

**3D MULTI-FIELD MULTI-SCALE FEATURES FROM RANGE DATA
IN SPACECRAFT PROXIMITY OPERATIONS**

A Dissertation

by

BRIEN ROY FLEWELLING

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

May 2012

Major Subject: Aerospace Engineering

3D Multi-Field Multi-Scale Features from Range Data in Spacecraft Proximity
Operations

Copyright 2012 Brien Roy Flewelling

**3D MULTI-FIELD MULTI-SCALE FEATURES FROM RANGE DATA
IN SPACECRAFT PROXIMITY OPERATIONS**

A Dissertation

by

BRIEN ROY FLEWELLING

Submitted to the Office of Graduate Studies of
Texas A&M University
in partial fulfillment of the requirements for the degree of

DOCTOR OF PHILOSOPHY

Approved by:

Co-Chairs of Committee,	John L. Junkins
	Daniele Mortari
Committee Members,	Tom Pollock
	Nancy Amato
Head of Department,	Dimitris Lagoudas

May 2012

Major Subject: Aerospace Engineering

ABSTRACT

3D Multi-Field Multi-Scale Features from Range Data in Spacecraft Proximity
Operations. (May 2012)

Brien Roy Flewelling, B.S., Syracuse University

Co-Chairs of Advisory Committee: Dr. John L. Junkins
Dr. Daniele Mortari

A fundamental problem in spacecraft proximity operations is the determination of the 6 degree of freedom relative navigation solution between the observer reference frame and a reference frame tied to a proximal body. For the most unconstrained case, the proximal body may be uncontrolled, and the observer spacecraft has no *a priori* information on the body. A spacecraft in this scenario must simultaneously map the generally poorly known body being observed, and safely navigate relative to it. Simultaneous localization and mapping (SLAM) is a difficult problem which has been the focus of research in recent years. The most promising approaches extract local features in 2D or 3D measurements and track them in subsequent observations by means of matching a *descriptor*. These methods exist for both active sensors such as Light Detection and Ranging (LIDAR) or laser RADAR (LADAR), and passive sensors such as CCD and CMOS camera systems. This dissertation presents a method for fusing time of flight (ToF) range data inherent to scanning LIDAR systems with the passive light field measurements of optical systems, extracting features which exploit information from each sensor, and solving the unique SLAM problem inherent to spacecraft proximity operations. Scale Space analysis is extended to unstructured 3D

point clouds by means of an approximation to the Laplace Beltrami operator which computes the scale space on a manifold embedded in 3D object space using Gaussian convolutions based on a geodesic distance weighting. The construction of the scale space is shown to be equivalent to both the application of the diffusion equation to the surface data, as well as the surface evolution process which results from mean curvature flow. Geometric features are localized in regions of high spatial curvature or large diffusion displacements at multiple scales. The extracted interest points are associated with a local multi-field descriptor constructed from measured data in the object space. Defining features in object space instead of image space is shown to be an important step making the simultaneous consideration of co-registered texture and the associated geometry possible. These descriptors known as *Multi-Field Diffusion Flow Signatures* encode the shape, and multi-texture information of local neighborhoods in textured range data. Multi-Field Diffusion Flow Signatures display utility in difficult space scenarios including high contrast and saturating lighting conditions, bland and repeating textures, as well as non-Lambertian surfaces. The effectiveness and utility of *Multi-Field Multi-Scale (MFMS)* Features described by Multi-Field Diffusion Flow Signatures is evaluated using real data from proximity operation experiments performed at the Land Air and Space Robotics (LASR) Laboratory at Texas A&M University.

To my Wife and Family, I am truly blessed.

ACKNOWLEDGEMENTS

It should be noted that each physical quantity, known or unknown, possesses a dimension proper to itself and that the terms in an equation cannot be compared one with another unless they possess the same dimensional exponent. - Baron Jean-Baptiste Fourier

