Deterministic Surface Registration at 10Hz Based on Landmark Graphs With Prediction

Fred W. DePiero
Electrical Engineering Department
CalPoly State University
San Luis Obispo, CA 93407, USA
fdepiero@calpoly.edu

Abstract

Landmark graphs provide a means for surface registration, based on determining subgraph isomorphism to find scene-to-scene correspondences. Surface data used herein included both range and colour imagery. Images were acquired of a static scene from a moving sensor. The continuous motion allowed the sensor position to be predicted, which helped stabilize graph formation. Landmarks were determined using the KLT corner detector. Graph structure was established using nodes (landmarks) and edges that agreed well with predicted locations. Subgraph matching was approximated using the LeRP algorithm. A 6 DOF rigid transformation including translation and rotation was found via Horn's method. Test results on real and synthetic images indicate that a substantial speed improvement is possible, with greater determinism than ICP, while maintaining accuracy. Tests incorporated relatively large image displacements, spanning up to 30% of the sensor FOV for the image stream. Mean absolute errors remained under 0.8% FOV. Mean compute rates were $\approx$ 10 Hz with standard deviation ranging 6-9%, for an image size of 200x200. Tests were run on a 900 MHz PC. 141 test trials are reported, with comparisons against a fast version of ICP.

1 Introduction

Future viewing systems could permit multiple users to simulataneously explore scenes of remote locations with greater flexibility than is possible today. Rather that provide each user with their own video camera, positioner and transmission channel, the techniques described herein support a different approach. Images for each user would be rendered from a common set of 3-D surface data, based on a selected viewpoint. Surface registration facilitates the rendering by aligning the 3-D data into an expansive, contiguous set. This is needed to extend the data set beyond the field of view of a single sensor image.

Tele-presence viewing and immersion VR require 3-D sensing, registration, transmission and visualization technologies, some of which are unavailable today for real-time systems. For example a system using small 320x240 sensor images acquired at 25 Hz (motion picture rate), would need to acquire and process 1900k points/sec. For perspective, some state-of-the-art commercial 3-D sensors are within an order of magnitude of

this rate such as the Perceptron LASAR at 325k pts/sec and others [10]. Real-time registration is another key technology, in addition to sensing, that is very demanding and still requires further study. Registration is the focus of this work and a rate of 400k pts/sec has been achieved to date.

Given a sensor that acquires both surface location and colour, and given a real-time registration capability, it would become possible to render live images from viewpoints that are distinct from sensor locations. This would provide a substantial degree of flexibility for remote viewing systems compared to current systems. Applications such as a "television with a joystick" would become possible. For example in a sports broadcast, some viewers might choose to watch the hands of a golfer, others the ball, others the whole putting green.

For tele-immersion, two such views could be computed, one for each eye. Another application is tele-medicine, where this sort of system could provide useful flexibility. For example if a field technician positioned a range and colour sensor over a patients wound, then a remote doctor could examine the injury. Furthermore, if the doctors viewpoint were graphically presented to the sensor technician, then the technician could anticipate the doctors viewing needs in terms of standoff or locations, for example. This would permit a much more efficient viewing experience for the doctor. In another approach, the technician could possibly be replaced by a robot, which would use the doctors viewpoint as a basis for path planning when positioning the sensor.

All of these advanced viewing systems rely on surface registration that needs to be accomplished in a fast and deterministic fashion. The fundamental reason that registration is required is because sensors such as laser range finders (or simple video cameras) are all line-of-sight devices. Hence either multiple sensors or multiple images (from a moving sensor) would typically be required to form a complete set of surface data across an entire scene. The focus of this effort has been on static scenes with a moving sensor.

2 Approach

The long-term goal of this research is to pursue a technique that performs view registration automatically at rates ≈ 25 Hz, with relatively large image sizes and large sensor displacements. This performance goal targets remote viewing applications, with rapid sensor motion.

