

Surface Registration at 10Hz Based on Landmark-Graphs: Benefits for a Scalable Remote Viewing System

Fred DePiero

CalPoly State University, San Luis Obispo, CA USA

fdepiero@calpoly.edu (805) 756-2917

Abstract – *Real-time surface registration is a key technology for the development of future remote viewing systems. An architecture for a video distribution system supporting multiple users, with individual viewpoint selection, is suggested. The approach would provide a transmission bandwidth independent of the number of users, for scalability. The proposed architecture uses a method of surface registration based on landmark-graphs. Results from 141 test trials on synthetic scenes indicate that a mean absolute positioning accuracy under 1% of the sensor field of view is possible. The mean rate for registration was 10Hz, with a standard deviation under 10%. Tests were benchmarked on a 900MHz PC. The sensor images were 200x200 pixels and contained both range and color imagery.*

Keywords: Registration, Image Processing, Rendering, Graph Matching

1. Flexible Remote Viewing Systems

The goal of this research is to further methods of surface registration, for the enhancement of remote viewing systems. Current viewing capabilities such as TV or teleconferencing are quite limited by restrictions in viewpoint as each user is fed the same view. Furthermore, the selection is restricted to discrete camera signals.

Improved remote viewing systems should provide viewpoint flexibility for multiple users. Preferably, this should be done without simply introducing a camera with pan & tilt for each user and without increasing the transmission bandwidth in proportion to the number of users.

Such improvements may be possible, given an ability to do real-time surface registration. This refers to ‘stitching together’ sections of a scene that have been acquired by sensor(s) from different vantage points. This permits a large contiguous set of surface data to be constructed, as a basis for rendering remote views.

Accomplishing registration in real-time means that the alignment calculations must be completed at the rate of sensor acquisition, thus permitting immediate use of the sensor data for remote viewing. Voxel-based rendering could then provide imagery with an arbitrary viewpoint.

Given the real-time registration capability new approaches to video distribution become possible. See Figure 1. The server acquires new sensor images, and then computes an alignment relative to previous inputs. By transmitting the new sensor data to clients along with alignment transformations, the rendering operations may then be off-loaded to client machines.

This approach permits each client to have an independent viewpoint. It also means that the bandwidth of the transmission is determined by the sensor(s) only, not by the number of users. The method also offloads considerable effort, by not centralizing all the processing and rendering calculations [1].

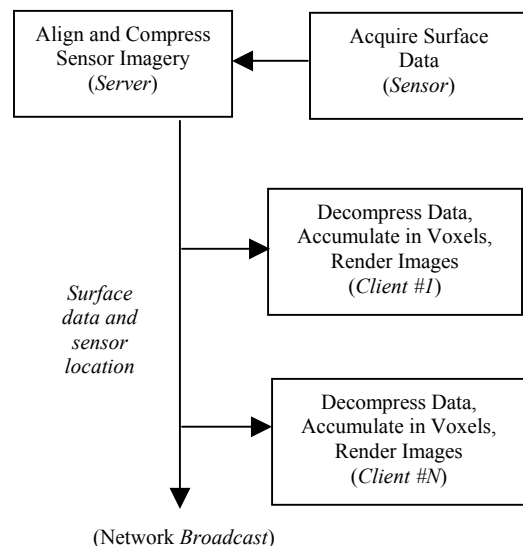


Figure 1. Architecture for a scalable remote viewing system with multiple users. View points are controlled by each user.

A visualization system should provide rapid response to user requests for new viewpoints. The proposed architecture is well optimized in this regard, as the viewpoint request and subsequent rendering are all local to the client machine. This makes the rendering frame rate and response to pan and tilt view changes all independent – and not limited by - the sensor data rate or the transmission rate.

2. Potential Applications

Applications such as a ‘television with a joystick’ would become possible, given the ability to perform real-time registration. This would support a broadcast transmission to many users, each with an independent viewpoint. For example with a sports broadcast, some viewers might choose to watch the hands of a golfer, others the ball, others the whole putting green. Scenes with an individual golfer would be amenable to this sort of remote viewing system. More complex scenes (such as a crowded street) could have a prohibitive level of occlusion, despite multiple sensors. For applications with tele-immersion, two such views could be computed, one for each eye.