I would first and foremost like to acknowledge my advisor Dr. John L. Junkins who accepted me into his distinguished group of outstanding students and researchers and has challenged me to be more than I have ever considered I could be. Learning from you a kind of critical thinking that is at the same time technically insightful as well as practical and applied has been more valuable to me and my educational development than any individual class or research experience. The technical challenges we have faced while getting into the world of “inventions” and developing a coherent approach to combining data processing, feature extraction, and motion estimation have been demanding and I have enjoyed working on a real problem with a high potential for space based applications. Thank you for allowing me to develop in my own way while guiding me on a path that has gotten me where I am today. I am forever grateful for everything you have provided me in my time at Texas A&M University including your support and concern for both me and my wife throughout the years. I am proud to be a part of those who are lucky enough to have spent an extended period of time learning from you and I hope that in the years to come I too will achieve the level of excellence that so many of your students have in their careers. I would also like to thank my Co-advisor Dr. Daniele Mortari for impressing

on me a love and enthusiasm for attitude estimation and image processing as well as many other algorithmic techniques used in our field. I am grateful for your support and direction and am inspired by your courage, strength, and genuine love for what you do. I sincerely hope to continue to learn from you throughout my career. There are many other professors who have listened and helped me to develop throughout my time at TAMU especially Dr. Tom Pollock and Dr. James Turner. Dr. Pollock, thank you for the many discussions and in lab sessions we have had together, I have learned a lot from you about how to work from the bottom up starting with the sensor itself. The amount of knowledge you have shared with me in the areas of applied optics and image processing will serve me well and I hope to continue discussing new sensor technology and processing methods with you in the future. Dr. Turner, I am forever grateful for the many hours you spent listening to me formulate and reject countless ideas and approaches on your white board. Sometimes the most valuable thing an educator can be is an effective listener and I firmly believe that having that in you was a key reason I was able to reach the results I present in this dissertation. I am also grateful to my fellow graduate students for the many experiences we have shared as well as for their critical support throughout my time in Aggieland. Especially the kind support of Dr. Manoranjan Majji who was truly instrumental in my technical development and with whom I have shared many discussions, debates, and experiences throughout our journey into the world of applied computer vision. Majji, you are an amazing technical mind and I am proud to have worked with you and to have learned from you in the process. I would also like to acknowledge the many years of support provided to me by the American Society for Engineering Education

through their Science, Mathematics, and Research for Transformation scholarship which I was very fortunate to receive. It is through this support that I was able to focus directly on my research. Additionally, the summer internships at the Naval Air Warfare Center in China Lake as well as the Air Force Research Laboratories Space Vehicles Directorate at Kirtland AFB, were excellent experiences which exposed me to many experts in the field and finally resulted in my position in the Guidance, Navigation and Controls group at AFRL where I am eager to continue my research as I start my professional career. Thank you to everyone who has listened to me stumble through the many challenges of this project, and also to everyone who has afforded me their courteous patience and flexibility with regard to this research and my education. Finally, I would like to expressly thank my wife, Kayla who, throughout all of the many ups and downs of my experience at Texas A&M has supported me and been my partner. The most positive result of all of my time in Texas is you.

TABLE OF CONTENTS

	Page
ABSTRACT	iii
DEDICATION	v
ACKNOWLEDGEMENTS	vi
TABLE OF CONTENTS	ix
LIST OF FIGURES	xii
LIST OF TABLES	xxiii
CHAPTER	
I INTRODUCTION	1
A. Aim and Objectives	3
1. Scientific Questions	5
2. Motivation	7
B. Background	9
1. A Brief History of Spacecraft Proximity Operations	9
2. Developments in 2D and 3D Data Processing	11
C. Overview and Approach	12
1. Hypothesis	12
2. Method	13
3. Contributions	14
4. Overview of Dissertation	15
II BACKGROUND AND LITERATURE REVIEW	17
A. Developments in Image Feature Extraction	17
1. Basic Image Structures and Gradients	17
2. Corner Detection	22
3. Multi-Resolution Image Analysis: The Scale Space	25
4. Image Pyramids	28
5. Invariance Principles and Feature Descriptors	31
6. The Scale Invariant Feature Transform	33
B. Developments in 3D Feature Extraction	39