To achieve fast and deterministic processing, iterative [1] [19] [16] [13] [24], compute intensive [14], or random [8] approaches were avoided. Note that reported methods often do not separate the steps of determining corresponding points and determining the transform [19]. This limits compute speed. In the new landmark-graph approach these steps have been kept separate, and are implemented in a non-iterative fashion. This is an important distinction. Another difference is that correspondence between the data sets is determined only for select landmarks as opposed to traditional ICP, for example [1], where an entire range image is used in the cost function. This improves processing speed, but it does potentially limit accuracy because not all the scene data is used to find the transform.

The landmark-graph method does not rely on an affine motion model [26] [15] [28]. This permits relatively large disparities to be accommodated. For example, some of the image sequences in [28] appeared to contain a displacement of 0.2% of the FOV be-

tween subsequent images. Tests scenes documented herein ranged 2.5% to 5%. Thus the landmark-graph approach is an alternative to a stereo-based approach [9] for the large disparities.

Stability of the landmark-based registration is achieved by maintaining consistent inter-landmark (3-D) geometry, which is verified via the LeRP [5] subgraph-matching algorithm. This is in contrast to approaches such as [28] which provide robustness based on checks of deviation in the path of each individual feature, but that do not enforce a specific geometrical structure (attributed graph) between features.

A few remaining points distinguishing the landmark-graph approach from other reported methods: this method does not require photometric normalization as with [26]. It is also not reliant on a smooth surface assumption as with [12]. No particular assumptions regarding scene content (such as planar surfaces [27]) are made but that the KLT landmark locator [18] by employable. (KLT responds to corners or sharp prominences in the range data). And finally, the method does solve for all 6 DOF of the translation and rotation, as opposed to [15] [20].

2.1 Notation and Processing Summary

The following notation is used to describe the processing and representation of a stream of surface data. The stream is composed of a sequence of range and colour images, indexed by $i = 0, 1, 2, \ldots$

1. F_i, Sensor coordinate frame for ith scene.
2. (R_i, C_i) Range & colour images acquired at location F_i.
3. L_i, Set of landmarks found in R_i (w/rt F_i).
4. G_i, Graph formed from landmarks L_i.
5. T_i, Coordinate transform relating F_i to F_0.
6. V_0, Graph associated with all landmarks for entire image stream.
7. V_i, Predicted subgraph of V_0, approximating G_i.

The following steps are used to process each (R_i, C_i) sensor input, to yield a coordinate transform T_i.

1. Acquire new sensor images R_i & C_i.
2. Predict V_i based on V_0 and on estimate of sensor motion.
3. Find landmarks L_i in range image R_i. Reject unstable L_i.
4. Form G_i using L_i, mimicking structure of V_i, (both nodes and edges).
5. Compute attributes for G_i, using R_i & C_i.
6. Use LeRP algorithm to match G_i to V_i, the resulting subgraph mapping gives the L_i to L_0 correspondences.

7. Find transform T_i via Horn's method, using the L_i to L_0 correspondences.

8. Check residual error from Horn. Remove outliers and recompute T_i.

9. Repeat

2.2 Determining Landmark Location

Important performance goals for landmark detection and localization are: accuracy of location, speed and determinism of computations, and stability. Here, stability of detection refers to the consistent appearance of a given landmark, despite small shifts in sensor position, or despite sensor noise. See examples of landmarks in Figure 1.

Some reported techniques use invariant features that involve curvature classification, moments, or spherical harmonics [24]. These kinds of features rely on local smoothness for proper surface characterization. Jump discontinuities [23] violate this smoothness assumption. Local peaks may be adequate in some applications, such as registering aerial range imagery [4]. However, this simple form of detection may not be sufficient to handle generic scenes.

Several methods for landmark detection were investigated in this effort. It was observed during experimental trials that ridge curves [17] [21] appeared to be relatively stable. However, this approach was not pursued in light of challenges in parameterizing their shape. The use of neural nets, operating on a binary image of jump discontinuities [23] was also studied. However, the corner detector developed for the KLT algorithm (Kanade-Lucas-Tomasi) [18] appeared to be more tolerant to sensor noise, in terms of node stability and accuracy, compared to the neural networks.