Another application area is tele-medicine. A scenario is proposed here that is more flexible than just the transmission of individual medical scans. Rather, more interactive modes of observation are envisioned. For example if a field technician positioned a sensor over a patient’s wound, then a remote doctor could examine the injury. Furthermore, if the doctor’s viewpoint could be graphically presented to the sensor technician, then the doctor’s viewing needs could be better anticipated.

In another remote-viewing scenario a robot could use the doctor’s viewpoint as a basis for path planning and sensor positioning.

Awareness of another person’s viewpoint is pre-attentive knowledge, when interacting directly. However, this knowledge can be lost in a remote-viewing scenario. Means to graphically present a remote user’s viewpoint may be a useful feature for advanced systems.

3. Areas of Investigation

All of these advanced viewing scenarios rely on surface registration. The fundamental reason that registration is required is because sensors such as laser range finders (and even simple video cameras) are 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. Figure 2 illustrates the line-of-sight nature of a range sensor. The 2nd image has been rendered from a viewpoint that was offset from the sensor, revealing missing surface data.

Approaches for registration and visualization need to be deterministic and computationally tractable for real-time implementation. Methodologies in these areas are the focus herein. Also, this study is restricted to cases with static scenes that are scanned by a moving sensor.

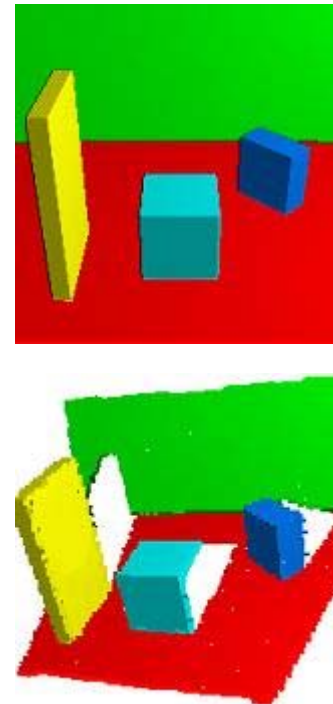


Figure 2. Sensor image (above, simulated) and a scene rendered from another distinct viewpoint. This illustrates the line-of-sight nature of range sensors. A low-resolution voxel array was used to store surface data.

4. The Challenge of Surface Registration

Surface registration is the process of determining the six DOF that describe changes in sensor location between a pair of input images. The goal here is to track changes in sensor location as the device is moved continuously across some arbitrary scene. The landmark-graph approach reveals sensor motion based entirely on an analysis of scene content – using no auxiliary sensors or alignment targets.

Work in registering range data dates back to random approaches such as RANSAC [2] and iterative

methods have been widely studied [3]. However, non-deterministic methods such as these are not preferred for real-time implementations. Robust methods that are computationally intensive have also been reported [4] but may not be able to achieve high frame rates. Other methods that track features [5] assume small image displacements and then use an affine motion model to describe local scene changes. The assumption of small displacements limits sensor velocity.

Some real-time methods have also been recently proposed [6][7]. However a direct comparison to these works cannot be made, as these rely on either a fixed camera position (rotation only) [6] or on an assumption of a particular type of scene content [7].

Reported methods typically do not separate the steps of determining corresponding points and determining the transform [8][3]. These steps are kept separate for landmark-graphs thanks to the LeRP algorithm for approximating subgraph isomorphism [9]. This is an important distinction with respect to computational efficiency.

5. Surface Registration and Remote Viewing With Landmark Graphs

Stability for the landmark-graph is provided via the similarity of inter-landmark geometry, which is verified via a subgraph-matching algorithm. This is in contrast to approaches such as [5] 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. See Figure 3.

The result of the graph matching processing step is a pair of subgraphs with identical structure (in terms of nodes and edges). The pair of subgraphs also has attributes that match to within specified tolerances. As such, a rigid transformation may be computed between the landmark correspondences given by the matching subgraphs.