CHAPTER		Page
	1. Smoothing 3D Data	40
	2. Differential Geometry	45
	3. 3D Point Based Registration by Spin Images	50
	4. Applying the Scale Invariant Feature Transform to Range Data	52
III	SENSOR CHARACTERIZATION AND SYSTEM CALIBRATION .	54
	A. Measuring Geometry by Light Detection and Ranging . . .	55
	B. Passive Field Measurements from Camera Imagery	58
	C. Camera to LIDAR Extrinsic Calibration	62
	1. The Vis-Nav Solution	64
IV	SCALE SPACE ANALYSIS OF MULTI-FIELD DATA ON MEASURED MANIFOLDS	73
	A. Geodesic Gaussian Convolution for Simultaneous Sur- face Evolution and Texture Analysis	90
	B. Feature Extraction in Multi-Field Data	97
V	MULTI-FIELD DIFFUSION FLOW SIGNATURES FOR RO- BUST 3D REGISTRATION	102
	A. Multi-Field Diffusion Flow Signatures	107
	B. Matching Diffusion Flow Signatures: Approach and Remaining Challenges	113
	C. OLTAE: The Optimal Linear Attitude and Transla- tion Estimator	119
VI	MFMS TRACKER: MULTI-RESOLUTION ANALYSIS OF MULTI-FIELD DATA FOR PROXIMITY OPERATIONS	128
	A. MFMSTracker Software	128
	B. Combined Model Registration and Relative Motion Estimation: MergeModels.m	138
VII	EXPERIMENTAL RESULTS	141
	A. Comparison with 2D Multi-Scale Feature Extraction . . .	142
	1. Complete Ground Scene	144
	2. Space Object Image	154
	3. Multi-Field SIFT vs. MFMS on the Hubble Space Telescope Model	160
	B. HST Model Case Study	166

CHAPTER	Page
1. Experimental Setup	167
2. Registration Using Scale Invariant Feature Transform	170
3. Registration Using Multi-Field Multi-Scale Features .	172
4. Registration using MFMS Features and OLTAE	
Residual Filtering	174
VIII CONCLUSIONS AND FUTURE WORK	178
A. Conclusions	179
B. Remaining Questions and Future Work	184
REFERENCES	186
VITA	192

LIST OF FIGURES

FIGURE	Page
1 General feature detection and tracking framework for opto-LIDAR fusion systems	14
2 The Sobel X and Y derivative operators.	20
3 The Gaussian Kernel and Partial Derivative Operators up to 2nd order.	23
4 The distribution of image structures as a function of the eigenvalues of the structure tensor.	25
5 Example Gaussian Pyramid: The image is repeatedly low pass filtered and re-sampled by a factor of two.	29
6 Laplacian Image Pyramid built by computing differences between original and expanded levels of the Gaussian Pyramid. For visualization purposes the absolute value is shown for each level.	29
7 The Laplacian of Gaussian operator is the result of applying the discrete isotropic second derivative filter to the Gaussian Operator. Its representation in spatial and frequency domain is shown where it represents a pass band filter.	30
8 Candidate corner feature matches in a stereo pair with local area descriptors compared by correlation.	32
9 Non-Maxima Suppression of the Scale Space: Each pyramid location is compared to its 26 neighbors and required to be greater than or less than all of them by a threshold.	35
10 Local sampling of gradient orientations occurs in the appropriate pyramid level associated with the feature scale, and with respect to the reference orientation of the feature.	38

FIGURE		Page
11	An example of $3D$ data representations, Left: a point cloud, Middle: Voxels representing volume data, Right: A surface mesh resulting from a Delaunay Triangulation	39
12	An example of $3D$ neighborhood definitions, Left: a spherical radius with 9 interior points, Right: $K = 20$ nearest neighbors containing 1 point at a large distance with the $K = 9$ neighborhood shown for comparison.	42
13	An example of Laplacian smoothing applied to surface data	44
14	Surfaces types classified by Mean and Gaussian Curvatures as indicated by Table 2.	47
15	An example Oriented Point consisting of a position and direction in space. In this basis only the in-plane radial distance α and the axial distance from the local tangent plane β are needed to describe a position in this frame.	51
16	A two axis scanning mirror as the origin of a spherical coordinate system.	56
17	This diagram illustrates the slight variation from traditional spherical coordinates to orthogonal axis coordinates.	57
18	An example range image, a $2D$ array of range values taken by a stationary scanner.	58
19	Return Light Intensity Data for the Asteroid Range Image.	59
20	A simple representation of a “Camera Obscura.”	59
21	The Colinearity equations describe the pinhole camera with respect to an arbitrary reference frame.	61
22	$2D$ - $3D$ point correspondence by associating the center of a best fit sphere to the centroid of the spheres image.	67

