

Fast Landmark-Based Registration via Deterministic and Efficient Processing, Some Preliminary Results^Ω

Fred DePiero

CalPoly State University, San Luis Obispo, CA 93407, fdepiero@calpoly.edu

Abstract

Preliminary results of a new method for range view registration are presented. The method incorporates the LeRP Algorithm, which is a deterministic means to approximate subgraph isomorphisms. Graphs are formed that describe salient scene features. Graph matching then provides the scene-to-scene correspondence necessary for registration. A graphical representation is invariant with respect to sensor standoff. Test results from real and synthetic images indicate that a reasonable tradeoff between speed and accuracy is achievable. A mean rotational error of ~ 1 degree was found for a variety of test cases. Mean compute times were found to be better than 2 Hz, with image sizes varying from 128x200 to 240x320. These tests were run on a 900 MHz PC. The greatest challenge to this approach is the stable localization and invariant characterization of image features via fast, deterministic techniques.

1. Approach Based on Subgraph Matching

The goal of this research is to pursue a technique that can perform view registration at rates approaching 10 Hz, without any user input for initial estimates. This performance goal is set to match the data rates of range cameras. It is also desired to have a method that computes the rigid transformation (translation plus rotation) in the presence of possible scale changes. Registration at these rates could permit sensor motion to be tracked in real-time. This would permit unconstrained movement of a sensor across a large scene.

To achieve fast and deterministic processing, iterative [1] [8] [13] [17] [18], compute intensive [10], or random [6] approaches were avoided. Established methods do not typically separate the steps of determining corresponding points and determining the transform. This limits compute speed. In the new approach these steps have been kept separate, and are implemented in a non-iterative fashion. This is an important difference for the new approach. Another difference is that correspondence between the data sets is determined only for select feature points. This improves processing speed, but it does limit accuracy.

Correspondence between the data sets is determined via graph matching. Graphs are formed using salient

features in each range image. Graph matching is accomplished using the LeRP Algorithm [2][4]. 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 range images, from which the absolute orientation may be found via closed-form solution [9].

A typical graph appears in Figure 1. The white segments designate an accurately matched subgraph for the scene.

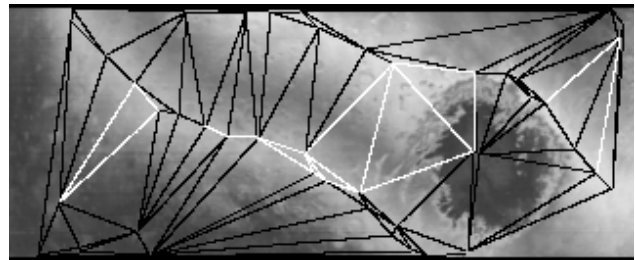


Figure 1. Graph associated with Mars Odyssey data. Nodes are associated with local maxima. Edges are determined via a Delaunay triangulation. White edges indicate a matched subgraph that contains nodes with an accurate mapping into the other scene.

1.1. Approach – Salient Features

Graphs describe the salient features in each range image. For rapid detection, local, well isolated, peaks in the range data were used to form the image features. These are desirable because they are likely to remain in view with small shifts in scene.

The significant percentage of black segments shown in Figure 1 demonstrates a lack of stability in feature detection. Stable and invariant feature detection – via fast, efficient, deterministic means – remains the greatest challenge in this new approach. While the lack of stability shown in the figure is undesirable, it was deemed an appropriate tradeoff in terms of processing speed. Other techniques employ more robust local features [10] but require more processing time. The approach taken here

^Ω Project sponsored, in part, by the Department of the Navy, Office of Naval Research.

was to rely on the LeRP matching algorithm to determine appropriate correspondences, despite noisy features.

Some reported techniques use invariant features that involve curvature, moments, or spherical harmonics [18]. These types of features react to jump discontinuities that may occur at a limb [19]. Such feature points were avoided in this new approach. Consider the occurrence of a limb where the line-of-sight of a sensor becomes tangent to a hillside. Slight movement of the sensor would alter the tangential viewing conditions. Hence jump discontinuities may be unstable in some situations. Ridge curves [20][21] or simply the use of isolated peaks are believed to be more stable.

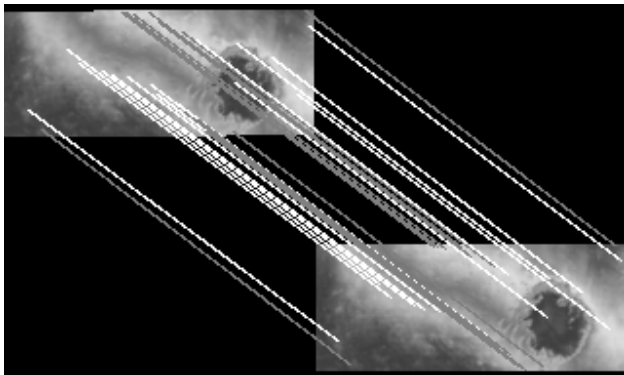


Figure 2. The connecting line segments indicate correspondences found via graph matching. White lines indicate acceptable matches and gray lines show correspondences that failed a test of the residual fit error.