KLT computes the eigenvalues associated with a covariance matrix of intensity gradients. The gradients are computed over a small 5x5 window. A corner is associated with jump discontinuities that occur (ideally) along two perpendicular axes. Corners give rise to two large eigenvalues for the covariance matrix. An implementation from [2] was integrated into the system. Note that only the feature detector of [18] was used, not the tracking part of the KLT algorithm.

It may be appropriate to mention that care should be exercised when finding landmarks near jump discontinuities. A landmark should be selected on the nearer side of the jump. The scene location at the far side of the jump along the sensor line-of-sight at the discontinuity is inherently unstable. A ridge along a cylindrical surface is another example of an inherently unstable landmark. Using the KLT corner detector, and selecting the nearer surface, avoids these kinds of instabilities.

Post processing was used to improve the accuracy of landmark locations. This was necessary in part because the [2] implementation incorporates a multiresolution image pyramid, for improved speed. This does however appear to limit accuracy. The post processing also assured that the landmark was on the nearer surface of a jump discontinuity.

In the post-processing step, a small neighbourhood surrounding the original location (from KLT) was examined to find the sharpest corner. The nearby pixel with the largest fraction of distant neighbours was selected as the new landmark. The more distant neighbours were identified using a simple jump threshold in the range image. Hence, the measure of the local sharpness was simply the fraction of more distant neighbours surrounding the landmark. This attribute proved useful for colouring the landmarks (described next) and for eliminating unstable landmarks.

The KLT corner detector reacts to any variation in shape that occurs within a window of pixels. It is possible that within a given window, a foreground object might have a simple straight edge, while a background region might present additional curvature. The net result can make the KLT detector misfire. This is an undesirable result as the straight edge of the foreground object is not a corner, and hence is an unstable location. The local sharpness measure helped eliminate these cases. A threshold of 60% was applied to the count of more distant pixels. This eliminated the straight edge cases. This general sort of refinement of the results of KLT is also discussed in [28].

2.3 Determining Landmark Attributes

Node colours were represented by a 2x1 vector quantity. The components included hue from the colour image and the measure of local sharpness, described above.

Due to imperfect alignment of the sensors range and colour images, it is desirable to provide improved robustness when combining data from these inputs. The examination of errors when using an inter-camera calibration model revealed problems near jump discontinuities (as to be expected). A simple approach of median filtering the hue values in the (mapped) neighbourhood of the landmark appeared beneficial.

2.4 Determining Graph Structure and Edge Attributes

Previously implementations of landmark graphs [4] used a Delaunay triangulation [7] [25] to determine which node locations were linked by an edge. For applications with large standoffs, such as with aerial range imagery, the triangularization may be an acceptable approximation. However, the assumption of planarity was not appropriate for the close-range scenes used in this investigation.

As described in Section 2.1, a graph V_0 is maintained for all landmarks in the sensor data stream. As such, V_0 could potentially grow without bound depending on the sensor trajectory. This is undesirable from the standpoint of subgraph matching. A limit on graph size is preferred for computational speed. To provide this size limit a subgraph V_i, of V_0, is predicted. The subgraph V_i contains nodes that are expected to be within the field of view of the sensor, given the sensors previous position and the velocity estimate. V_i is formed using V_{i-1} and then adding or removing portions of V_0.

A graph G_i must also be generated for each new set of sensor data (R_i & C_i). The structure of G_i is established in a manner that mimics the structure of V_i as much as possible. For this, an edge in G_i is introduced between two landmarks L_i^a and L_i^b if

$$|L_i^a - L_0^x| < t \; and \; |L_i^b - L_0^y| < t \tag{1}$$

Where L_0^x and L_0^y are a pair of landmarks in G_0 that are connected by an edge, and "t" is some appropriate threshold of distance.

