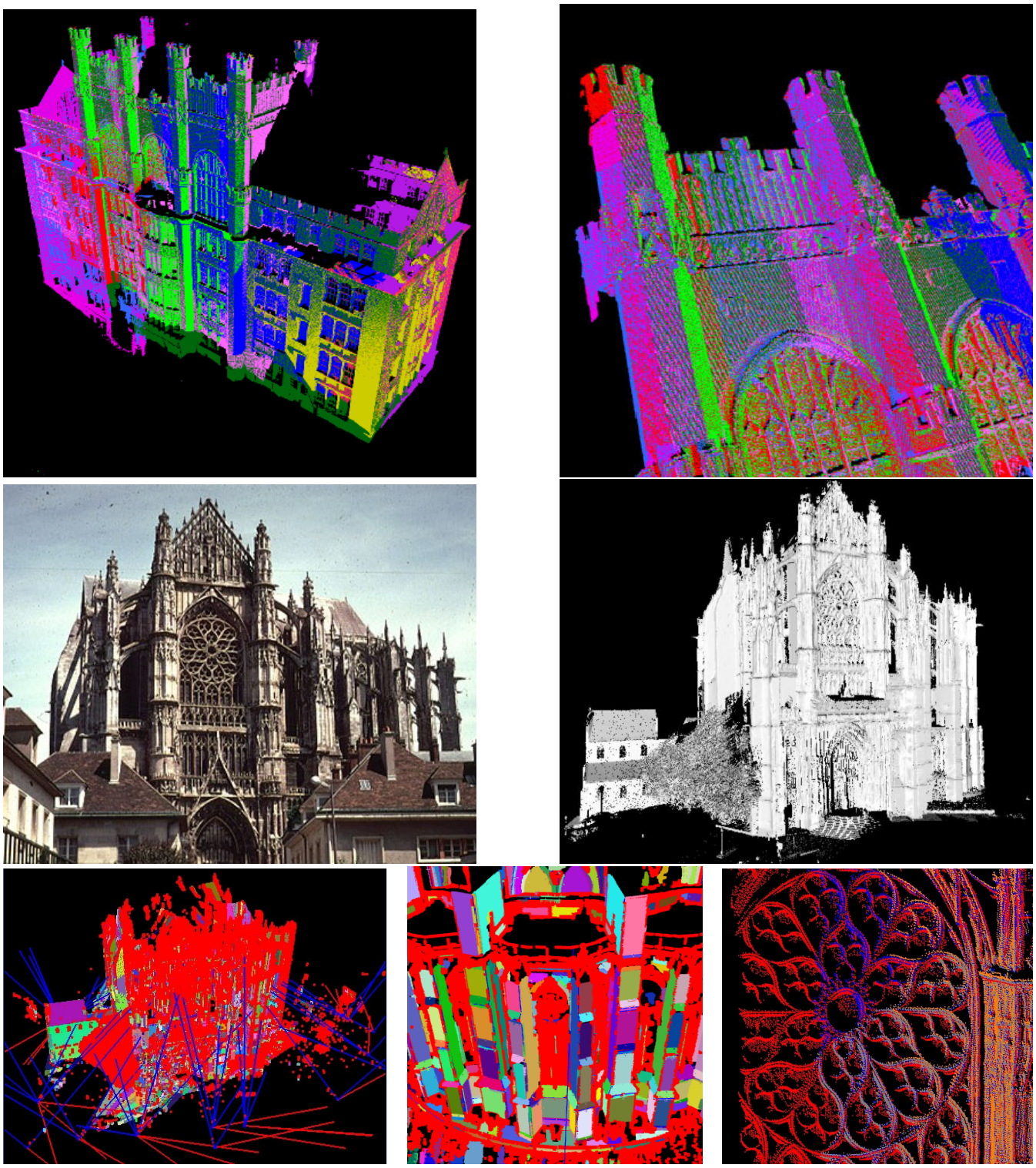

Figure 5: **[TOP ROW] (a) Ten automatically registered scans of Thomas Hunter. Each scan is represented with different color. For details on the performance of pairwise registrations see Table 1. (b) Registration detail. [MIDDLE ROW] (c) Image of the Ste. Piérre Cathedral, Beauvais, France. (d) 3D mesh of the Cathedral after all twenty-seven scans has been placed in the same coordinate system. Note that the mesh is slightly rotated with respect to image (c). [BOTTOM ROW] (e) All twenty-seven segmented registered scans of the Cathedral. The local coordinate system of each individual scan is shown. The z-axis of each scan points toward the scene. (f) 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. (g) Registration detail. Each scan is shown with different color. Note the accuracy of the alignment of points. COLOR images can be found at http://www.hunter.cuny.edu/cs/Faculty/Stamos/CVPR2003.**