To improve speed and to reduce sensor noise, the range images were subsampled by a factor of 8, making images 1/64 the original area. A 3x3-averaging kernel was used repeatedly, to subsample. During the subsampling process, occurrences of pixels with missing range data (due to occlusions, for example) were reduced in number.

When finding local peaks in the range data, a 3x3 window was used in the smallest subsampled image. This corresponded to a 24x24 window in the original image. A peak was defined as being a pixel with at most 1 other higher (3x3) neighbor.

To help recover the accuracy lost by peak detection in the subsampled images, feature locations were adjusted along the local gradient. A fixed number of steps, S-2, were used for an S-factor of subsampling (S=8, here).

1.2. Approach – Graphical Representation

Nodes were associated with each feature (local peak) in an image and were colored by the sharpness of the

peak. Sharpness was computed by finding the volume enclosed under a fitted 3x3 surface patch, centered at the peak.

The existence of an edge was established via a Delaunay triangulation [5] [16]. Distances between nodes were used to color edges. (Distances were computed in world coordinates, not just a pixel distance, making the edge coloring tolerant to standoff changes.) This formed an object-centered representation that could be compared without first aligning the range images.

A Delaunay triangulation was chosen because it is efficient, requiring on the order $O(F^2)$ effort, for F features [5]. This is the case for a 2-D mesh, associated with the 2 1/2-D range image. A Delaunay triangulation is also invariant with respect to translation and rotation, and with respect to node ordering. Also, the triangulation yields a graph (as opposed to a tree) and this works better with the LeRP algorithm.

1.3. Approach – Finding Subgraph Isomorphisms

Noisy sensor data introduces some fundamental limitations to the repeatability and stability of extracted features. 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 graph matching has limited use – and consumes inappropriate compute time. Hence using an approximate method of graph matching is a reasonable approach.

The LeRP algorithm [4] [11] approximates subgraph isomorphisms by comparing the number of length-r paths in each graph. These are found via A^R , where A is the adjacency matrix [7]. LeRP identifies the node-to-node mapping between the graphs by finding matching path counts, favoring assignments with higher values of R . As nodes are added to the mapping, the structural consistency with previously mapped nodes is enforced.

Node and edge colors are also compared during the matching process. As these colors are continuous quantities, some threshold on color differences was needed to verify similarity. A threshold of 2% was used for node colors and 1/2% for edge colors, in all the tests reported herein.

Processing effort for the graph matching is on the order of $O(F^3 D^2 R)$, where F is the number of nodes, D is the average degree. The parameter R is actually a weak function of F (see [4]) but was set to a constant in all tests reported herein. If the number of features increased significantly, then additional subsampling would be appropriate to maintain reasonable processing times. This would permit R to remain constant. In the reported trials,

F ranged from 25-80, approximately. Of these nodes, 20%-40% were typically matched.

1.4. Algorithm – Absolute Orientation

Horn’s method [9] was used to determine absolute orientation. This reveals the rigid transformation as well as any scale changes. It operated on the corresponding features identified by the graph matching routine.

The residual fitting error of each corresponding feature was checked. The process of finding the transform and checking consistency was repeated (fewer than) F times, for F local peaks. Hence the effort in this stage of the processing is bounded by $O(F^2)$.

2. Summary of New Algorithm for Registration

- 1) Find salient features in range images.
- 2) Refine feature locations.
- 3) Form graph. Use Delaunay triangulation to establish edges and compute feature descriptions.
- 4) Match graphs with LeRP algorithm to find corresponding features.
- 5) Find absolute orientation and check residual error.

Summary of Processing Effort

Step	Sub-Step	Effort
Find Range Features	Subsample	$O(NM)$
	Find peaks	$O(NM)$
Refine Features	Refine peak locations	$O(F)$
Form Graph	Delaunay triangulation	$O(F^2)$
	Find feature descriptions	$O(F)$
Match Graphs	LeRP algorithm	$O(F^3D^2)$
Find Absolute Orientation	Horn’s technique	$O(F^2)$

Table 1. Summary of processing effort required for each step in the algorithm. Each step has a polynomial bound on worst-case effort. Images were $N \times M$, containing F features. D is the mean degree of feature graphs.