This can lead to the introduction of "redundant" or "parallel" edges in G_i, compared to the structure of V_i. As such, graphs G_i tend to be supersets of ideal matching subgraphs. This is acceptable because the subgraph matching algorithm works well in the presence of this type of structural difference. See test trials described in [6].

The attribute, or colouring, for graph edges is simply the Euclidean distance between nodes. Distances were computed in world coordinates, not just a pixel distance, making

the edge colouring tolerant to standoff changes. This formed an object-centered representation that could be compared without first aligning the range images.

2.5 Approximating Subgraph Isomorphism

Noisy sensor data introduces some fundamental limitations to the accuracy and stability of landmarks. This limits the similarity of scene graphs. Varying occlusion with different sensor viewpoints also limits the similarity of scene graphs. For these reasons, graphs made from real sensor data tend to be imperfect representations. Since the graphs are imperfect, an exact method of determining subgraph isomorphism has limited use and consumes inappropriate compute time. Hence using an approximate method of graph matching is a reasonable approach.

Subgraph matching is accomplished using the LeRP Algorithm [5]. Also see Appendix. LeRP approximates a subgraph isomorphism via a deterministic procedure, based on the comparison of length-r paths. The LeRP algorithm yields a set of corresponding locations in the two input scene images, from which the absolute orientation may be found.

Node and edge colours are compared during the matching process. As these colours are continuous quantities, a threshold on colour differences was applied.

2.6 Determining Absolute Orientation

Horn's method [11] was used to determine absolute orientation. This reveals all 6 degree of freedom in the rigid transformation. It operated on the corresponding landmarks identified by the graph matching operation.

After an initial pass with Horn's technique, the residual error of each (3-D) landmark location is computed. The presence of outliers is checked via a simple threshold and the associated landmarks are removed and the transform recomputed. The process of finding the transform and checking consistency was repeated (fewer than) F times, for F landmarks. Hence the effort in this stage of the processing is bounded by $O(F^2)$. A minimum of 3 pairs of corresponding points is needed to Horn's method. Typically no outliers were removed, but occasionally some were present and this step helped improve accuracy.

3 Summary of Computational Effort

Each of the processing steps requires worst-case effort that has a polynomial bound. The effort described in Table 1 assumes: NxM range images and F landmarks. D is the mean degree of landmark graphs and Q is the total number of edges.

Processing effort for the graph matching is on the order of $O(F^3D^2R)$. The parameter R is actually a weak function of F (see [5]) but was set to a constant in all tests reported herein.

4 Testing and Results

As shown in Table 2, two types of scene data were used. The "Sensor" data was acquired using a structured light sensor, similar to [6], that was built in-house. The device includes

Find Graph V_i	Predict Sensor Motion	$O(Constant)$
	Find V_i from V_{i-1}, and V_0	$O(F)$
Find Landmarks L_i	Find KLT Corner Features	$O(NM)$
	Refine Landmark Locations	$O(F)$
	Reject Unstable Landmarks	$O(F)$
Find Graph G_i	Find Structure By Mimicking V_i	$O(QF^2)$
	Colour Nodes	$O(F)$
	Colour Edges	$O(F^2)$
Match Graphs V_i and G_i	LeRP algorithm	$O(F^3D^2)$
Find Coordinate Transform T_i	Horn's Technique	$O(F)$
	Remove Outliers & Recompute	$O(F^2)$
Update Graph V_0	Refine Landmark Locations	$O(F)$

Table 1: Computational Effort for Landmark Graph-Based Registration

a mechanical positioner, permitting it to collect the 3-D data set. The sensor also includes a colour camera [3].

The in-house sensor is relatively slow, relative to commercial 3-D range cameras. Furthermore, it is these state-of-the-art range cameras that are driving the goals for the new registration technique. Hence range images from the sensor were acquired and then stored for use in testing. Point clouds generated by the sensor were resampled to form scene imagery.

