PCL Toyota Code Sprint

Multi-Descriptor Registration across the 2D/3D Modalities

Adam Barclay

Open Perception Inc.

Supervised by:

Joseph Djugash, Ph.D.
Dr. Radu B. Rusu

2014

Contents

1.0 Introduction

With the recent ease of acquisition of co-registered RGB and depth scenes, there is a growing need, particularly in robotic applications that require long term autonomy, for enhanced performance by leveraging both 2D and 3D modalities for robust pose estimation, object recognition and grasping to name a few. Figures 1 and 2 reflect the challenges facing current techniques in image registration and pose estimation.

Figure 1: Image registration failure as commonly found on google maps [19] as of December 2013, highlighted by the author.

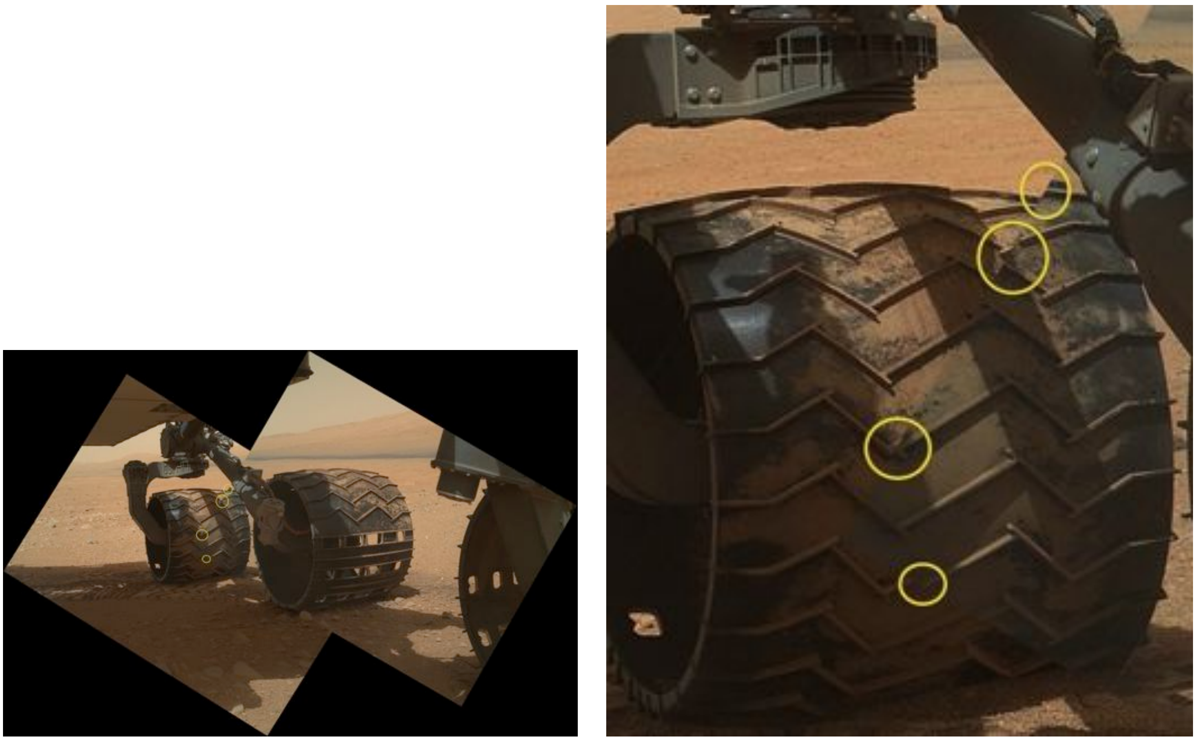


Figure 2: Image registration failure shown on a NASA stitched pair of images for the Mars Curiosity rover wheels [18], highlighted by the author.

The goal of this code sprint is two fold: The first is to develop a framework for benchmarking a set of state of the art descriptors across both 2D and 3D modalities. The second is to explore techniques, utilizing correspondence sets of multiple descriptor types across both the 2D and 3D modalities, that could be applied to improve RANSAC-based registration and pose estimation, as opposed to the classic single-descriptor type correspondence approach.

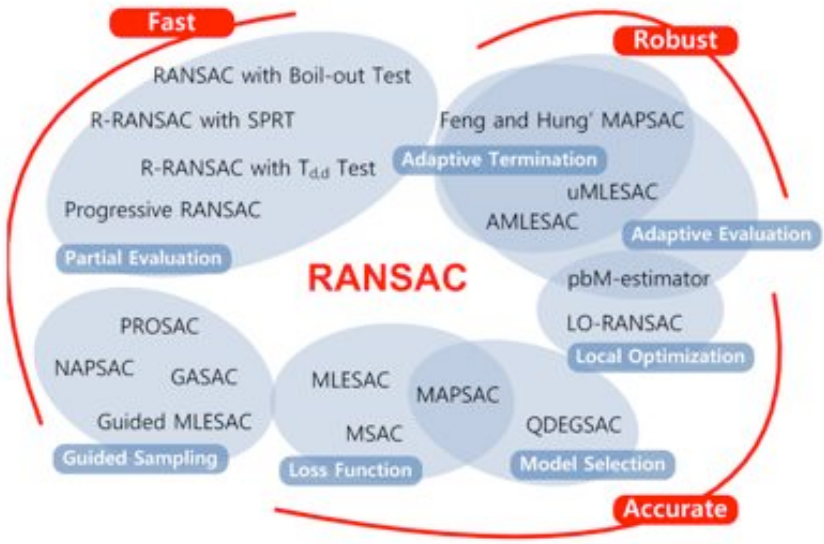


Figure 3: An overview of state of the art RANSAC approaches [1].

In the first and second milestones of this code sprint, a benchmarking framework for computing, evaluation, and analysis of 7 state of the art descriptor types (4 in the 2D modality and 3 in the 3D modality as detailed in section 2.1) has been developed, and was applied to the evaluation of each of the descriptors' behavior across real-world scenes. Furthermore, in the first milestone, various correspondence rejection filters as detailed in section 2.3 and in appendix D were implemented and evaluated in depth. In the second milestone, homography estimation based on RANSAC [3] was then applied to sets of multiple types of descriptors' correspondences. Based on detailed analysis and experimentation of the evaluation results, the multi-descriptor voting approach, MDv, was then introduced in the third milestone. The proposed MDv has been able to achieve a level of performance that is more consistent than that of state of the art single descriptor-type approaches such as SIFT.

The structure of this report, whose aim is to highlight key aspects and accomplishments of this code sprint, is as follows: Section 2 covers design considerations, descriptor selection, and multi-descriptor factors for RANSAC-based evaluation. In Section 3, implementation aspects, SDK issues and multi-threading support are discussed. Section 4 introduces a sample of the dataset used in the evaluation. Section 5 displays and analyzes the evaluation results. Concluding remarks and future work items are presented in section 6. Appendix A includes a reference page for correspondence Estimation, while appendix B includes a reference page for correspondence rejection. Appendix C includes a full listing of the framework's evaluation results for a sample run that spans 0-60 degree simulated rotations with 5-degrees of resolution. And finally, appendix D includes a full listing of the evaluation results of the first milestone. The reference section provides further details on related work, though it is assumed that the reader is already familiar with the concepts of image registration in both 2D (intensity) and 3D (point cloud) domains, keypoint and feature descriptors, correspondence estimation, and RANSAC among others. The source code and data set for this code sprint could be downloaded from: github.com/mult-desc/md**.**

2.0 Design

2.1 Design Considerations

One aspect that influenced the overall design is the interdependence between a given feature descriptor type and the associated keypoint type in the 2D modality, as opposed to those in the 3D modality. Scale and orientation invariant descriptors in the 2D modality such as SIFT inherit scale information from the underlying keypoints. The 3D descriptor types in this project do not have such dependency, and do not require scale or rotation information from the underlying keypoints as the chosen 3D descriptors are rotation invariant due to the employed local reference frame (LRF), but are not scale invariant by design. The chosen 2D descriptor types, on the other hand, are all rotation and scale invariant. Follows are the selected state of the art descriptors with references to the original published work:

- Classic 2D feature descriptors: SIFT [12], SURF [14]
- Binary 2D feature descriptors: ORB [7,15], BRISK [8,16]
- LRF-based 3D feature descriptors: FPFH [9], SHOT352 [10],[11], SHOT1344 [10],[11]

Furthermore, the following RANSAC-based registration estimation measures (also referred to in this report as derived measures) are introduced for multi-descriptor evaluation:

- MD2dR: RANSAC of k-best 2D multi-descriptor correspondences for homogaphty estimation.
- MD3dR: RANSAC of k-best 3D multi-descriptor correspondences for homography estimation.
- MD3dUR: RANSAC of the intersection set of k-best 3D multi-descriptor correspondences for homography estimation.

The evaluation results of the above techniques inspired the following approach for multi-descriptor voting. Further details on the evaluation results and the multi-descriptor voting can be found in the Experimental Evaluation section:

- MDv: Registration estimation computed as the average of the winning votes from the set of estimations of 2D, 3D, and derived (MD2dR, MD3dR, MD3dUR) multi-descriptor registration estimations.

2.2 Multi-Descriptor Evaluation Considerations

Multi-descriptor correspondence evaluation is based on the best correspondences, which are defined as the top K correspondences with minimal L2 distance measure; where K is an input parameter. There are several factors that needed to be considered in building a model for multi-descriptor correspondence evaluation. One factor is the numerical value of this minimal distance measure, which varies depending on the type and modality of the chosen descriptor. As a result, attempting to sort the correspondences after merging all correspondence sets will not produce the desired results. It is critical that distance-based sorting of correspondence sets precedes the merge process. That way, the best K correspondences of each descriptor type is included in the final merged set.

Another factor is the RANSAC threshold, which is dependent on descriptor's modality. In the 2D modality, the RANSAC threshold applies to pixel space where the 2D correspondences operate on keypoints with sub-pixel precision. In the 3D modality, the RANSAC threshold applies to the metric space of the point cloud. In each modality, the set of outliers has its own characteristics influencing the choice of the associated threshold value.