FIGURE		Page
23	The Deltasphere 3000 scanner operates by pulsing a laser against a mirror which rotates along the horizontal axis, the entire apparatus is rotated by a motor along the vertical axis resulting in a raster scan range image.	68
24	Left: A Return Light Intensity image of six calibration spheres. Right: The Visible Light image taken by the camera for which a calibration is sought.	68
25	The object association phase of the calibration process allows the user to manually identify each calibration object in each image and associate a neighborhood with each data set over which the best fit sphere and image centroid is determined.	69
26	Alignment of LIDAR and Camera using Calibration spheres and the Vis-Nav approach. The blue asterisks indicate the location of the $2D$ characteristic points, the red circles display the projected locations of the $3D$ -points through the collinearity equations which result from the calibration.	70
27	Alignment of LIDAR and Camera using Calibration spheres and the Vis-Nav approach. The blue asterisks indicate the location of the $2D$ characteristic points, the red circles display the projected locations of the $3D$ -points through the collinearity equations which result from the calibration.	72
28	Left: A range image of a model asteroid with zero pixels representing bad range measurements. Right-Top: A surface model showing the result of Gaussian convolution. Right-Bottom: The same surface textured by Return Light Intensity values achieved using sampled Gaussian Convolutions.	75
29	A uniformly sampled 1D signal smoothed by the finite difference approximation to the Laplacian	79
30	A non-uniformly sampled 1D signal smoothed by the finite difference approximation to the Laplacian	80
31	A non-uniformly sampled 1D signal smoothed by the distance corrected finite difference approximation to the Laplacian	81

FIGURE	Page
32	Asteroid image after 1, 5, and 10 applications of Laplacian smoothing. 84
33	Noisy Non-uniform 2D sampling of an asteroid image simulating the observation of the image surface by a LIDAR. 85
34	The sum of the per pixel error between the consecutive Laplacian smoothings of a uniformly sampled signal and the Laplacian and Graph Laplacian smoothing of the non-uniformly re-sampled signal. The graph Laplacian achieves lower errors over all blurs due to correct nearest neighbor weighting. 86
35	Equal angle sampling of a fronto-parallel plane by a two axis scanning LIDAR. The central point, whose measurement direction is orthogonal to the plane observes a symmetric distribution in its neighbors while every other point for the same operator observes an asymmetric distribution. 87
36	Left: Graphical representation of the textured LIDAR measurement array. Lines indicate assumed admissible paths along the surface. Right: A 7x7 graph Laplacian Operator. 89
37	Measurement array values of range, azimuth angle, elevation angle, return light, and visible light textures. 92
38	An example of range based scaling of operator scale. 95
39	Successive surface evolution by geodesic Gaussian convolution on the HST model. Top: The back plate of the HST model textured by return light intensity after 0,2, and 5 convolutions. Bottom: The side view of the surface after 0,2,and 5 convolutions. . . 97
40	Image representations of the diffusion displacements computed by the difference of geodesic Gaussians. Top: Mean Curvature, Return Light Intensity, and Visible Light for 2 convolutions. Bottom: The same for 5 convolutions. 98
41	A persistent geometric feature represented as a surface bump measured by a LIDAR system. 99

FIGURE		Page
42	An example of multi-field multi-scale feature extraction on the HST model. Left: The range field with curvature based features. Middle: The return light intensity field with texture features. Right: The visible light field associated with camera texture. The radius of the identified interest points indicates 2D feature scale.	101
43	The local sampling geometry associated with 5 steps Δ_α and ± 2 steps Δ_β	104
44	Spin Image population for a uniform distribution of points on a planar surface.	106
45	The 2D and 3D Multi-Field Diffusion Flow Signature Geometry. . .	109
46	Top: Diffusion Flow Signature Coordinates for an arbitrary feature with three fields. Bottom: The corresponding 2D arrays comprising the Multi-Field Diffusion Flow Signature.	111
47	Surface evolution encoded in a Diffusion Flow Signatures. The signature coordinates describe the euclidian distance between a feature point center and neighboring measurements along with the normalized magnitude of the diffusion flow vectors.	112
48	35 Matched features from approximately 100 extracted features in two scenes displaced by a 4 degree rotation of the HST spacecraft model.	116
49	N= 24 remaining feature points following OLTAE Residual Post-Filtering	116
50	Example feature matching for descriptors based on two different size localities. The smaller neighborhood feature does not have sufficient data to discriminate between the candidate locations for its match.	118