The following notation is used, to describe the processing and representation of an image stream. The stream is composed of a sequence of sensor images, indexed by $i = 0, 1, 2, \dots$

- F_i , Sensor coordinate frame for i_{th} scene.
- (R_i, C_i) Range & color images acquired at F_i .
- L_i , Set of landmarks found in R_i (w/rt F_i).
- G_i , Graph formed from landmarks L_i .
- T_i , Coordinate transform relating F_i to F_0 .
- V_0 , Graph associated with all landmarks for entire image stream.
- V_i , Predicted subgraph of V_0 , approximating G_i .

The world coordinate frame for the voxel array is aligned with F_0 . Registration calculations are based on comparisons between the i_{th} sensor location, F_i , and the initial location, F_0 .

In a remote viewing system based on landmark-graph registration, the server could execute the following steps:

- 1) Acquire new sensor image.
- 2) Predict V_i based on V_0 and motion estimate.
- 3) Find landmarks L_i in range image R_i .
- 4) Form G_i using L_i , mimicking structure of V_i .
- 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) Compress R_i & C_i and broadcast to clients, along with T_i .
- 9) Update landmark positions L_0 and attributes stored in V_0 . Grow V_0 using any new territory exposed in G_i .
- 10) Repeat

The client could execute these steps:

- 1) Receive R_i & C_i along with T_i . Decompress sensor imagery.
- 2) Accumulate R_i & C_i into voxel array using the T_i transform.
- 3) Repeat.

The client would also continuously render scene images, based on the current voxel array content. This could be done asynchronously; at whatever rate the client platform can manage.

Previous work by this author with landmark-graphs restricted analyses to individual pairs of sensor images, not to image streams [10]. Stream processing is more appropriate for the continuous sensor movement. With an image stream, prediction may be used, as in [5]. Results of the landmark-graph approach, including prediction, are superior to those previously reported by this author [10]. More information on LeRP, the graph matching technique is

available in [9]. As LeRP is a relatively new algorithm, pseudo-code is included in the appendix.

6. Transmission Subsystem

Given that range data is available in addition to standard intensity images, and given the alignment data, there are new opportunities for image compression for the transmitted sensor data. Because sensor data is in the form of images, some simple variation on standard image compression techniques may be useful for the remote viewing system.

For example, the coordinate transforms T_i and T_{i-1} could be used to warp the images C_{i-1} & R_{i-1} to approximate the current images C_i & R_i . An image difference operation could then provide better compression over a method such as MPEG thanks to the warping operation that would make subsequent images more similar. Note the receiver would of course have to perform an un-warping operation.

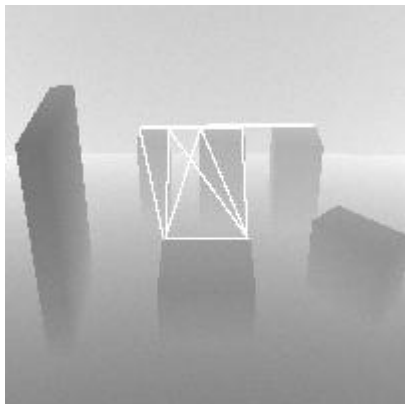


Figure 3. Landmark graph

7. Visualization Subsystem

In the proposed architecture, the client machine is tasked with accumulating range data and rendering user images. This offloads computations from the server side, making for a more balanced load. This also facilitates each user having their own viewpoint.

The method of shear-warp ray casting [11] is proposed for rendering. This method introduces a shear offset between adjacent layers effectively giving the voxel array a parallelogram shape. A projection of voxels then occurs along rows and columns of the array. This sort of projection is much more efficient than ray-tracing, for example. Projections are

performed back-to-front, relative to the user viewpoint. The warp operation restores proper image aspect ratio.

The compute performance demands on client processors in this architecture may be somewhat beyond the capability of today – depending on sensor data rates and image size. However, less expensive memory, faster general-purpose processors, and voxel visualization boards [12] may all contribute to meeting these increased demands, soon.