A third factor is that RANSAC, by definition, relies on the randomness of the input set in selecting candidates for each iteration. By having a sorted input, the error in homography estimation for a given single-type descriptor correspondences increased not in all, but in some 2D descriptor types such as ORB and SURF. Interestingly, the error in homography estimation decreased with a sorted RANSAC input (top 30%) for single-type correspondences of 3D descriptors and BRISK in the 2D modality. This observation inspired further experimentation with different randomizations [5] of the RANSAC input in evaluating multi-descriptor correspondence homographies. Based on experimental results, lower error rate in homography estimation could be achieved if the RANSAC input consists of best-K multi-descriptor correspondences that were merged 2-3 times. Further expansion (more than 3x) of the RANSAC input usually results in higher error rate, as redundancy could lead to degenerate conditions [4] outweighing the value of re-randomization. The code for this experiment has been disabled (could be easily enabled) for clear evaluation of the voting process which provided the best results in the multi-descriptor case.

2.3 Correspondence Estimation & Rejection

Correspondence estimation is the process of matching each keypoint from the source image with a keypoint in the target image based on the L2 distance between the associated descriptors in this implementation. As this approach assumes that each source keypoint must have a matching target keypoint, an assumption that may not always hold true, correspondence rejection must be applied to remove outliers from the estimated correspondence set. The following correspondence rejection approaches were explored, with detailed experimental evaluation shown in appendix D:

Correspondence cross check, also known as 2-way correspondence matching: In this approach source->target correspondences are matched against target->source correspondences. Correspondences that are not present in both sets are removed.

Lowe's L2 ratio measure[13]: In this approach, the ratio of the L2 distances of top 2 nearest neighbours is computed. Lowe's proposal is to reject correspondences whose ratio is >= 0.5. With the implementation of the first milestone, a ratio of 0.1 is found to be a practical threshold for correspondences in the 3D modality. One drawback of Lowe's ratio measure is the additional floating point division per correspondence. Lowe's measure should not be viewed as an alternative to the cross check approach described above, as Lowe's measure preserves correspondences based on local matching strength, essentially a form of non-max suppression whose desirability is application dependent.

Uniqueness filter: This filter could be viewed as an extension to the cross-check correspondence filter when applied to correspondence sets of multiple descriptor types. This rejection filter requires that multi-descriptor correspondences to be computed for the same set of keypoints. Although this filter returns a high percentage of inliers as could be expected, the pose estimation error may actually grow due to the smaller size of the inliers' set.

3.0 Implementation

In this project, the available implementation of keypoints, descriptors, correspondence matching, correspondence rejection and homography or registration computation of OpenCV 2.4.6 [20] and PCL 1.7 [23] were employed throughout. Dependency libraries such as Eigen[21] and boost[22] were also used. Early on, in the first milestone as detailed in appendix D, correspondence estimation and rejection were implemented from scratch to experiment with a uniqueness filter and Lowe's L2-ratio filter [13], exact vs. FLANN-based nearest neighbour search, and techniques to maximize the inlier rate using multiple 3D descriptors and filters. Figure 4 shows a sample input configuration parameters of the implemented framework.

```
===============================================
Configured Parameters:
max_rotation:   65.000
rotation_res:    5.000
thread_count: 2
best_k_sorted_2D: 100
best_k_sorted_3D: 140
leaf_size: 0.01
vote_threshold: 0.20
ransac_inlier_threshold_2D:    0.100
ransac_inlier_threshold_3D:    0.010
enable_graphics: 0
output_level: 1
live_sensor: 0
===============================================
```

Figure 4: Sample input configuration parameters

The parallel paradigm in which both openCV and PCL are used in this project highlighted the architectural differences between both libraries and areas for future improvements. For example, the openCV library (2.4.4-2.4.7) does not allow controlling the number of iterations when computing the homography using RANSAC, per the cv::findHomography() function. Another issue with the C++ wrapper for openCV (2.4.4-2.4.7) is the over-usage of the opaque OutputArray, effectively hiding the required datatypes. The point cloud library has been in general pleasant to work with. Though one aspect that is worth noting is the deep propagation of template types into the inheritance tree, making polymorphism a non-trivial task. One such example is descriptor computation, another example is CorrespondenceEstimation<T,T> when applied to multiple descriptor types.

Scalability via multi-threading is implemented through the boost library[22], where up to 5 threads could be dispatched. The number of threads could be automatically configured to match the underlying hardware, or explicitly specified. About 60% performance improvement has been measured when dispatching 2 threads instead of a single thread on an Intel Core II processor.

Figure 5: Sample CPU load w/2 threads, Intel Core II

4.0 Dataset

In this section, images of the dataset are presented, along with the requirements, reasoning, and acquisition method. In this implementation, the input dataset is represented by the point cloud library's PCD file format, ASCII or binary, that consists of co-registered, organized, 640x480 pixels, RGB along with depth data, commonly known as RGB-D. The source of the depth data, which forms our point cloud in the 2.5D space, could be passive as in stereo-based techniques, or active as in structured light, or time-of-flight techniques.

One requirement of the dataset is a real-world challenging environment, with regions of invalid sensor data and limited texture. Aside from the fact that such an environment is expected to be encountered naturally and frequently, as opposed to the highly textured Graffiti data set of figure 7 for example, the limited texture and invalid regions allow for experimental evaluation of the limiting factors of each descriptor type. The other requirement is a minimal dataset for efficient and tractable computational performance evaluation. The evaluation dataset is shown in figures 6, 8, and 9, with best 10-2D correspondences superimposed on the image to highlight the complimentary nature of the mixed sensor configuration. Each row of images contains two separate snapshots of the same scene, to contain sensor noise, then a simulated rotation is applied to provide a ground truth for the evaluation process.

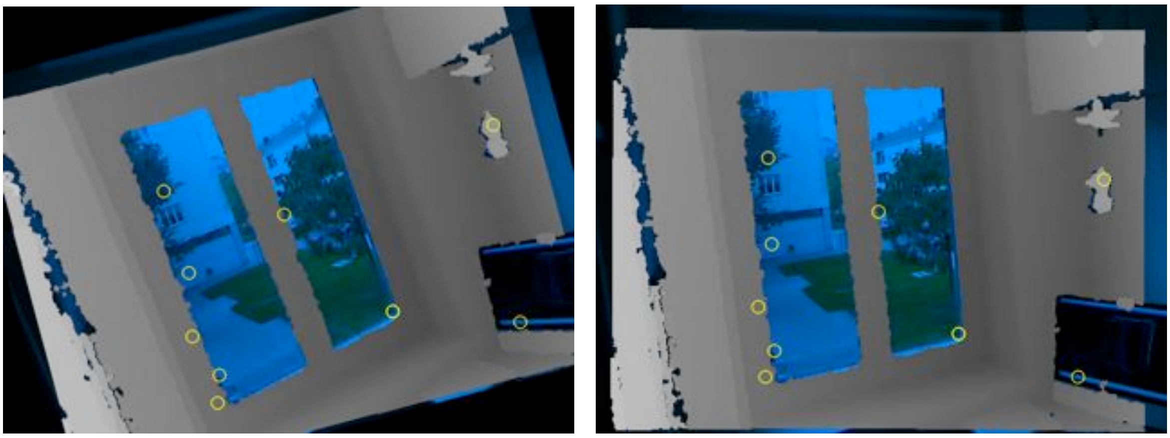


Figure 6: Sample of a real-world challenging dataset, with invalid depth regions (transparent blue), and limited textured regions (gray).

Figure 7: Sample of the Oxford VGG Graffiti Images, http://www.robots.ox.ac.uk/~vgg/data/data-aff.html

For the acquisition of the evaluation dataset, the Microsoft Kinect[1] sensor was utilized, though any other sensor or pre-processed dataset with the above requirements could be used. As the 3D descriptor evaluation of this project relies solely on the Kinect's depth data, it is worth elaborating further on the noise level of the Kinect depth sensor, and its contribution to the homography estimation error when compared with the 2D (intensity) modality – particularly for MD3dUR which relies on a smaller set size with high percentage of inliers. Per the sensor characteristics and noise properties of the Kinect sensor as reported in Andersen et. al. [17], the expected spatial resolution is around 4 pixels (ca. 15 mm) in both the x and y directions at a distance of 2.2 meters. The precision of the depth measurement, on the other hand, is around 40mm at a distance of 2 meters, with 4 levels of fluctuations that could range from 4-40mm. Such levels of noise are reflected in the homography estimation error across the various 3D descriptor types.

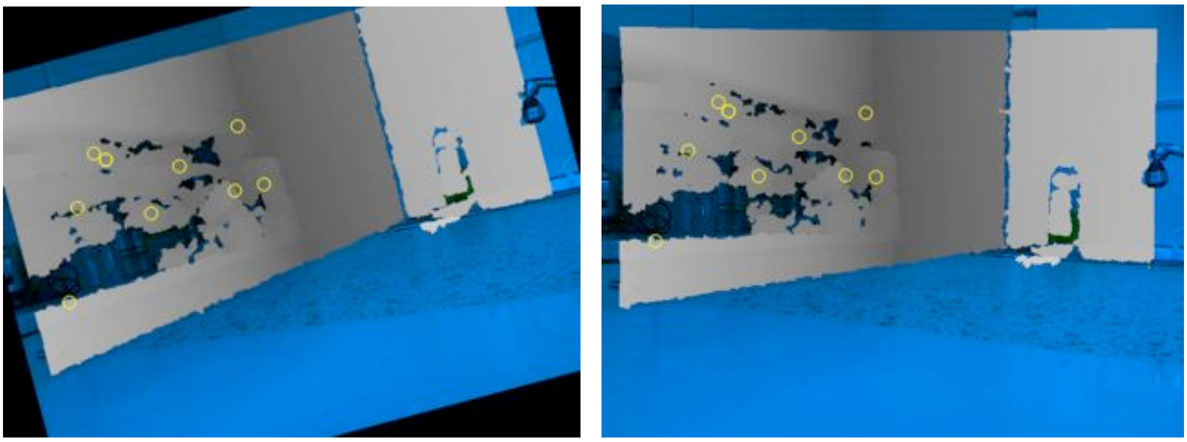


Figure 8: A fully indoor scene of the dataset, with sensor noise and simulated ground truth rotation.