FIGURE

Page

51	Point based features extracted from a rotationally symmetric structure. High potential for ambiguity is demonstrated by comparing two candidate sections of the arc representing the locus of points representing curvature maxima and maximum texture gradients. It is virtually impossible to tell where the extracted local features were taken from.	118
52	Feature Fitness criteria based on OLTAE residuals. Top: A good match with a low OLTAE residual score. Bottom: A mismatch with a high residual score, scores above a threshold cause feature matches to be removed.	123
53	Top: Matched Features used to estimate the alignment of figure 48a. Bottom: Feature matches reduced by OLTAE residual post filtering used to estimate the alignment of figure 48b.	125
54	Left: Resulting OLTAE surface alignment from unassisted feature matching. Right: Resulting OLTAE alignment using residual based post filtering.	125
55	Covariance for Gibb's vector and translation vector components estimated by the OLTAE algorithm for 28 consecutive relative motion estimates.	126
56	Iterative Closest Point correction values for the Gibb's vector and translation vector components with 3σ covariance bounds for 28 consecutive relative motion estimates.	127
57	The MFMSTracker GUI environment.	129
58	The MFMSTracker Calibration Routine.	130
59	The MFMSCalibrate calibration process.	130
60	Resulting calibration performed manually for the HST model.	131
61	The Visualization Toolbar is used to change the view to display different field, and scale space level information for the Gaussian Scale Space.	133

FIGURE	Page
62	The Visualization Toolbar is used to display information in the Difference of Gaussian Scale Space. 134
63	Detected features displayed over the current view in the MFM-Tracker Software 136
64	Nonlinear transformation in Return Light Intensity for powers of 0.5 and 2 respectively. 137
65	Transformations causing artificial re-sampling associated with scaling and rotating the data. 137
66	Process Flow for the MergeModels function. Feature matching is iterated with motion estimation to identify reliable features to track and evaluate motion estimation performance. 139
67	An image texture is associated with a constant range surface sampled in equal angle segments. 143
68	Original texture used to color a constant range surface sampled in equal angle segments. 145
69	Scale Space Feature Extraction on the Graffiti image texture field for the Scale Invariant Feature Transform and Diffusion Flow techniques. 146
70	The original Graffiti and Asteroid textures are corrupted and transformed to evaluate feature descriptor matching performance. 146
71	The image noise is varied from zero to 10 percent causing a consistent decrease in the number of matches. 148
72	Rotational and Translational accuracy vs. image noise for the MFMS and SIFT algorithms. 148
73	Residual RMS and Variance vs. image noise for the MFMS and SIFT algorithms. 149
74	Number of matches reported for the MFMS and SIFT algorithms vs. lighting variations. 150

FIGURE		Page
75	Rotational and translational accuracies for the MFMS and SIFT algorithms vs. lighting variations.	151
76	Residual RMS and Variance values for the MFMS and SIFT algorithms vs. lighting variations.	151
77	Number of matches reported for the MFMS and SIFT algorithms vs. rotation.	152
78	Rotational and translational accuracies for the MFMS and SIFT algorithms vs. rotation.	153
79	Residual RMS and Variance values for the MFMS and SIFT algorithms vs. rotation.	153
80	Scale Space Feature Extraction on the Asteroid image texture field for the Scale Invariant Feature Transform and Diffusion Flow techniques. Notice that for the SIFT technique, many features are extracted in locations which do not reside on the measured/ object.	155
81	The image noise is varied from zero to 8 percent causing a consistent decrease in the number of matches.	156
82	Rotational and Translational accuracy vs. image noise for the MFMS and SIFT algorithms.	156
83	Residual RMS and Variance vs. image noise for the MFMS and SIFT algorithms.	157
84	Number of matches reported for the MFMS and SIFT algorithms vs. lighting variations.	157
85	Rotational and translational accuracies for the MFMS and SIFT algorithms vs. lighting variations.	158
86	Residual RMS and Variance values for the MFMS and SIFT algorithms vs. lighting variations.	158
87	Number of matches reported for the MFMS and SIFT algorithms vs. rotation.	159

FIGURE	Page
88	Rotational and translational accuracies for the MFMS and SIFT algorithms vs. rotation. 159
89	Residual RMS and Variance values for the MFMS and SIFT algorithms vs. rotation. 160
90	Range, Return Light Intensity, and Visible Light Fields for the HST model. 161
91	Extracted features for each of the 3 Fields for the HST model. 161
92	The noise in each texture field is varied from zero to 8 percent causing a consistent decrease in the number of matches. 162
93	Rotational and Translational accuracy vs. image noise for the MFMS and SIFT algorithms. 163
94	Residual RMS and Variance vs. image noise for the MFMS and SIFT algorithms. 163
95	Number of matches reported for the MFMS and SIFT algorithms vs. lighting variations. 164
96	Rotational and translational accuracies for the MFMS and SIFT algorithms vs. lighting variations. 164
97	Residual RMS and Variance values for the MFMS and SIFT algorithms vs. lighting variations. 165
98	Number of matches reported for the MFMS and SIFT algorithms vs. rotation. 165
99	Rotational and translational accuracies for the MFMS and SIFT algorithms vs. rotation. 166
100	Residual RMS and Variance values for the MFMS and SIFT algorithms vs. rotation. 166
101	Experimental setup for the HST partial circumnavigation experiment. The spacecraft is rotated in precise 2 deg increments in front of a Camera and LIDAR which make simultaneous observations. 168