The "Synthetic" test scenes were generated using models of blocks, with ray tracing calculations to determine each pixel of the scene. Gaussian noise was added to the range and hue images during testing, spanning 2% of the intensity range.

Tests with synthetic scenes benchmarked accuracy for both sensor translation and rotation. Translational steps were 0.1 inches 3% of the 3.6 inch sensor FOV. The total shift over an image sequence was 22% of the FOV. Rotational steps were of 1 degree spanned a total range of 5 degrees. A total of 23 unique synthetic scenes were used in testing. Synthetic scenes were reused used on multiple test trials, by adding random noise. Scenes scanned with the in-house sensor had translational steps of 2.5% (of an 8 inch FOV) with an overall span of 30% FOV. Objects scanned with the real sensor had a fairly consistent standoff. Hence the synthetic images appeared somewhat more cluttered.

4.1 Benchmarks Against a Fast-ICP Algorithm

An ICP algorithm was used for comparison purposes [1] [24]. ICP was implemented with a simplex optimisation routine [22]. To reduce the effort of evaluating the cost function, a simple image difference was used. In each evaluation the point cloud was transformed and a range image was formed. The mean absolute difference in pixels was then computed. This avoids the step of having to search for the closest point in the point cloud. It does introduce error, however, because the closest 3-D point might actually be found via a lateral shift to an adjacent pixel. To improve compute speed only the grey level difference with the current pixel was considered. This type of resampling is also described in [15].

Another variation on ICP used herein maintained a fixed number of iterations. This permitted more direct comparisons between ICP and the landmark-graph method, as both

Scene	Type	Num Trials	Trans. Error *Graph*	Trans. Error *ICP*	Rot. Error *Graph*	Rot. Error *ICP*	Rate (Hz) *Graph*	Rate (Hz) *ICP*
Synthetic	Rot.	50	0.8%	1.5%	0.7	1.1	$10 \pm 7\%$	$0.14 \pm 19\%$
Synthetic	Trans.	80	0.1%	0.2%	1.7	0.7	$10 \pm 9\%$	$0.13 \pm 22\%$
Sensor	Trans.	11	0.6%	0.6%	1.1	0.3	$10 \pm 6\%$	$0.14 \pm 6\%$

Table 2: Results of test trials indicate that the accuracy of the landmark graph approach is comparable to Fast-ICP, while providing a substantial speed improvement. ICP was run with 200 (a constant number of) iterations. Mean absolute error values are reported. All images were 200x200; tests run on a 900MHz PC.

algorithms were deterministic. Exactly 200 iterations of ICP were run in each trial.

Another speed optimisation step for ICP involved limiting the region of interest used in the cost function comparisons to areas near landmarks. This somewhat intertwined the implementation of the two techniques. This approach is being investigated as a potential means to create a hybrid technique that combines ICP with the landmark-graph method.

4.2 Test Results

Table 2 summarizes a total of 141 trials. Each "trial" consisted of one registration operation on a given pair of images. Percent error was computed via the absolute mean displacement error, and then expressed relative to the sensor field of view. Rotational errors are also given as absolute means. All means were computed over the entire image sequence, including initial transient.

As seen in Table 2, the landmark graph approach rivals the accuracy of Fast-ICP, while executing much faster, at ≈ 10 Hz. The landmark-based approach also provides a greater degree of determinism than ICP as indicated by the standard deviation of the processing time. Fast-ICP was run with 200 iterations and the images for all test trials were 200x200. Tests were run on a 900MHz PC.

A reason ICP suffers variations in processing rate despite the use of a fixed number of iterations is due to the simplex-based optimisation [22]. The simplex method will use a varying number of evaluations of the cost function, depending on the type of simplex movement ("contraction" versus "flip"). Note that other optimisation techniques also possess this type of computational variation per iteration, such as Hooke-Jeeves.