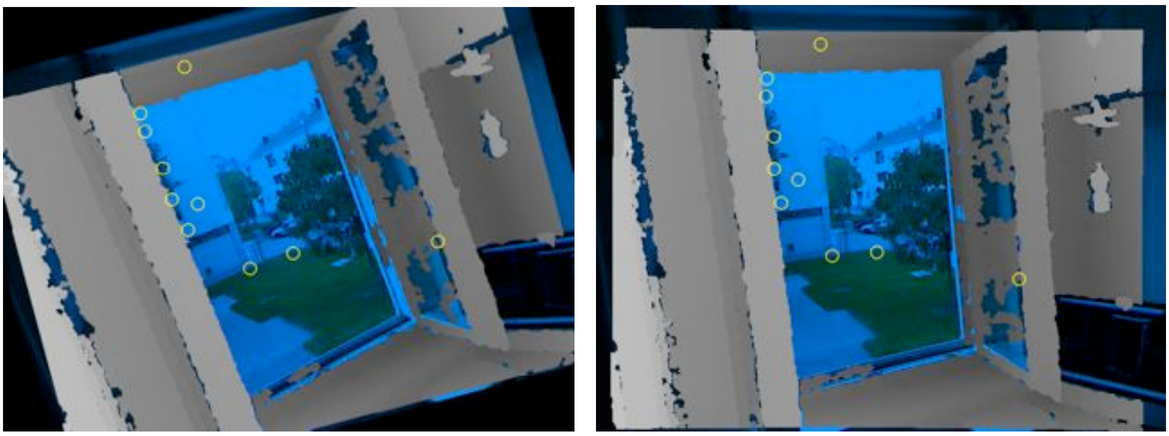


Figure 9: A mixed indoor/outdoor scene, with sensor noise and simulated ground truth rotation.

[1] Microsoft, Kinect are trademarks of Microsoft Corporation.

5.0 Experimental Evaluation

As shown in Figure 10, each set of correspondences is evaluated by applying a RANSAC based homography estimation. The error is the difference between the computed rotation and the known simulated rotation. In Figure 11, the promising performance of the proposed approach; MDv, when compared to 3D-based evaluations is illustrated. Figure 12 reflects similar promising performance when compared to 2D-based evaluations. Appendix C includes a full listing of a sample run w/simulated rotations of 0-60 degrees with 5-degree increments.

DescID	ComputedRot	EstError	InlierRate	ObjFuncEstErr	ObjFuncRot	numVotes
MDv (10)	5.007	0.007	n/a	0.007	5.007	7
CSHOT (6)	4.969	0.031	0.757	0.007	5.007	7
MD2dR (7)	5.040	0.040	0.093	0.007	5.007	7
MD3dR (8)	4.951	0.049	0.671	0.007	5.007	7
SHOT (5)	4.917	0.083	0.679	0.007	5.007	7
SIFT (0)	5.084	0.084	0.220	0.007	5.007	7
SURF (1)	4.911	0.089	0.140	0.007	5.007	7
FPFH (4)	4.872	0.128	0.593	0.007	5.007	7
MD3dUR(9)	5.174	0.174	0.697	0.007	5.007	7
BRISK (3)	4.757	0.243	0.080	0.007	5.007	7
ORB (2)	5.794	0.794	0.070	0.007	5.007	7

Figure 10: RANSAC-based homography estimation output, as compared with MDv, the proposed multi-descriptor voting estimate, computed for a simulated rotation of 5.0 degrees.

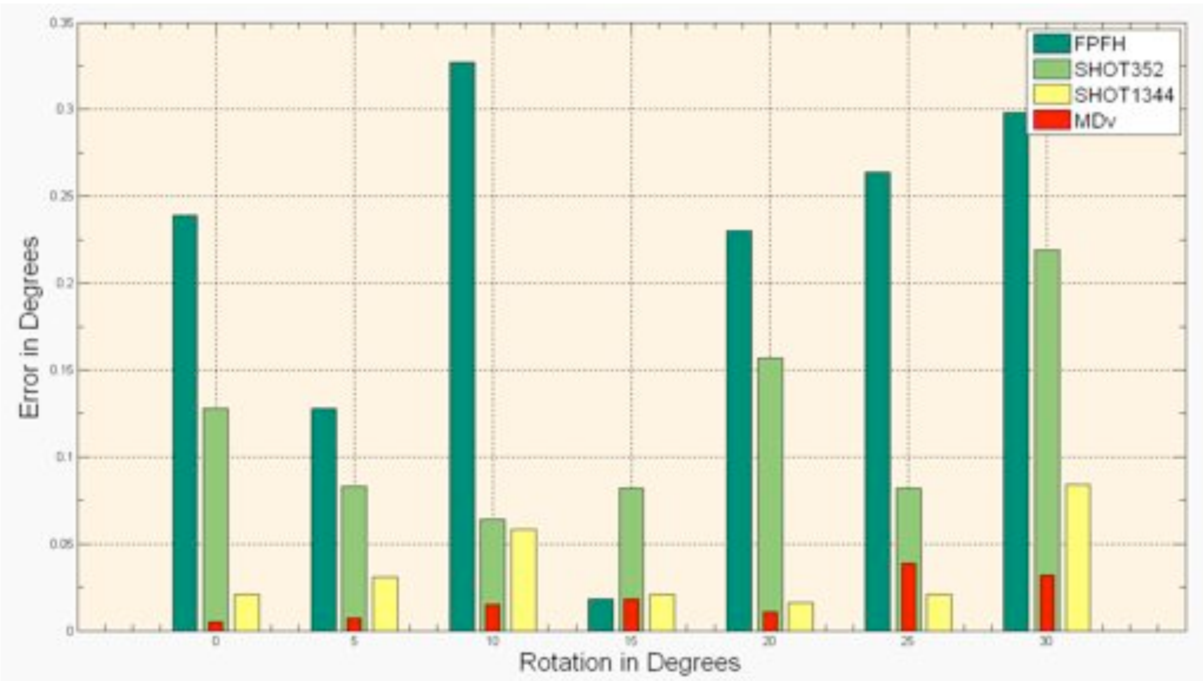


Figure 11: A bar graph representing the error associated with each 3D correspondence type evaluation, for 0-30 degree rotation, in 5-degree increments. Note the consistent low error of the proposed MDv.

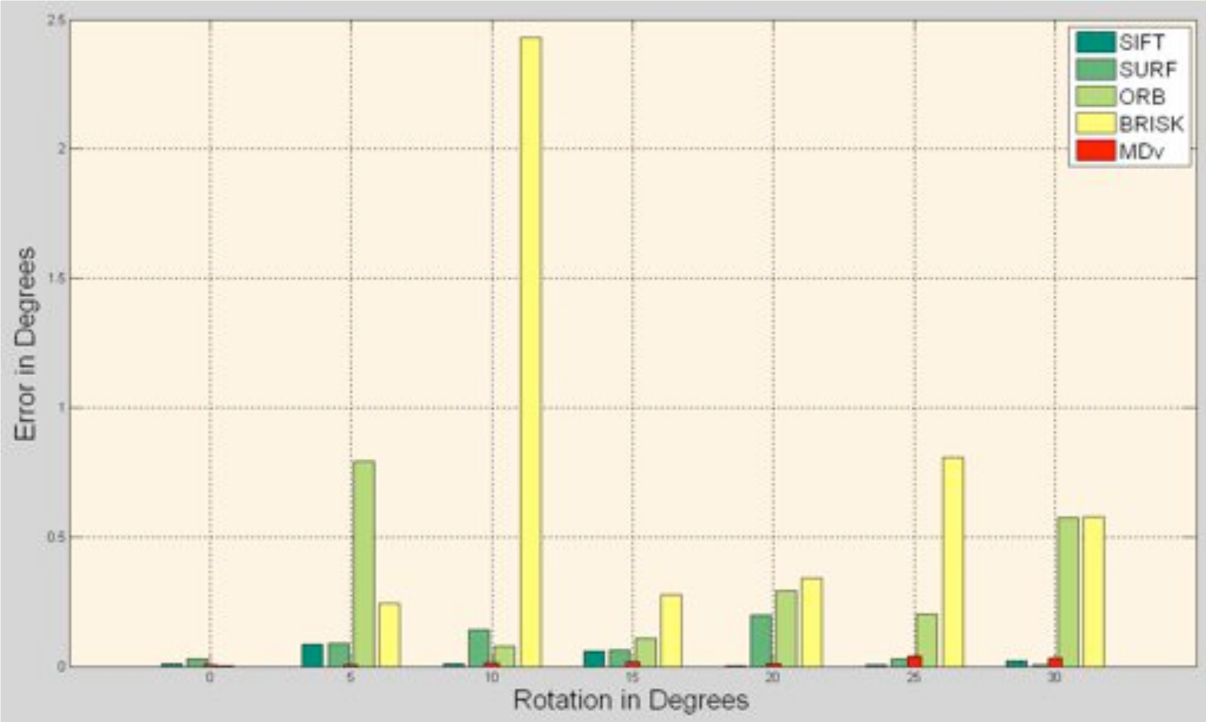


Figure 12: A bar graph representing the error associated with each 2D correspondence type evaluation, for 0-30 degree rotation, in 5-degree increments. Note the consistent low error of the proposed MDv.