Each of the above steps requires worst-case effort that has polynomial bound. The effort indicated assumes: $N \times M$ range images and F range features. Note that there are additional parameters that effect the processing time, such as the degree of subsampling for the original range image, and the window size used when looking for range features. These were omitted from the Table 1, for clarity.

3. Testing

Referring to Table 2, three types of range data were used. The ‘NASA’ data was from the Odyssey probe,

acquired Nov. 13, 2001. This is actually an intensity image in the visible - thermal spectrum, not a range image. The ‘Sensor’ data was acquired by scanning a relief map of the Great Smoky Mountains National Park. The ‘Synthetic’ data was generated randomly. The size of the various images is given in the table.

The ‘Sensor’ images appearing herein were acquired using a structured light range sensor, similar to [3]. The sensor used a laser line generator that casts a plane of light from a diode source. An inexpensive web camera then observed the intersection of the laser illumination with an object in the scene. One profile of an object was revealed in each camera image. A linear positioner advanced the laser and camera across the scene. Sensor calibration is also described in [3].

This sensor is relatively slow, relative to available 3-D range cameras. Furthermore, it is these 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 test trials. Also note the time necessary to load the range images was not included in the measures of execution time that are presented below.

3.1. Testing - Results

Image data was resampled at a random rotation in each test trial. The measured and true rotation angles were compared to estimate the accuracy of the registration measurement. Accuracy is described in terms of the mean absolute error of the rotation angles.

The ICP algorithm was used for comparison purposes [1][18]. The simplex optimization routine [15] was used to implement ICP. To reduce effort for ICP, only lateral displacements and a rotation in the image plane were optimized (3 DOF, not 6). This was done for simplicity. Also, test cases were constructed that made it easy for the ICP algorithm to converge in each case. In this way an apples-to-apples comparison with the new algorithm could be more easily made.

The nominal rotational difference was 5 degrees. This small rotational difference helped ensure the convergence of the ICP algorithm. Exactly 20 iterations of ICP were run, i.e. the termination was deterministic - resulting in a varying accuracy, rather than a varying processing time.

See Table 2. Twenty-five trials were run for each data set. Tests were run on a 900 MHz PC.

Results show that the new algorithm can achieve processing rates of 2 Hz or better. This is considered good. However, the new algorithm did not always yield 100% successful results (as in the 75 trials reported below). This was more common in test cases with higher rotational offsets. Improved feature stability should help improve reliability and overall accuracy.

Data Set	Size of Data Set	Abs. Rotational Error [New]	Abs. Rotational Error [ICP-3DOF]	Duration [New]	Duration [ICP-3DOF] (20 Iterations, Fixed)
NASA	128x320	0.78 Deg	0.46 Deg	0.26 Sec	1.6 Sec
Sensor	128x200	1.1 Deg	0.56 Deg	0.28 Sec	1.6 Sec
Synthetic	240x320	0.59 Deg	0.45 Deg	0.45 Sec	1.6 Sec

Table 2. Accuracy and execution speed for image registration test trials. Results indicate that a reasonable tradeoff between speed and accuracy is achievable. Twenty-five trials were run for each data set. For simplicity, only 3 DOF were optimized in the ICP routine and iterations were limited to 20.

4. Conclusion and Future Studies

The graph-based approach appears to provide a desirable tradeoff between speed and accuracy. It is possible that this type of approach could eventually be used to determine sensor motion on-line.

This new approach has its share of challenges. Most significant is the stable computation of invariant features, when computed via deterministic techniques that permit high-speed registration. The method employed herein is simplistic and further study and refinement is appropriate.

Future extensions include marrying the new technique with ICP in a post-processing step. ICP could be run with a fixed number of iterations to help improve accuracy (as all the image data would then be employed). Processing video streams could be enhanced by using results from previous images to predict subsequent scene conditions. Also, some experiments have been performed using features that are based on ridge curves [20]. These appear to be more stable than the simple method using isolated peaks described herein.

5. Acknowledgement

The Mars images from the NASA Odyssey probe were downloaded from the JPL website.

6. References

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

[2] F. W. DePiero, M. M. Trivedi and S. Serbin, Graph Matching Using a Direct Classification of Node Attendance, *Pattern Recognition Journal* 29 (6) (1996) 1031-1048.

[3] F. W. DePiero and M. M. Trivedi, "3-D Computer Vision Using Structured Light: Design, Calibration and Implementation Issues," *AIC*, vol.43 (1996).

[4] F. W. DePiero and D.W. Krout, LeRP: An algorithm using length-r paths to approximate subgraph isomorphism, Accepted by *Pat Rec J*. See www.ee.calpoly.edu/~fdepiero.

[5] O. Faugeras, *3-D Computer Vision – A Geometric Viewpoint*, MIT Press, Cambridge, MA, 1993.