The landmark-graph approach also suffers some variations in processing rates, as shown. This is due to varying numbers of landmarks. The degree of this variation can be mitigated somewhat, for example if the system processes an image stream with substantial changes to scene content, by adjusting detection threshold of KLT. Dynamic means of threshold adjustment are of interest in future studies. Limits on F are also possible when computing Graph V_i. See Section 2.1.

5 Conclusions and Future Studies

Reported results are encouraging. The landmark-based approach is able to achieve rates $\approx$ 10 Hz for 200x200 images on a single PC. A 1% accuracy appears comparable to ICP and to other reported techniques. Yet landmark-graphs provide higher speed and greater determinism than ICP. The method also yields 6 DOF alignment parameters and can process image streams with relatively large changes in sensor position (5% in each image). The ability to handle large changes in sensor position not only allows for fast sensor motion, but also indicates that precise predictions of sensor motion are not critical for accurate registration.

The landmark-graph approach does have its share of challenges. Most significant is the computation of landmarks via a method that is both fast and stable. The KLT feature detector appears to work well. It would be beneficial to improve the rotational invariance. A circular neighbourhood might help in this regard. This is an on-going area of investigation.

Stable placement of graph edges was a challenge in a previous implementation [4]. The predictive approach that mimics graph structure of V_0 seems much better.

Future extensions could also include marrying the new technique with ICP in a post-processing step. This would permit more scene data to enter into the final transform calculations than just the landmarks. ICP could begin iterations using the alignment parameters derived via landmark graphs. A fixed number of iterations could also be used for determinism. A goal here would be to target improved rotational accuracy and it might be best to just search over those 3 DOF. As with the implementation here, the region of interest used in cost function evaluations for Fast-ICP can be restricted using the image regions near corresponding landmarks.

A general improvement to the landmark-graph approach is planned in the near future. This involves a post-processing step, where the graph V_0 is grown in size after each new image is aligned. This is an $O(F)$ operation, see Table 1. Growing V_0 in an on-line fashion would permit extended regions of surface data to be incorporated into a single contiguous data set. The implementation reported here is more suited for repeated sensor scans over a small area, from differing views.

Future work is also planned for an integrated visualization subsystem to provide a complete real-time remote viewing system.

6 Acknowledgements

The research described in this paper was carried out at CalPoly, under contract with the U.S. Department of the Navy, Office of Naval Research, Under Contract N000-14-02-1-0754. I would like to thank Kurtis Kredo, Tim Jackson, Brian Gleason and Ryan Manes for their assistance with video capture and image processing software, and with laboratory set-up and networking. I would also like to acknowledge Albert Liddicoat regarding discussions of the hybrid approach (future version).

References

[1] P.J. Besl and N.D. McKay. A method for registration of 3-d shapes. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 14(2):239–256, 1992.

[2] Birch. Implementation of the klt tracker, http://vision.stanford.edu/ birch/klt/.

[3] F. DePiero. Structured light range sensor, http://www.ee.calpoly.edu/ fdepiero.

[4] F. W. DePiero. Fast landmark-based registration via deterministic and efficient processing, some preliminary results. In *Proc. 1st Intl. Symposium on 3-D Data Processing, Visualization and Transmission (3DPVT)*, June 2002.

[5] F. W. DePiero and D.W. Krout. Lerp: An algorithm using length-r paths to approximate subgraph isomorphism. *Pattern Recognition Journal*, 24:33–46, 2003.

[6] F. W. DePiero and M. M. Trivedi. 3-d computer vision using structured light: Design, calibration and implementation issues. *Advances In Computers*, 43, 1996.

[7] O Faugeras. *3-D Comp. Vision, A Geometric Viewpoint*. MIT Press, 1993.

[8] M. Fischler and R. Bolles. Random consensus: a paradigm for model fitting with applications in image analysis and automated cartography. *Communications of the ACM*, 24:381–395, 1981.

[9] A. Fusiello, V. Roberto, and E. Trucco. Efficient stereo with multiple windowing. *Computer Vision and Pattern Recognition*, pages 858–863, 1997.