Critical parameters in the evaluation process include RANSAC thresholding value for 2D homography estimation, RANSAC thresholding for 3D homography estimation, and multi-descriptor voting threshold. Further fine tuning could be accomplished per descriptor type. For example, it was observed that FPFH is rather sensitive to the normal estimation radius, while ORB and BRISK-based homography estimations are sensitive to the sorting process and to the parameter k that determines the best-k correspondences.

As compared with a homography estimation based on a single-type descriptor correspondence set, RANSAC-based multi-descriptor homography estimations; MD2dR, MD3dR, and MD3dUR, reflect mild but not significant reduction in the estimated error. In the 3D modality, the same set of keypoints across all 3D descriptors has been used to compute MD3dR and MD3dUR, impacting the randomization process. High sensor noise of the 3D modality, as opposed to the intensity noise in the 2D modality, is another factor, which is further magnified in MD3dUR. The use of the same keypoints across different correspondence sets increased the chances of degenerate cases in RANSAC, but none were encountered in the testing phase of the 3D modality. In the 2D case, each descriptor type used a different set of keypoints, which helped further randomize the RANSAC-based multi-descriptor homography estimation, MD2dR. The error output for MD2dR has been in general better than 2 out of the 4 2D descriptors of its class, but not consistent, as could be observed with the results at 55 degree simulated rotation.

On the other hand, the proposed multi-descriptor voting process (MDv) has outperformed homography estimations based on a single descriptor type of both 2D and 3D modalities, as well as those based on multi-descriptor types, in its ability to maintain a consistently stable output with a predictable accuracy in the 0.0X range, as shown in Figures 11 and 12, and in Appendix C across 0-60 degrees of simulated rotations.

6.0 Concluding Remarks and Future Work

In this code sprint, which is sponsored by Toyota in collaboration with Open Perception, homography estimation based on multi-descriptor correspondence sets has been explored, and inspired the introduction of the multi-descriptor voting approach (MDv) . The proposed MDv approach achieved a consistent accuracy in the 0.0X range, a level of consistency that is better than those based on single-type state of the art descriptors including SIFT. In the process, a framework for analyzing and evaluating single and multi-descriptor performance has been developed, and employed to validate the robustness of MDv, as compared with homography estimations based on a single descriptor type, as well as those based on RANSAC registration of best-K multi-descriptor correspondence sets. The outcome of this work is clearly promising, particularly to robotic applications that require long-term autonomy.

Based on the promising results of this code sprint, recommended future work items include:

- To explore residual sorting approaches as in [6].
- To apply colinearity testing or model verification, among other techniques, to detect and avoid degenerate configurations, as touched upon in [4,6]
- To define a per descriptor type confidence factor, that would act as a prior for the voting process. Although an attempt was made to define such a confidence factor, further extensive research is needed to take into account the mathematical definition of each descriptor type as well as scene-related characteristics, and error models.

7.0 References

1. Choi, Kim, Yu, “Performance Evaluation of RANSAC Family”, in Proceedings of the British Machine Vision Conference,1-12 (2009).
2. S. Taylor, E. Rosten, T. Drummond. Robust feature matching in 2.3µs. In IEEE CVPR Workshop on Feature Detectors and Descriptors: The State Of The Art and Beyond. 2009.
3. M. A. Fischler, R. C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Communications of the ACM, 24(6):381–395, 1981
4. Raguram, R., Frahm, J.M., Pollefeys, M.: A comparative analysis of RANSAC techniques leading to adaptive real-time random sample consensus. In: ECCV 2008
5. O. Chum and J. Matas. Matching with PROSAC - pro- gressive sample consensus. In C. Schmid, S. Soatto, and C. Tomasi, editors, *Proc. of Conference on Computer Vision and Pattern Recognition (CVPR)*, volume 1, pages 220–226, Los Alamitos, USA, June 2005. IEEE Computer Society
6. Chin, Yu, Suter: “Accelerated Hypothesis Generation for Multi-Structure Robust Fitting”, ECCV 2010.
7. E. Rublee, V. Rabaud, K. Konolige, G. R. Bradski. ORB: An efficient alternative to SIFT or SURF. In D. N. Metaxas, L. Quan, A. Sanfeliu, L. J. V. Gool, editors, ICCV, pp. 2564–2571. IEEE, 2011.
8. S. Leutenegger, M. Chli, R. Siegwart. BRISK: Binary Robust invariant scalable keypoints. In D. N. Metaxas, L. Quan, A. Sanfeliu, L. J. V. Gool, editors, ICCV, pp. 2548–2555. IEEE, 2011.

9. R. B. Rusu, N. Blodow, and M. Beetz, “Fast point feature histograms (FPFH) for 3D registration,” in IEEE International Conference on Robotics and Automation (ICRA), pp. 3212 –3217, May 2009

10. F. Tombari, S. Salti, L. Di Stefano, “Unique Signatures of Histograms for Local Surface Description“, in Proceedings of the 11th European Conference on Computer Vision (ECCV), Heraklion, Greece, September 5-11 2010.

11. F. Tombari, S. Salti, L. Di Stefano, „A Combined Texture-Shape Descriptor For Enhanced 3D Feature Matching“, in Proceedings of the 18th International Conference on Image Processing (ICIP), Brussels, Belgium, September 11-14 2011.

12. D. G. Lowe. Object recognition from local scale-invariant features. In International Conference on Computer Vision, 20–25 September, 1999, Kerkyra, Corfu, Greece, Proceedings, volume 2, pp. 1150–1157. 1999

13. D. G. Lowe. Distinctive Image Features from Scale-Invariant Keypoints. International Journal of Computer Vision, 60:91–110, 2004.

14. H. Bay, A. Ess, T. Tuytelaars, L. V. Gool. SURF: Speeded Up Robust Features. Computer Vision and Image Understanding (CVIU), 110:346–359, 2008.

15. E. Rosten, T. Drummond. Machine learning for high-speed corner detection. In European Conference on Computer Vision, volume 1, pp. 430–443. 2006.

16. E.Mair,G.D.Hager,D.Burschka,M.Suppa,G.Hirzinger. ”Adaptive and Generic Corner Detection Based on the Accelerated Segment Test”. In K. Daniilidis, P. Maragos, N. Paragios, editors, ECCV (2), volume 6312 of Lecture Notes in Computer Science, pp. 183–196. Springer, 2010.

17. M.R. Andersen, T. Jensen, P. Lisouski, A.K. Mortensen, M.K. Hansen, T. Gregersen and P. Ahrendt: Kinect Depth Sensor Evaluation for Computer Vision Applications, 2012. Department of Engineering, Aarhus University. Denmark. 37 pp. - Technical report ECE-TR-6

18. https://maps.google.com

19. http://www.space.com/17523-mars-rover-curiosity-self-portrait-photos.html

20. Open Source Computer Vision Library, http://opencv.org

21. Gaël Guennebaud, Benoît Jacob, et al. Eigen v3. http://eigen.tuxfamily.org, 2010.

22. Kempf, Williams, Escriba, Boost thread library. http://www.boost.org/docs/libs/1_55_0/libs/libraries.html

23. The Point Cloud Library, http://www.pointclouds.org

Appendix A: Correspondence Estimation, A Reference.

As the effectiveness of feature descriptors depends on the choice of correspondence estimation and filters used, a topology of the available correspondence estimation classes and filtering techniques in PCL 1.7 is presented here. For a high level treatment of the PCL registration API, please refer to the PCL registration API and tutorials.

Correspondence estimation attempts to match keypoints in a source cloud to keypoints in a target cloud, based on some similarity measure, feature descriptors in our case. Although applying scene relevant descriptor parameters and correspondence thresholds may reduce false matches, outliers persist with impact on pose estimation. This is due to the implied assumption that for each source keypoint, a corresponding target keypoint exists. The difficulty in estimating model or scene-specific descriptor parameters is another factor.

Below is a quick reference guide for the available correspondence estimation classes with remarks extracted from the PCL 1.7 source code. Please see the source code for more details.

Correspondence Estimation Class Name	**Remarks [extracted from PCL 1.7 source code]**
CorrespondenceEstimation	• Standard one and 2-way correspondence estimation. • Distance threshold is squared internally.
CorrespondenceEstimationOrganizedProjection	• Target cloud must be organized • Depth and distance thresholds can be applied • Defaults are set to Kinect's intrinsic values • No use of kd-trees • Fast, but less precise than kd-trees. • Only one-way correspondence estimation • 2-way doesn't improve the results (Auth.).
CorrespondenceEstimationBackProjection	• Correspondences are based on minimum perpendicular distance to the normal. • Requires normals for both source and target clouds. • Defaults to 10 nearest neighbor kd-tree search.
CorrespondenceEstimationNormalShooting	• Correspondences are based on the minimum distance between the normal on the source cloud and the corresponding point in the target cloud. • Defaults to 10 nearest neighbor kd-tree srch.

Appendix B: Correspondence Rejection, A Reference

Correspondence Rejection Class Name	Remarks [extracted from PCL source code]
CorrespondenceRejectorDistance	• Standard elimination using distance threshold. • The threshold distance is squared internally • Equivalent to distance threshold applied by the CorrespondenceEstimation routine.
CorrespondenceRejectorMedianDistance	Eliminates correspondences using median-distance as a threshold.
CorrespondenceRejectorOneToOne	Eliminates duplicate match indices in the correspondences, keeping ones with minimum distance measure.
CorrespondenceRejectorSampleConsensus	Implements correspondence rejection using RanSac.
CorrespondenceRejectorSampleConsensus2D	Pixel-based correspondence rejection using RanSac.
CorrespondenceRejectorTrimmed	Correspondence rejection based on the best K correspondences, where K is some estimate of the overlap between the source and the target. • Overlap ratio range: [0.0-1.0] • Reference „The Trimmed ICP Algorithm" by Chetverikov, Svirko, Stepanov, and Krsek.
CorrespondenceRejectorVarTrimmed	Rejection based on % of correspondences with least distances, as in „Outlier Robust ICP for minimizing Fractional RMSD, J.M. Philips et al."
CorrespondenceRejectorSurfaceNormal	Correspondence rejection method based on the angle between the normals at correspondence points.
CorrespondenceRejectorPoly	Rejection is performed by thresholding edge lengths, of virtual polygons that embody the correspondences. • Polygon's cardinality must be less than number of input correspondences. • Similarity threshold must be in the range of [0,1] • Similarity threshold is squared internally • Reference: „Pose Estimation using local Structure-Specific Shape and Appearance Context" by Buch, Kraft, Petersen, and Krueger.