Choices of using a voxel array, and shear-warp, were driven by the use of 3-D point clouds of sensor data and the need for real-time processing. The voxel array is well matched to the storage needs of the 3-D data points. Shear-warp then provides efficient means for rendering.

Despite the simplicity of a point cloud approach, it may have some advantages over methods that use a polygonal surface representation [13]. Consider a situation with the sensor being swept back and forth over a static scene. As new 3-D points are acquired and aligned, a simple algorithm may be used to accumulate the data into the array – for example, just replacing the old points with new ones. Alternatively some type of averaging color values (hue) could be used when accumulating data. In contrast to this, consider a polygon-based approach. The polygons output from a sensor subsystem would have to be continuously merged to avoid unbounded growth of the surface description [13]. Such recombination and merging could be challenging in real-time. The voxel-based approach avoids this sort of problem.

8. Testing and Results

This is an on-going effort and the results of the registration with prediction are currently the main focus of investigation. Additional results documenting the effect of compression are under study.

The test suite included cases with both rotational movement and translation. Both real and synthetic sensor data has been included. Zero mean Gaussian noise was added to the synthetic sensor images.

The mean absolute position error is given as a percentage of the sensor field of view. The number of pixels across the sensor and voxel array was the same in these tests. Hence the percent error in position indicates the amount of misregistration expected in the voxel array. See Table 1.

Reports of accuracy and computational rate are given in Table 1, for both the landmark-graph approach and for a ‘fast-ICP’ method [10]. The fast-ICP method used a simple image difference, rather than point-by-point search for correspondence. It also

ran with a fixed number of iterations (200) to yield a deterministic algorithm that is more directly comparable to the landmark-graph approach.

	Translation Synthetic Scenes	Rotation Synthetic Scenes	Translation Real Scenes
Mean Absolute Error For LG	0.1%	0.7 ^o	0.6%
Mean Absolute Error For ICP	0.2%	1.1 ^o	0.6%
LG Rate	10 Hz	10 Hz	10 Hz
Mean +/- Std. Dev.	+/- 9%	+/- 7%	+/- 6%
ICP Rate	0.13 Hz	0.14 Hz	0.14 Hz
Mean +/- Std. Dev.	+/- 22%	+/- 19%	+/- 6%

Table 1. Test results for surface registration demonstrate a faster rate and greater determinism for landmark-graphs, compared to fast-ICP.

Test results in Table 1 show relatively low errors under 1% of the sensor field of view. These mis-registration errors result in a blurring of the surface data accumulated in the voxel array. Hence these error rates of are considered good. Figure 4 shows a relatively crisp image, after the accumulation of 10 registered sensor images.

The landmark-graph method was benchmarked to be ~70x faster than ICP. Landmark-graphs also provide better determinism, see standard deviations on processing rates. These factors make the landmark-graph approach superior for a real-time system.

The processing rates are given for a 900MHz PC. Although the rates are considered good relative to other reported methods, these would still need to increase for a broadcast system. Also note that the sensor image size was only 200x200 pixels. The new method does seem promising, nonetheless, given the modest compute platform.

AVI-format video clips are available for download [14]. The clips contain images rendered during the testing discussed below. The still image in Figure 4 is from one of these sequences.



Figure 4. Rendered image from voxel array after initial sensor image, and after the accumulation of 10 images (2nd). Note the new portions of the scene encountered after all 10 images are accumulated. Also note the reduction in the missing data (white areas).

9. Conclusions and Future Studies

Test results for the landmark-graph method of surface registration appear to yield relatively crisp imagery, with registration errors under 1%. The technique could provide the basis for a new means for distribution of surface data in a remote viewing system. Such a system could support multiple users and would be a scalable architecture. Opportunities for sensor image compression are superior to standard image streams because of the registration data, which could be used to align sensor images prior to compression.

Lossy compression methods will degrade the voxel data and the final user images, as will sensor noise and registration errors. To help mitigate some of the degradation a median operation could be performed on the voxel array. This step would retain the three most recent contributions to a voxel, and use

the median of the three for rendering purposes. This and other possible post processing steps are underway.