[10] John Hancock, Dirk Langern, Ryan Sullivan Martial Hebert, Darin Ingimarson, Markus Mettenleiterc Eric Hoffmanb, and Christoph Froehlichc. Active laser radar for high performance measurements. In *Proc 1998 IEEE Intl. Conf on Robotics and Automation*, May 1998.

[11] B.K.P. Horn. Closed-form solution of absolute orientation using unit quaternions. *J. Optical Society of America A*, 4(4):629–642, 1987.

[12] B.K.P. Horn and J.G. Harris. Rigid body motion from range image sequences. *CVGIP: Image Understanding*, 53(1):1–13, 1991.

[13] S. Hsu. Multiple-view constrained video registration and its applications. In *Workshop on Video Registration, IEEE Computer Society*, July 2001.

[14] A. E. Johnson and S. B. Kang. Registration and integration of textured 3-d data. *Image and Vision Comput.*, 17:135–147, 1999.

[15] A. E. Johnson and A. Miguel San Martin. Motion estimation from laser ranging for autonomous comet landing. *IEEE Computer Vision and Pattern Recognition*, 2:413–420, 2000.

[16] C. Kapoutsis, C. Vavoulidis, and I. Pitas. Morphological iterative closest point algorithm. *IEEE Trans. On Image Processing*, 8(11):1644–1646, 1999.

[17] J. T. Kent, K. V. Mardia, and J. M. West. Ridge curves and shape analysis. In *BMVC1996*, 1996.

[18] B. D. Lucas and T. Kanade. An iterative image registration technique with an application to stereo vision. *IJCAI*, pages 674–679, 1981.

[19] B. Luo and E.R. Hancock. Structural graph matching using the em algorithm and singular value decomposition. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23(10):1106–1119, 2001.

[20] E. Noirfalise, J.T. Laprest, F. Jurie, and M Dhome. Real-time registration for image mosaicing. In *Electronic Proc. of The 13th BMVC*, September 2002.

[21] X. Pennec, N. Ayache, and J-P. Thirion. *Landmark-based registration using features identified through differential geometry, Handbook of Medical Imaging*. Academic Press, 2000.

[22] W.H. Press, B.P. Flannery, S.A. Teukolsky, and W.T. Vetterling. *Numerical Recipes in C*. Cambridge University Press, 1988.

[23] L.G. Shapiro and G. C. Stockman. *Computer Vision*. Prentice-Hall, 2001.

[24] G. C. Sharp, S. W. Lee, and D. K. Wehe. Icp registration using invariant features. *IEEE Trans. Pattern Analysis and Machine Intelligence*, 24(1):90–102, 2002.

[25] J.R. Shewchuk. Triangle: Engineering a 2d quality mesh generator and delaunay triangulator. In *Proc. First Workshop on Applied Computational Geometry*, pages 124–133, May 1996.

[26] J. Shi and C. Tomasi. Good features to track. In *Proc. Computer Vision and Pattern Recognition*, pages 593–600, 1994.

[27] Gille Simon and Marie-Odile Berger. Real time registration of known or recovered multi-planar structures. In *Electronic Proc. of The 13th BMVC*, September 2002.

[28] T. Tommasini, A. Fusiello, E. Trucco, and V. Roberto. Making good features track better. In *Proc. IEEE Computer Society Conference on Computer Vision Pattern Recognition*, pages 145–149, 1998.

7 Appendix: LeRP Algorithm for Approximating Subgraph Isomorphism

Main Routine
Input: Graph G with nodes g_i, $0 \leq i < N_G$ and Graph H with nodes h_k, $0 \leq k < N_H$
Output: Mapping m(), that gives h_k = m(g_i).