Appendix C: The Evaluation Output for 0-60 degrees

Note: Multi-descriptor voting (MDv) is shown below with DescID 10, where the inlier rate does not apply due to the voting process.

```
==================================================
Configured Parameters:
max_rotation:    65.000
rotation_res:     5.000
thread_count: 2
best_k_sorted_2D: 100
best_k_sorted_3D: 140
leaf_size: 0.01
vote_threshold: 0.20
ransac_inlier_threshold_2D:     0.100
ransac_inlier_threshold_3D:     0.010
enable_graphics: 0
output_level: 1
live_sensor: 0
==================================================
Vote List: 0, avg:   -0.005, Vote Count: 8, err:    0.005, desIDs:   0  1  2  3  5  6  7  8
------------------------------------------------------------
Vote List: 1, avg:   -0.239, Vote Count: 1, err:    0.239, desIDs:   4
------------------------------------------------------------
Vote List: 2, avg:    0.642, Vote Count: 1, err:    0.642, desIDs:   9
------------------------------------------------------------
^^^^ ChosenVote: Rot:   -0.005 , Err:    0.005, Count: 8
==================================================================================
DescID ComputedRot EstError InlierRate ObjfuncEstErr ObjFuncRot numVotes
==================================================================================
   3    0.000       0.000     1.000         0.005      -0.005      8
   2   -0.003       0.003     0.690         0.005      -0.005      8
  10   -0.005       0.005     0.000         0.005      -0.005      8
   0   -0.010       0.010     0.310         0.005      -0.005      8
   6   -0.021       0.021     0.829         0.005      -0.005      8
   1    0.028       0.028     0.360         0.005      -0.005      8
   7    0.035       0.035     0.493         0.005      -0.005      8
   8    0.061       0.061     0.736         0.005      -0.005      8
   5   -0.128       0.128     0.750         0.005      -0.005      8
   4   -0.239       0.239     0.636         0.005      -0.005      8
   9    0.642       0.642     0.813         0.005      -0.005      8
=================================================================================
Vote List: 0, avg:    5.007, Vote Count: 7, err:    0.007, desIDs:   0  1  5  6  7  8  9
------------------------------------------------------------
Vote List: 1, avg:    5.794, Vote Count: 1, err:    0.794, desIDs:   2
------------------------------------------------------------
Vote List: 2, avg:    4.815, Vote Count: 2, err:    0.185, desIDs:   3  4
------------------------------------------------------------
^^^^ ChosenVote: Rot:    5.007 , Err:    0.007, Count: 7
==================================================================================
DescID ComputedRot EstError InlierRate ObjfuncEstErr ObjFuncRot numVotes
==================================================================================
  10    5.007       0.007     0.000         0.007       5.007      7
   6    4.969       0.031     0.757         0.007       5.007      7
   7    5.040       0.040     0.893         0.007       5.007      7
   8    4.951       0.049     0.671         0.007       5.007      7
   5    4.917       0.083     0.679         0.007       5.007      7
   0    5.084       0.084     0.220         0.007       5.007      7
   1    4.911       0.089     0.140         0.007       5.007      7
   4    4.872       0.128     0.593         0.007       5.007      7
   9    5.174       0.174     0.697         0.007       5.007      7
   3    4.757       0.243     0.080         0.007       5.007      7
   2    5.794       0.794     0.070         0.007       5.007      7
=================================================================================
```

```
Vote List: 0, avg:   10.015, Vote Count: 7, err:    0.015, desIDs:   0  1  2  5  6  7  8
-----------------------------------------------------------------------
Vote List: 1, avg:   12.430, Vote Count: 1, err:    2.430, desIDs:   3
-----------------------------------------------------------------------
Vote List: 2, avg:   10.276, Vote Count: 2, err:    0.276, desIDs:   4  9
-----------------------------------------------------------------------
^^^^ ChosenVote: Rot:   10.015 , Err:    0.015, Count: 7
=================================================================================
DescID ComputedRot EstError InlierRate ObjFuncEstErr ObjFuncRot numVotes
=================================================================================
   0    9.989       0.011     0.220          0.015     10.015     7
  10   10.015       0.015     0.000          0.015     10.015     7
   7   10.037       0.037     0.065          0.015     10.015     7
   6    9.942       0.058     0.786          0.015     10.015     7
   5   10.064       0.064     0.787          0.015     10.015     7
   2   10.077       0.077     0.070          0.015     10.015     7
   8   10.134       0.134     0.681          0.015     10.015     7
   1    9.858       0.142     0.110          0.015     10.015     7
   9   10.225       0.225     0.706          0.015     10.015     7
   4   10.327       0.327     0.557          0.015     10.015     7
   3   12.430       2.430     0.000          0.015     10.015     7
================================================================================
Vote List: 0, avg:   14.982, Vote Count: 7, err:    0.018, desIDs:   0  1  2  4  5  6  7
-----------------------------------------------------------------------
Vote List: 1, avg:   15.293, Vote Count: 2, err:    0.293, desIDs:   3  8
-----------------------------------------------------------------------
Vote List: 2, avg:   15.672, Vote Count: 1, err:    0.672, desIDs:   9
-----------------------------------------------------------------------
^^^^ ChosenVote: Rot:   14.982 , Err:    0.018, Count: 7
=================================================================================
DescID ComputedRot EstError InlierRate ObjFuncEstErr ObjFuncRot numVotes
=================================================================================
   4   14.982       0.018     0.550          0.018     14.982     7
  10   14.982       0.018     0.000          0.018     14.982     7
   6   14.979       0.021     0.771          0.018     14.982     7
   0   14.942       0.058     0.250          0.018     14.982     7
   1   15.061       0.061     0.090          0.018     14.982     7
   7   14.934       0.066     0.065          0.018     14.982     7
   5   15.082       0.082     0.686          0.018     14.982     7
   2   14.892       0.108     0.070          0.018     14.982     7
   3   15.276       0.276     0.000          0.018     14.982     7
   8   15.311       0.311     0.669          0.018     14.982     7
   9   15.672       0.672     0.719          0.018     14.982     7
================================================================================
Vote List: 0, avg:   19.989, Vote Count: 6, err:    0.011, desIDs:   0  1  5  6  7  8
-----------------------------------------------------------------------
Vote List: 1, avg:   19.654, Vote Count: 2, err:    0.346, desIDs:   2  9
-----------------------------------------------------------------------
Vote List: 2, avg:   20.285, Vote Count: 2, err:    0.285, desIDs:   3  4
-----------------------------------------------------------------------
^^^^ ChosenVote: Rot:   19.989 , Err:    0.011, Count: 6
=================================================================================
DescID ComputedRot EstError InlierRate ObjFuncEstErr ObjFuncRot numVotes
=================================================================================
   0   19.999       0.001     0.230          0.011     19.989     6
  10   19.989       0.011     0.000          0.011     19.989     6
   7   20.013       0.013     0.055          0.011     19.989     6
   6   20.016       0.016     0.750          0.011     19.989     6
   8   19.943       0.057     0.660          0.011     19.989     6
   5   20.157       0.157     0.671          0.011     19.989     6
   1   19.804       0.196     0.000          0.011     19.989     6
   4   20.230       0.230     0.564          0.011     19.989     6
   2   19.709       0.291     0.060          0.011     19.989     6
   3   20.340       0.340     0.100          0.011     19.989     6
   9   19.599       0.401     0.808          0.011     19.989     6
================================================================================
```

```
Vote List: 0, avg:   24.961, Vote Count: 7, err:    0.039, desIDs:   0  1  5  6  7  8  9
------------------------------------------------------------------
Vote List: 1, avg:   25.233, Vote Count: 2, err:    0.233, desIDs:   2  4
------------------------------------------------------------------
Vote List: 2, avg:   25.806, Vote Count: 1, err:    0.806, desIDs:   3
------------------------------------------------------------------
^^^^ ChosenVote: Rot:   24.961 , Err:    0.039, Count: 7
==========================================================================
DescID ComputedRot EstError InlierRate ObjFuncEstErr ObjFuncRot numVotes
==========================================================================
   0   24.995       0.005     0.220         0.039      24.961      7
   6   25.025       0.021     0.764         0.039      24.961      7
   1   24.971       0.029     0.070         0.039      24.961      7
   9   24.967       0.033     0.700         0.039      24.961      7
  10   24.961       0.039     0.000         0.039      24.961      7
   5   25.082       0.082     0.693         0.039      24.961      7
   7   24.855       0.145     0.043         0.039      24.961      7
   8   24.840       0.160     0.674         0.039      24.961      7
   2   25.202       0.202     0.060         0.039      24.961      7
   4   25.264       0.264     0.536         0.039      24.961      7
   3   25.806       0.806     0.070         0.039      24.961      7
==========================================================================
Vote List: 0, avg:   30.032, Vote Count: 6, err:    0.032, desIDs:   0  1  5  6  7  8
------------------------------------------------------------------
Vote List: 1, avg:   29.424, Vote Count: 1, err:    0.576, desIDs:   2
------------------------------------------------------------------
Vote List: 2, avg:   30.579, Vote Count: 1, err:    0.579, desIDs:   3
------------------------------------------------------------------
Vote List: 3, avg:   29.731, Vote Count: 2, err:    0.269, desIDs:   4  9
------------------------------------------------------------------
^^^^ ChosenVote: Rot:   30.032 , Err:    0.032, Count: 6
==========================================================================
DescID ComputedRot EstError InlierRate ObjFuncEstErr ObjFuncRot numVotes
==========================================================================
   1   30.005       0.005     0.060         0.032      30.032      6
   0   30.020       0.020     0.250         0.032      30.032      6
  10   30.032       0.032     0.000         0.032      30.032      6
   8   29.968       0.032     0.662         0.032      30.032      6
   7   30.062       0.062     0.043         0.032      30.032      6
   6   29.916       0.084     0.743         0.032      30.032      6
   5   30.219       0.219     0.693         0.032      30.032      6
   9   29.760       0.240     0.545         0.032      30.032      6
   4   29.702       0.298     0.536         0.032      30.032      6
   2   29.424       0.576     0.070         0.032      30.032      6
   3   30.579       0.579     0.080         0.032      30.032      6
==========================================================================
Vote List: 0, avg:   35.049, Vote Count: 5, err:    0.049, desIDs:   0  2  6  7  8
------------------------------------------------------------------
Vote List: 1, avg:   35.480, Vote Count: 2, err:    0.480, desIDs:   1  3
------------------------------------------------------------------
Vote List: 2, avg:   34.711, Vote Count: 2, err:    0.289, desIDs:   4  9
------------------------------------------------------------------
Vote List: 3, avg:   35.225, Vote Count: 1, err:    0.225, desIDs:   5
------------------------------------------------------------------
^^^^ ChosenVote: Rot:   35.049 , Err:    0.049, Count: 5
==========================================================================
DescID ComputedRot EstError InlierRate ObjFuncEstErr ObjFuncRot numVotes
==========================================================================
   0   35.015       0.015     0.210         0.049      35.049      5
   2   34.967       0.033     0.080         0.049      35.049      5
   7   35.041       0.041     0.043         0.049      35.049      5
   6   35.048       0.048     0.757         0.049      35.049      5
  10   35.049       0.049     0.000         0.049      35.049      5
   8   35.172       0.172     0.657         0.049      35.049      5
   5   35.225       0.225     0.686         0.049      35.049      5
   4   34.719       0.281     0.564         0.049      35.049      5
   9   34.704       0.296     0.765         0.049      35.049      5
   3   35.452       0.452     0.080         0.049      35.049      5
   1   35.507       0.507     0.070         0.049      35.049      5
==========================================================================
```

```
Vote List: 0, avg:   40.032, Vote Count: 7, err:    0.032, desIDs:   0  1  2  5  6  7  8
------------------------------------------------------------------
Vote List: 1, avg:   40.805, Vote Count: 1, err:    0.805, desIDs:   3
------------------------------------------------------------------
Vote List: 2, avg:   40.290, Vote Count: 1, err:    0.290, desIDs:   4
------------------------------------------------------------------
Vote List: 3, avg:   39.774, Vote Count: 1, err:    0.226, desIDs:   9
------------------------------------------------------------------
^^^^ ChosenVote: Rot:   40.032 , Err:    0.032, Count: 7
==============================================================================
DescID ComputedRot EstError InlierRate ObjFuncEstErr ObjFuncRot numVotes
==============================================================================
   6   40.012       0.012     0.764         0.032      40.032     7
   0   40.017       0.017     0.190         0.032      40.032     7
  10   40.032       0.032     0.000         0.032      40.032     7
   7   39.926       0.074     0.038         0.032      40.032     7
   1   40.001       0.001     0.060         0.032      40.032     7
   5   39.892       0.108     0.714         0.032      40.032     7
   8   40.131       0.131     0.688         0.032      40.032     7
   2   40.165       0.165     0.060         0.032      40.032     7
   9   39.774       0.226     0.758         0.032      40.032     7
   4   40.290       0.290     0.543         0.032      40.032     7
   3   40.805       0.805     0.080         0.032      40.032     7
==============================================================================
Vote List: 0, avg:   44.963, Vote Count: 7, err:    0.037, desIDs:   0  1  5  6  7  8  9
------------------------------------------------------------------
Vote List: 1, avg:   45.989, Vote Count: 1, err:    0.989, desIDs:   2
------------------------------------------------------------------
Vote List: 2, avg:   45.577, Vote Count: 1, err:    0.577, desIDs:   3
------------------------------------------------------------------
Vote List: 3, avg:   45.215, Vote Count: 1, err:    0.215, desIDs:   4
------------------------------------------------------------------
^^^^ ChosenVote: Rot:   44.963 , Err:    0.037, Count: 7
==============================================================================
DescID ComputedRot EstError InlierRate ObjFuncEstErr ObjFuncRot numVotes
==============================================================================
   0   45.015       0.015     0.260         0.037      44.963     7
   7   45.023       0.023     0.047         0.037      44.963     7
  10   44.963       0.037     0.000         0.037      44.963     7
   1   44.957       0.043     0.080         0.037      44.963     7
   6   44.904       0.096     0.757         0.037      44.963     7
   9   44.890       0.110     0.688         0.037      44.963     7
   8   45.117       0.117     0.671         0.037      44.963     7
   5   44.833       0.167     0.664         0.037      44.963     7
   4   45.215       0.215     0.550         0.037      44.963     7
   3   45.577       0.577     0.070         0.037      44.963     7
   2   45.989       0.989     0.060         0.037      44.963     7
==============================================================================
Vote List: 0, avg:   50.069, Vote Count: 6, err:    0.069, desIDs:   0  2  5  6  7  8
------------------------------------------------------------------
Vote List: 1, avg:   49.826, Vote Count: 3, err:    0.174, desIDs:   1  4  9
------------------------------------------------------------------
Vote List: 2, avg:   49.503, Vote Count: 1, err:    0.497, desIDs:   3
------------------------------------------------------------------
^^^^ ChosenVote: Rot:   50.069 , Err:    0.069, Count: 6
==============================================================================
DescID ComputedRot EstError InlierRate ObjFuncEstErr ObjFuncRot numVotes
==============================================================================
   6   49.984       0.016     0.771         0.069      50.069     6
   2   49.984       0.016     0.060         0.069      50.069     6
   5   50.052       0.052     0.707         0.069      50.069     6
  10   50.069       0.069     0.000         0.069      50.069     6
   7   50.077       0.077     0.043         0.069      50.069     6
   0   50.133       0.133     0.220         0.069      50.069     6
   1   49.840       0.160     0.070         0.069      50.069     6
   9   49.830       0.170     0.750         0.069      50.069     6
   8   50.182       0.182     0.664         0.069      50.069     6
   4   49.809       0.191     0.536         0.069      50.069     6
   3   49.503       0.497     0.070         0.069      50.069     6
==============================================================================
```