FIGURE	Page
102	Top: Range field information for scans 1-7 in the partial circumnavigation experiment. Middle: Return Light Intensity field information. Bottom: Visible Light field information obtained from the camera. 169
103	Top: Range field information for scans 8-14 in the partial circumnavigation experiment. Middle: Return Light Intensity field information. Bottom: Visible Light field information obtained from the camera. 169
104	Top: Range field information for scans 15-21 in the partial circumnavigation experiment. Middle: Return Light Intensity field information. Bottom: Visible Light field information obtained from the camera. 169
105	Top: Range field information for scans 22-28 in the partial circumnavigation experiment. Middle: Return Light Intensity field information. Bottom: Visible Light field information obtained from the camera. 170
106	RMS Error for OLTAE residuals for SIFT matches used to obtain motion estimates. 170
107	Variance of OLTAE residual values of the SIFT matches used to estimate the relative motion. 171
108	The resulting registration of 28 textured scans using matched SIFT features. 172
109	RMS Error for OLTAE residuals for MFMS matches used to obtain motion estimates. 173
110	Variance of OLTAE residual values of the MFMS matches used to estimate the relative motion. 173
111	The resulting registration of 28 textured scans using matched MFMS features. 174
112	RMS Error for OLTAE residuals for the post filtered MFMS matches used to obtain motion estimates. 175

FIGURE	Page
113	Variance of OLTAE residual values of the post filtered matches used to estimate the relative motion. 175
114	The resulting registration of 28 textured scans using post fil- tered MFMS matches. 176
115	RMS Error for OLTAE residuals for the post filtered MFMS matches used to obtain motion estimates. 176
116	Variance of OLTAE residual values of the post filtered matches used to estimate the relative motion. 177
117	The resulting registration of 27 textured scans using post fil- tered MFMS matches. 177
118	The final model resulting from autonomous feature tracking using MFMS features and OLTAE residual post filtering. 182

LIST OF TABLES

TABLE		Page
1	Arbitrary image pixel and its eight nearest neighbors.	19
2	Classification of surface types based on the Mean and Gaussian Curvature Values.	47

CHAPTER I

INTRODUCTION

This dissertation develops a methodology for the generalized tracking of $3D$ features in spacecraft proximity operations. The developments are of a basic nature, however, and contributions to other fields such as computational vision and related applications are anticipated to flow from this work. While much work has been done in developing relative navigation solutions in the setting of autonomous spacecraft proximity operations, some aspects of this work also accommodate human input, active beacons, or communication between proximal bodies. The general problem of identifying and tracking “resident” features in an imaged scene remains an open research problem. Features in this dissertation are unique geometric or textural fingerprints whose appearance and especially, its computed descriptor is ideally invariant to a class of transformations associated with relative motions. Effective tracking requires extracted features to match accurately over large viewpoint variations encountered in practice, even in the presence of occlusions and discretization noise artifacts. Furthermore, virtually all of the common methods identify and characterize features in “image space,” rather than in the $3D$ “object space”. The advent of high speed, high spatial density, and accurate range sensors offers an important opportunity to extend $2D$ methods to $3D$ and simultaneously fuse passive vision with active ranging because the image space texture can be immediately associated with the measured co-registered $3D$ geometry.

Several techniques have been developed to approach the problem of autonomous proximity operations each with varying degrees of success. Perhaps most notably

This dissertation follows the style of *Journal of Guidance, Control, and Dynamics*.

DARPA's Orbital Express used a suite of algorithms and sensors to successfully perform a series of supervised, semi-autonomous proximity operations. The Autonomous Rendezvous and Capture Sensor System (ARCSS [1]) included both vision based techniques as well as active range measurements through the use of a laser ranging device. The vision aided proximity operations performed during the on orbit experiment used a template matching approach where a library of spacecraft image templates were used as match filters and compared to observations. The number of template comparisons was reduced by using the information provided by the laser ranger. While this on-orbit experiment was successful, it also served to establish that autonomous proximity operations research needs to address improved precision, bandwidth, efficiency and robustness.