1. Compute powers of adjacency matrices A^R and B^R for graphs G and H
2. $beta_{peak}$[][] = *FindBestBeta*(G,H,A^r,B^r)
3. Clear node-to-node mappings
4. For each L, $0 \leq L < minimum(N_G, N_H)$
   (a) Let peak = 0
   (b) For each unmapped node g_i
   (c) For each unmapped node h_k
      i. Verify consistency of mapping g_i to h_k given current m()
      ii. ρ = 0
      iii. For each mapped edge e_{ij}
         A. lookup associated edge e_{kl} where l=m(j)
         B. β = *compare*(i,j,k,l)
         C. γ = *compare*(j,j,l,l)
         D. $\rho = 1 - (1-\rho)(1-\beta)(1-\gamma)$
      iv. Next j
      v. α = *compare*(i,i,k,k)
      vi. $\rho = 1 - (1-\rho)(1-\alpha)(1-beta_{peak}[i][k])$

vii. If $\rho > peak$ Then
    A. g_{peak}=i
    B. h_{peak} =k
    C. peak=ρ
viii. End If

(d) Next k

(e) Next i

(f) If peak=0 Then GoTo END

(g) Let m(g_{peak})=h_{peak}

5. Next L
6. If (L=N_G) and (L=N_H) Then G is ISOMORPHIC to H, refer to mapping m()
7. Else a subgraph isomorphism exists between G and H, refer to mapping m()
8. END

Function: *FindBestBeta*(G,H,A^r,B^r)

1. For each node g_i
2. For each node h_k
    (a) For each edge e_{ij}
    (b) For each edge e_{kl}
        i. beta = *compare*(i,j,k,l)
        ii. Save $beta_{peak}$[i][k]=beta if maximal for nodes i,k
    (c) Next l
    (d) Next j
3. Next k
4. Next i
5. Return $beta_{peak}$[][]

Function: *compare*(i,j,k,l)

1. For $1 \leq r \leq R$
    (a) If $a_{ij}^{(r)} ! = b_{kl}^{(r)}$ Then Break
2. Next r
3. Return $(r/N)^2$

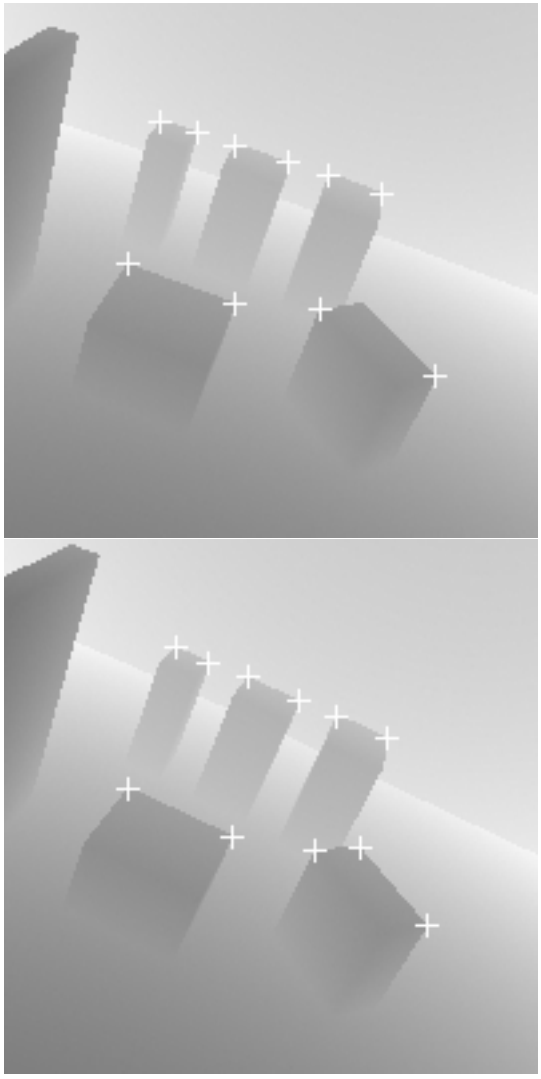

Figure 1: Example of scene landmarks found in two range images. Landmarks are determined via the KLT corner detector. These images contained a 5-degree shift.