Campus Building Results (average error 7.4mm) —— Cathedral Results (average error 17.3mm)														
	Line	Pre	S2	S3	M	t	Err	Line	Pre	S2	S3	M	t	Err
	Pairs	%	% (#)	% (#)		s	mm	Pairs	%	% (#)	% (#)		s	mm
1	301x303	16	1.7 (1555)	0.38 (346)	35	15	10.99	406x464	7	0.9 (1650)	0.3 (615)	42	39	9.37
2	303x290	17	2.8 (2429)	0.84 (735)	25	29	6.28	464x269	7	0.7 (888)	0.3 (443)	34	16	16.9
3	290x317	21	2.8 (2572)	1.88 (1728)	36	52	2.77	406x269	11	0.7 (794)	0.1 (104)	13	9	56.08
4	317x180	19	3.4 (1955)	1.15 (656)	28	21	14.96	151x406	21	1.1 (668)	0.8 (480)	16	7	5.34
5	211x180	21	4.6 (1759)	2.1 (802)	31	19	9.26	269x387	11	0.7 (702)	0.4 (369)	19	9	15.8
6	180x274	17	2.6 (1306)	0.34 (168)	22	9	11.42	326x197	10	0.9 (597)	0.1 (49)	24	4	11.68
7	114x274	19	1.6 (507)	2.2 (894)	33	6	5.61	197x143	15	1.0 (290)	0.3 (82)	30	3	6.44
8	274x138	16	1.8 (667)	1.5 (557)	31	5	3.08	143x194	16	1.9 (520)	0.1 (31)	11	3	29.24
9	138x247	18	2.3 (791)	1.3 (429)	20	5	1.36	151x406	21	1.1 (668)	0.8 (480)	16	7	5.34

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 ($(\mathrm{l}_1, \mathrm{r}_1)$ and $(\mathrm{l}_2, \mathrm{r}_2)$)). The efficiency of our algorithm is due to the great reduction of the pairs that need to be considered in this stage. Column *M* presents the number of matched pairs that the algorithm establishes. The running *Time t* of the algorithm (in secs) is shown for every pair (2GHz Intel machine). Finally, the pairwise registration *Error Err* 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. Note that the application of the well-known Iterative Closest Point algorithm will improve the results even further.**

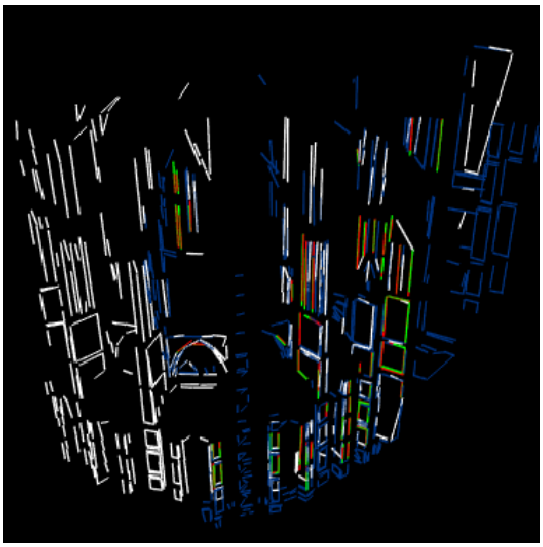

Figure 6: **Registration of lines between a pair of overlapping scans: left lines (white), right lines (blue), matched lines (red and green). Note, that due to the overlap of the matched lines some times only the red or the green component dominates.**

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

[3] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, pages 303–312, 1996.

[4] Cyrax technologies, 2000. http://www.cyra.com.

[5] O. Faugeras. *Three–Dimensional Computer Vision*. The MIT Press, 1996.

[6] B. K. P. Horn. Closed–form solution of absolute orientation using unit quaternions. *Journal of the Optical Society of America*, 4(4):629–642, Apr. 1987.

[7] D. Huber and M. Hebert. Fully automatic registration of multiple 3D data sets. In *IEEE Comp. Soc. Workshop on Comp. Vis. Beyond the Visible Spectrum*, Dec. 2001.

[8] A. Johnson and S. B. Kang. Registration and integration of textured 3–D data. In *3–D Dig. Imaging and Modeling*, pages 234–241, Ottawa, Canada, May 1997.

[9] L. Lucchese, G. Doretto, and G. M. Cortelazzo. A frequency domain technique for range data registration. *IEEE Trans. on PAMI*, 24(11):1468–1484, Nov. 2002.

[10] K. Nishino and K. Ikeuchi. Robust simultaneous registration of multiple range images. In *ACCV2002: The 5th Asian Conference on Computer Vision*, Jan. 2002.

[11] K. Pulli. Multiview registration for large datasets. In *Second Int. Conf. on 3D Dig. Im. and Modeling*, 1999.

[12] M. Reed and P. K. Allen. Constrained-based sensor planning for scene modeling. *IEEE Trans. on PAMI*, 22(12):1460–1466, Dec. 2000.

[13] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3D model acquisition. In *SIGGRAPH*, 2002.

[14] I. Stamos and P. K. Allen. Geometry and texture recovery of scenes of large scale. *Computer Vision and Image Understanding*, 88:94–118, Nov. 2002.

[15] G. Turk and M. Levoy. Zippered polygon meshes from range images. In *SIGGRAPH*, 1994.

[16] Visual Information Technology Group, Canada, 2000. http://www.vit.iit.nrc.ca/VIT.html.

[17] S. Weik. Registration of 3–D partial surface models using luminance and depth information. In *3–D Dig. Imag. and Modeling*, pages 93–100, Ottawa, Canada, May 1997.