Recent advances in scanning LIDAR technology have enabled high rate range sensing which provides High Definition (HD) range images refreshed at speeds comparable to the frame rates of existing cameras. While prior methods treated passive imaging and active range sensing as separate systems, this dissertation considers the real-time fusion of high rate range scanning and photogrammetry. The methodology adopted in this dissertation has been inspired by pioneering works in 2D and 3D data registration, the Scale Invariant Feature Transform [2,3] its recent extension to range images [4,5], Spin-Images [6,7], and their recent variations [8,9]. In the image and range data fusion case, scale is not ambiguous, it is explicitly measured and therefore can be exploited to characterize extracted features more accurately and uniquely. Furthermore, scale space extrema are evaluated over a prescribed set of length scales and those extrema which persist over multiple scales are defined over their observable range. The relation between feature scale and sampling is considered and used on the fly as part of the feature extraction and descriptor generation processes. An integrated analysis of texture and geometry is developed in which geometric and textural

features are localized at extrema in diffusion flow velocity, which is equivalent to surface curvature for geometric features and corners for texture features. A confidence measure which indicates the uniqueness and localization accuracy of the feature is also developed which takes into account the strength and persistence of the geometric and textural information. Finally, this framework is applied to real measurements during proximity operations experiments performed at the Land, Air, and Space, Robotics (LASR) Laboratory at Texas A&M University.

A. Aim and Objectives

Recent advances in range scanning technology have enabled High Definition range imaging, refreshed at video rates. A range image, is a $2D$ array whose elements contain the distance to the nearest object space point along an associated line of sight direction. Some “flying spot” LIDAR systems will also report the energy of the measured return light or “Return Light Intensity”. This scalar value is also associated with array elements. While flash LIDAR has been available for several years, it has lacked the range depth of field variation and lateral spatial resolution afforded by scanning LIDAR. The natural energy per unit area on the flash LIDAR focal plane array is proportional to $(range)^{-4}$, whereas for scanning LIDAR the return energy is proportional to $(range)^{-2}$. Therefore, scanning LIDAR is suitable for lower power imaging for long range standoff distances from the imaged surface. As in computer vision, it is now possible to generate local models for each “frame” of range data and register overlapping frames on the fly. In computer vision this process is known as structure from motion and requires relative camera-to-scene motion to be estimated as well as the world scene geometry. In applying traditional monocular structure from motion algorithms, several image frames must be acquired before an adequate

displacement has occurred to make triangulation numerically well conditioned, effectively generating the needed baseline for triangulation using the motion of the sensor itself. The density of the estimated scene geometry is dependent on the observable texture. For bland scenes or objects, stereo systems typically require augmentation such as a projected texture from a structured light source. Vision based methods alone are also very sensitive to photometric variations in the scene making their use difficult in the space environment. Ambient lighting in space may run the gamut from very few photons to complete saturation of the image and vary with time due to a variety of factors. Expensive autonomous space systems cannot afford the poor reliability of traditional vision systems as a primary means of relative navigation. Range images on the other hand depend on active illumination and measurement of diffusely reflected energy, to estimate the round-trip time of flight for a scanning pulsed laser over a field of view. They are known to be more robust to photometric variations and directly obtain scene geometry with no need for triangulation. Direct measurement of scene geometry at frame rate allows for the shape and relative pose of proximal bodies to be more precisely determined as well as real and fine co-registration of imaged texture and object geometry. Image features tracked using correlation in overlapping images with large relative displacements may fail in the presence of perspective distortion due to local surface curvature and sampling. These affects can now be reduced, using the methods of this dissertation, allowing for rigorously linear estimation of relative motion while fusing measurements made from various sensors-to-scene poses. Also of relevance, lens-based imaging, for a given sensor, can produce properly focused images only for a finite set of ranges between the sensor and the scene. Laser scanning based LIDAR can operate over many orders of magnitude larger range variations and therefore affords a more flexible and robust space sensor system. Therefore measured geometry can only be co-registered with the corresponding texture when well-focused

imagery exists. These facts have motivated this dissertation, and the results represent a substantial set of contributions.

1. Scientific Questions

The following scientific questions are addressed in this dissertation:

- By what data driven process can $3D$ object-space features be best extracted from textured range imagery and tracked through general relative motion?