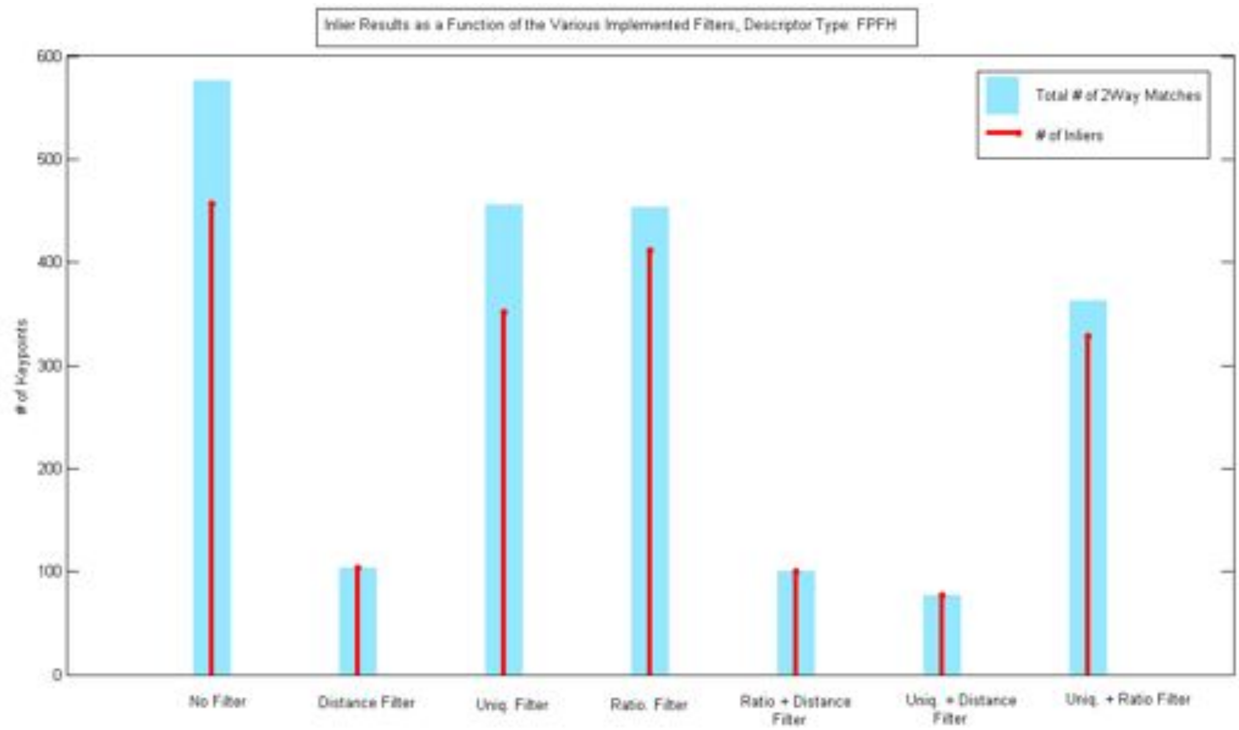
```
Vote List: 0, avg:   55.053, Vote Count: 6, err:    0.053, desIDs:   0  1  4  6  8  9
----------------------------------------------------------------------
Vote List: 1, avg:   54.867, Vote Count: 3, err:    0.133, desIDs:   2  5  7
----------------------------------------------------------------------
Vote List: 2, avg:   54.424, Vote Count: 1, err:    0.576, desIDs:   3
----------------------------------------------------------------------
^^^^ ChosenVote: Rot:   55.053 , Err:    0.053, Count: 6
===============================================================================
DescID ComputedRot EstError InlierRate ObjFuncEstErr ObjFuncRot numVotes
===============================================================================
   9   54.998       0.002      0.828          0.053      55.053     6
   1   54.977       0.023      0.070          0.053      55.053     6
   6   54.957       0.043      0.750          0.053      55.053     6
  10   55.053       0.053      0.000          0.053      55.053     6
   5   54.880       0.120      0.671          0.053      55.053     6
   8   55.120       0.120      0.664          0.053      55.053     6
   4   55.125       0.125      0.536          0.053      55.053     6
   2   54.869       0.131      0.060          0.053      55.053     6
   0   55.139       0.139      0.190          0.053      55.053     6
   7   54.852       0.148      0.040          0.053      55.053     6
   3   54.424       0.576      0.070          0.053      55.053     6
===============================================================================
Vote List: 0, avg:   59.964, Vote Count: 8, err:    0.036, desIDs:   0  2  4  5  6  7  8  9
----------------------------------------------------------------------
Vote List: 1, avg:   58.935, Vote Count: 1, err:    1.065, desIDs:   1
----------------------------------------------------------------------
Vote List: 2, avg:   59.271, Vote Count: 1, err:    0.729, desIDs:   3
----------------------------------------------------------------------
^^^^ ChosenVote: Rot:   59.964 , Err:    0.036, Count: 8
===============================================================================
DescID ComputedRot EstError InlierRate ObjFuncEstErr ObjFuncRot numVotes
===============================================================================
   5   60.008       0.008      0.679          0.036      59.964     8
   7   59.991       0.009      0.038          0.036      59.964     8
   0   59.985       0.015      0.210          0.036      59.964     8
  10   59.964       0.036      0.000          0.036      59.964     8
   6   60.061       0.061      0.729          0.036      59.964     8
   4   60.123       0.123      0.543          0.036      59.964     8
   2   59.875       0.125      0.060          0.036      59.964     8
   8   59.861       0.139      0.655          0.036      59.964     8
   9   59.810       0.190      0.767          0.036      59.964     8
   3   59.271       0.729      0.070          0.036      59.964     8
   1   58.935       1.065      0.060          0.036      59.964     8
===============================================================================
```