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

[7] L.R. Foulds, *Graph Theory Applications*, Springer-Verlag, New York, 1992.

[8] 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) (2001) 1106-1119.

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

[10] A. E. Johnson and S. B. Kang, "Registration and integration of textured 3-D data." *Image and Vision Comput.*, 17, 135-147, '99.

[11] D.W. Krout, LeRP: An Algorithm for Finding Subgraph Isomorphisms with Applications to VLSI, Master's Thesis, CalPoly State University, San Luis Obispo, CA ('01).

[12] N. Nikolaidis, I. Pitas, *3-D Image Processing Algorithms*, John Wiley & Sons, NY (2001).

[13] C. Kapoutsis, C. Vavoulidis, and I. Pitas, Morphological iterative closest point algorithm, *IEEE Trans. On Image Processing*, v. 8, no. 11, pp 1644-1646, Nov. (99).

[14] Tefas, C. Kotropoulos, I. Pitas, Using support vector machines to enhance the performance of elastic graph matching for frontal face authentication, *IEEE Trans. Pattern Analysis and Machine Intelligence*, 23 (7) (2001) 735-746

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

[16] J.R. Shewchuk, Triangle: Engineering a 2D Quality Mesh Generator and Delaunay Triangulator, *Proc. First Workshop on Applied Computational Geometry*, Philadelphia, Pennsylvania, pages 124-133, ACM (May 1996).

[17] S. Hsu, Multiple-view constrained video registration and its applications, *Workshop on Video Registration*, IEEE Computer Society, Vancouver Canada, (July, 2001).

[18] G. C. Sharp, S. W. Lee, D. K. Wehe, ICP registration using invariant features, *IEEE PAMI*, 24 (1) (2002) 90-102.

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

[20] J. T. Kent, K. V. Mardia, J. M. West, Ridge curves and shape analysis, *British Machine Vision Conference*, University of Edinburgh (1996).

[21] X. Pennec, N. Ayache, J-P. Thirion, Landmark-based registration using features identified through differential geometry, *Handbook of Medical Imaging*, I. N. Bankman ed., 499-513, Academic Press (Sept 2000).

[22] A. E. Johnson, Surface landmark selection and matching in natural terrain, *IEEE Computer Vision and Pattern Recognition*, v. 2, 413-420 (2000).

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 $h_k = m(g_i)$.

Steps:

1. Compute powers of adjacency matrices A^R and B^R for graphs G and H
1. $\mathbf{beta_peak}[][] = \mathbf{find_best_beta}(G, H, A^r, B^r)$
2. Clear node-to-node mappings
3. For each L , $0 \leq L < \mathbf{minimum}(N_G, N_H)$
 - a. Let $\mathbf{peak} = 0$
 - 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. $\mathbf{rho} = 0$
 - iii. For each mapped edge e_{ij}
 1. lookup associated edge e_{kl} where $l=m(j)$
 2. $\mathbf{beta} = \mathbf{compare}(i, j, k, l)$
 3. $\mathbf{gamma} = \mathbf{compare}(j, j, l, l)$
 4. $\mathbf{rho} = 1 - (1-\mathbf{rho})(1-\mathbf{beta})(1-\mathbf{gamma})$
 - iv. Next j
 - v. $\mathbf{alpha} = \mathbf{compare}(i, i, k, k)$
 - vi. $\mathbf{rho} = 1 - (1-\mathbf{rho})(1-\mathbf{alpha})(1-\mathbf{beta_peak}[i][k])$
 - vii. If $\mathbf{rho} > \mathbf{peak}$ Then
 1. $\mathbf{g_peak} = i$
 2. $\mathbf{h_peak} = k$
 3. $\mathbf{peak} = \mathbf{rho}$
 - viii. End If
 - d. Next k
 - e. Next i
 - f. If $\mathbf{peak} = 0$ Then GoTo END
 - g. Let $m(\mathbf{g_peak}) = \mathbf{h_peak}$
4. Next L
5. If $(L = N_G)$ and $(L = N_H)$ Then G is ISOMORPHIC to H , refer to mapping $m()$.
6. Else a subgraph isomorphism exists between G and H , refer to mapping $m()$.
7. END

Function: $\mathbf{find_best_beta}(G, H, A^r, B^r)$

- a. For each node g_i
- b. For each node h_k
 - i. For each edge e_{ij}
 - ii. For each edge e_{kl}
 1. $\mathbf{beta} = \mathbf{compare}(i, j, k, l)$
 2. Save $\mathbf{beta_peak}[i][k] = \mathbf{beta}$ if maximal for nodes i, k
 - iii. Next l
 - iv. Next j
- c. Next k
- d. Next i
- e. Return $\mathbf{beta_peak}[][]$

Function: $\mathbf{compare}(i, j, k, l)$

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