An outstanding issue in the design of the proposed architecture has to do with the introduction of new users. If surface transmissions are underway when a new client accesses the broadcasts, then the new client's voxel array will not match the state of other clients, nor of the server. Hence some means of voxel refresh would likely be required. One possibility is to provide a secondary, non-real-time transmission from the server to the clients for this purpose. The secondary transmission might consist of only filled voxels (to reduce data rates).

10. 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'd like to thank Kurtis Kredon, Tim Jackson, Brian Gleason and Ryan Manes for their assistance with video capture and image processing software, and with laboratory set-up and networking.

11. References

- [1] VisServer by SGI, <http://www.sgi.com>.
- [2] M. Fischler, R. Bolles, Random Consensus: a paradigm for model fitting with applications in image analysis and automated cartography, *Communications of the ACM*, 24, pp. 381-395, 1981.
- [3] P.J. Besl, N.D. McKay, A method for registration of 3-D shapes, *IEEE Trans. PAMI*, 14 (2) 239-256, 1992.
- [4] A. E. Johnson and S. B. Kang. "Registration and integration of textured 3-D data." *Image and Vision Comp.* 17, 135-147, 1999.
- [5] 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*, pp. 145-149, 1998.
- [6] E. Noirfalise, J.T. Lapresté, F. Jurie and M. Dhome, Real-time Registration for Image Mosaicing, *Electronic Proc. of The 13th BMVC*, University of Cardiff, September 2002.
- [7] G. Simon and M.-O. Berger, Real time registration of known or recovered multi-planar structures: application to AR, *Electronic Proc. of The 13th BMVC*, University of Cardiff, September 2002.
- [8] B. Luo and E.R. Hancock, Structural graph matching using the EM algorithm and singular value decomposition, *IEEE Trans. PAMI*, 23 pp. 1106-1119, 2001.

- [9] 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.

- [10] F. W. DePiero, "Fast Landmark-Based Registration via Deterministic and Efficient Processing, Some Preliminary Results," *Proc. 1st Intl. Symposium on 3-D Data Processing, Visualization and Transmission (3DPVT)*, (Padova, Italy), June, 2002.

- [11] P. Lacroute, and M. Levoy, Fast Volume Rendering Using a Shear-Warp Factorization of the Viewing Transformation, *Proc. SIGGRAPH. ACM*, Orlando, FL, pp. 451-458. 1994.

- [12] Voxel visualization board, by TeraRecon, <http://www.rtviz.com>.

- [13] Heckbert, P., and Garland, M. Survey of polygonal surface simplification algorithms. Tech. Rep. CMU-CS-95-194, Carnegie Mellon University, 1995.

- [14] DePiero, AVI Clip showing results of L-G Registration, <http://www.ee.calpoly.edu/~fdepiero>.

Appendix – LeRP Algorithm for Approximating Subgraph Isomorphism [9]

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
2. $\mathbf{beta}_{\mathbf{peak}}[i][k] = \mathbf{find_best_beta}(G, H, A^r, B^r)$
3. Clear node-to-node mappings
4. 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. $\mathbf{rho} = 1 - (1-\mathbf{rho})(1-\mathbf{alpha})(1-\mathbf{beta}_{\mathbf{peak}}[i][k])$
 - vii. If $\mathbf{rho} > \mathbf{peak}$ Then
 1. $\mathbf{g}_{\mathbf{peak}} = i$
 2. $\mathbf{h}_{\mathbf{peak}} = k$
 3. $\mathbf{peak} = \mathbf{rho}$
 - viii. End If
 - d. Next k
 - e. Next i
 - f. If $\mathbf{peak} = 0$ Then GoTo END
 - g. Let $m(\mathbf{g}_{\mathbf{peak}}) = \mathbf{h}_{\mathbf{peak}}$
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: $\mathbf{find_best_beta}(G, H, A^r, B^r)$

- 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 $\mathbf{beta}_{\mathbf{peak}}[i][k] = \mathbf{beta}$ if maximal for nodes i, k
 - iii. Next l
 - iv. Next j
- c. Next k
- d. Next i
- e. Return $\mathbf{beta}_{\mathbf{peak}}[i][k]$

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 $(r/N)^2$