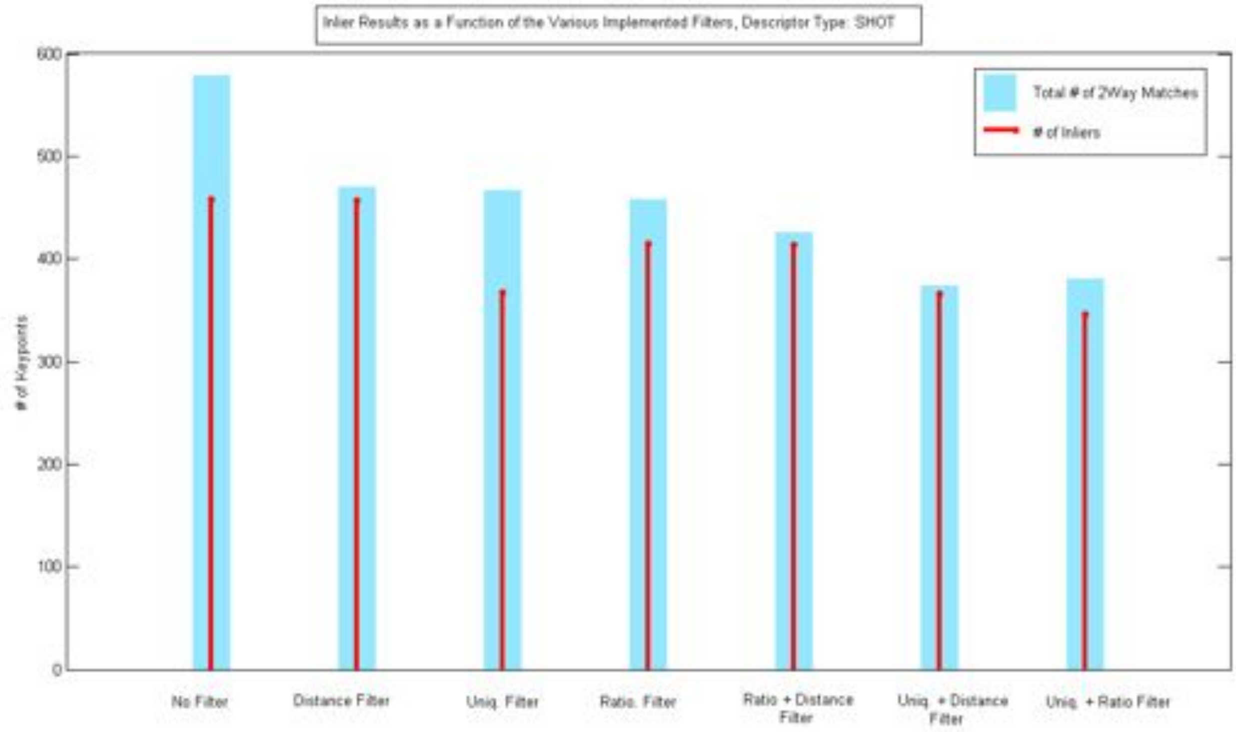
Appendix D: The Evaluation Output for the First Milestone

As mentioned in the roadmap, one of the goals is to implement a framework that captures vital statistics of selected descriptors and correspondence types. These vital statistics would then be fed to one or more objective function(s) to enable scene based optimizations.

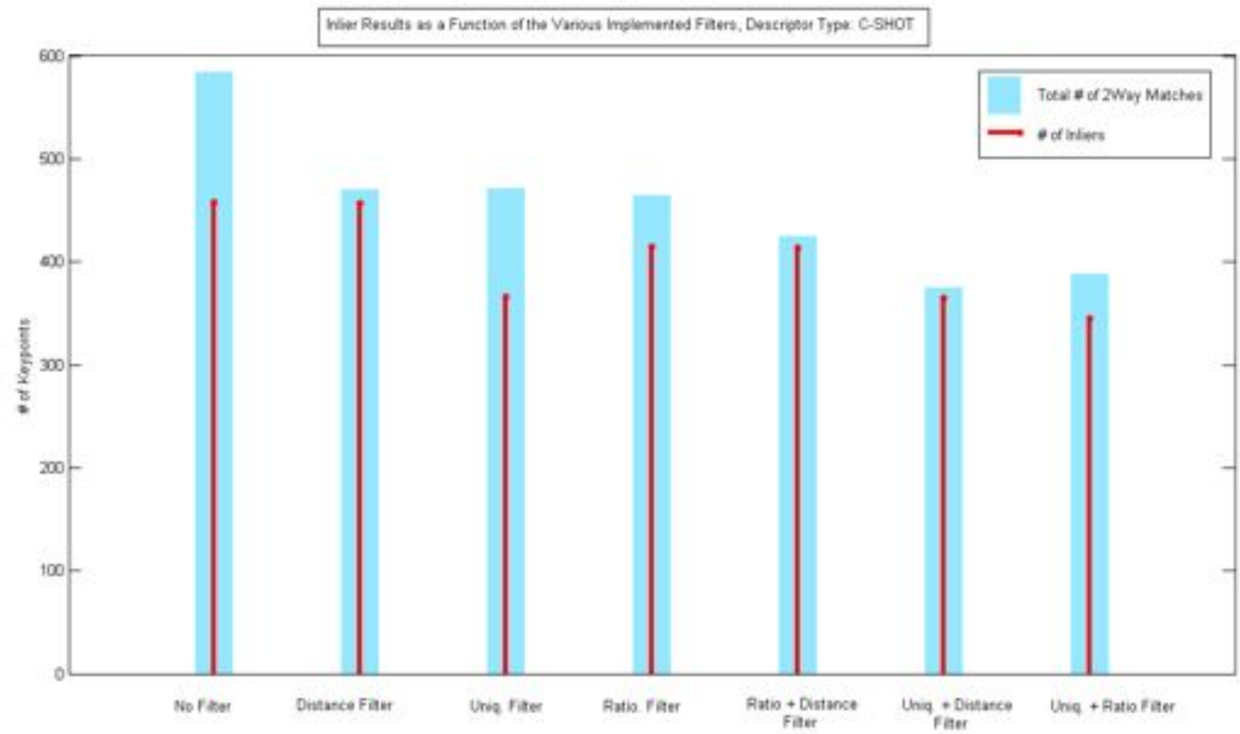
The framework milestone for 3D descriptors is now complete, and its output is in-line with the characteristics cited in Rublee et. al. ICCV 2011 paper.

Specifically, the framework computes the intended vital statistics including: 2-Way and multi-descriptor matching and inlier rates. The filter banks include L2-distance, L2-ratio, and uniqueness measure. A simulated ground truth is also implemented and is generated during runtime. The framework is tested and applied to a set of 3D descriptors (FPFH, SHOT352, and SHOT1344) across a range of downsampling leaf-sizes (0.01-0.07) and across a range of in-plane (0-90 degrees) rotations. A sample of the results is illustrated in the charts below, which reflect the various metrics, computed at a 30 degree simulated rotation and at 2 levels of downsampling: 0.01 for the top 3 charts, and 0.07 for the bottom 3 charts. In total, 1680 rates were computed for analysis by the objective function(s) for further optimizations.

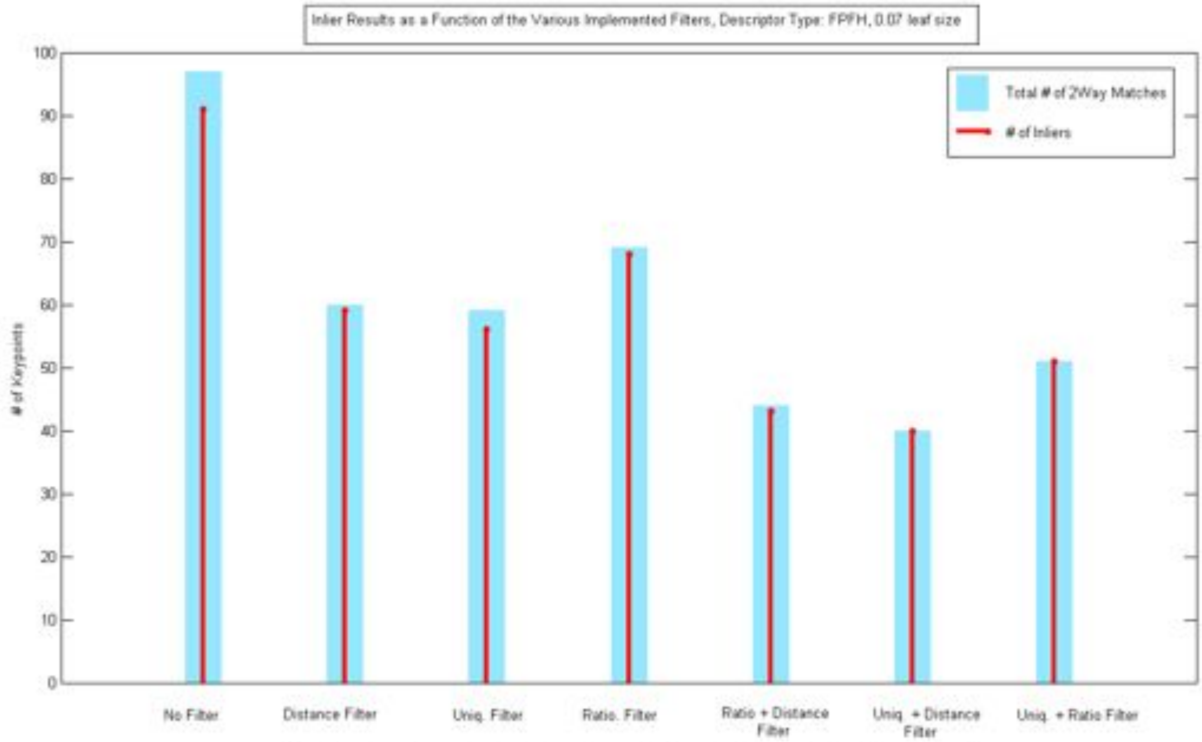


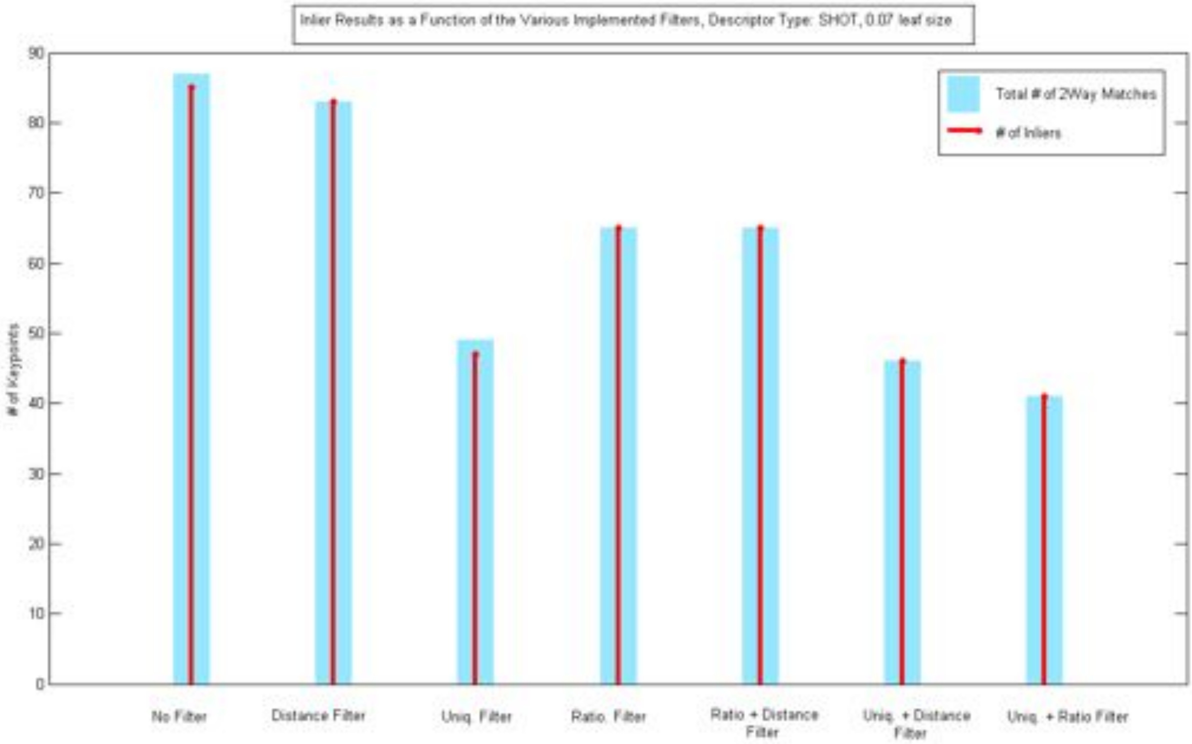


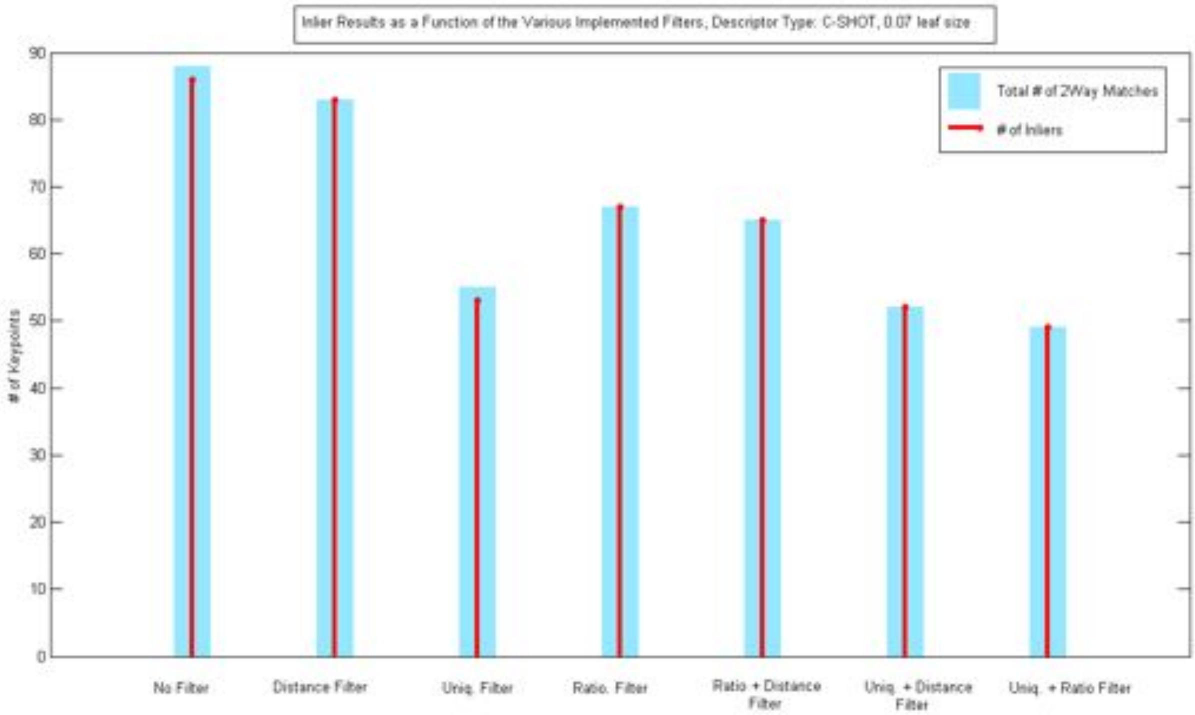
Inlier Results as a Function of the Various Implemented Filters, Descriptor Type: SHOT
Total # of 2Way Matches
# of Inliers
# of Keypoints
600
500
400
300
200
100
0
No Filter
Distance Filter
Uniq. Filter
Ratio. Filter
Ratio + Distance Filter
Uniq. + Distance Filter
Uniq. + Ratio Filter



Inlier Results as a Function of the Various Implemented Filters, Descriptor Type: C-SHOT
Total # of 2Way Matches
# of Inliers
# of Keypoints
600
500
400
300
200
100
0
No Filter
Distance Filter
Uniq. Filter
Ratio. Filter
Ratio + Distance Filter
Uniq. + Distance Filter
Uniq. + Ratio Filter


Inlier Results as a Function of the Various Implemented Filters, Descriptor Type: FPFH, 0.07 leaf size
Total # of 2Way Matches
# of Inliers
# of Keypoints
No Filter
Distance Filter
Uniq. Filter
Ratio. Filter
Ratio + Distance Filter
Uniq. + Distance Filter
Uniq. + Ratio Filter


Inlier Results as a Function of the Various Implemented Filters, Descriptor Type: SHOT, 0.07 leaf size
Total # of 2Way Matches
# of Inliers
# of Keypoints
No Filter
Distance Filter
Uniq. Filter
Ratio. Filter
Ratio + Distance Filter
Uniq. + Distance Filter
Uniq. + Ratio Filter



Inlier Results as a Function of the Various Implemented Filters, Descriptor Type: C-SHOT, 0.07 leaf size
Total # of 2Way Matches
# of Inliers
# of Keypoints
0
10
20
30
40
50
60
70
80
90
No Filter
Distance Filter
Uniq. Filter
Ratio. Filter
Ratio + Distance Filter
Uniq. + Distance Filter
Uniq. + Ratio Filter