The application of the diffusion equation to image data performed in the Scale Invariant Feature Transform can be rigorously extended to $3D$ range measurements and associated textures. The resulting multi-field multi-scale interest points are analogs of corner-like features in image processing. Spin Images are typically used for determining surface point feature correspondences. A new descriptor inspired by this perspective, the Multi-Field Diffusion Flow Signature, is used to combine geometric and textural information, and proves useful in regions that are texturally unique but geometrically non-unique. It is the ability to distinguish features that are texturally non-unique but geometrically unique as well as features that are geometrically non-unique but texturally unique which allows for robust feature point detection and tracking in the presence of repeating structures, symmetry, textured simple surfaces, and bland curved surfaces.

- How can local geometry and texture be combined to allow point based descriptors to enable rigorously linear six degree of freedom relative motion estimation?

The addition of curvature information and more generally, geometric granularity measurements in both the localization of features and the population of their descriptors, allows for effective $3D$ localization of features on geometrically sig-

nificant structures as well as in texturally significant locations. Understanding the geometry of textured features allows us to discriminate between texture features that represent resident object space texture, and photometric textures due to shadows and reflections. Ensuring feature points are static with respect to object geometry enables effective relative motion estimation by the use of unique rotational coordinates and the transformations implicit in the Optimal Linear Translation and Attitude Estimator (OLTAE), [10]. We show that OLTAE eliminates the usual nonlinear least square iteration (based on local Taylor series expansions) in favor of a rigorously linear least square fusion of overlapping point clouds. Since the residual errors are a consequence of feature localization errors only (no linearization approximation), statistically methods can be involved to reject spurious features and to characterize rigorously local errors.

- Does the addition of direct range measurement allow for significant increases in precision or robustness?

Since feature scale is no longer ambiguous, it can be directly used to characterize features. Can a scale space combining geometry with multiple textures provide a means to more accurately localize unique variations in object space resulting in more effective feature tracking in the space environment? Does the simultaneous consideration of geometric characteristics and texture characteristics give rise to more unique features?

- Can the combined scale space analysis of 3D object space manifolds and associated texture along with the application of Multi-Field Diffusion Flow Signatures result in robust motion estimation for real data in proximity operations?

Experimental evaluation and validation is performed to understand the perfor-

mance of the proposed architecture. An interactive software is developed in Matlab to acquire, calibrate, and assess active and passive electro-optical measurements using scale space analysis. This software is used to evaluate feature detection, localization, and motion estimation for simple relative motions subject to noise, lighting variation, as well as in and out of plane rotations.

It is noted that the software developed is for research purposes, to efficiently answer the questions posed above. It follows that subsequent efforts are required to realize efficient codes for real-time implementation. There are no claims that the novel algorithms are implemented in a way to achieve computational optimality.

2. Motivation

Space based assets will benefit greatly from precise, robust, and routine ability to execute autonomous proximity operations. Many space operation tasks such as spacecraft refueling, inspection, and repair will one day rely on autonomous sensing and relative navigation. In ideal cases a model is available for the target body involved in a proximity operation and communication between both bodies is possible. However in other applications such as debris capture and removal, little may be known about a proximal body, and communication is not possible. A spacecraft must be capable of discerning from its measurements how to safely navigate relative to a poorly known proximal body. Recent advances in Light Detection and Ranging systems have enabled a next generation suite of sensors to replace systems like the ARCSS system [1] where the measurements are co-registered by calibration to permit real-time fusion of texture and geometry. Having in hand co-registered range and image sensors, both measuring millions of points per second over a large (say 30 deg) field of view, enables a new generation of algorithms for SLAM and vastly improves

accuracy, robustness, and versatility. This enabling sensor technology in fact allows us to develop methods which can dramatically reduce the computational burden of 6 DOF relative navigation solutions using 3D geometric and resident textural features, while simultaneously achieving improved robustness. Conventional methods for extracting from image data the “most trackable features” vary, but the most promising detect the most uniquely textured image regions and localize their position in image (x, y) coordinates and their image-space scale σ . Once the features are localized, a feature descriptor encodes the gradient information in a locality sampled consistently in image space. While all of these methods are designed to be as “invariant as possible” in some sense, in practice they all exhibit frequent failures due to:

- Lack of sufficiently dense and unique texture, resulting in too few features.
- Large out of plane rotation, resulting in incorrectly sampled descriptor vectors.
- 3D effects due to local surface curvature, resulting in poorly characterized features whose descriptors change with relative pose.
- Occlusions, resulting in multi-view feature variations, poor feature matching, and localization errors.
- Non-linear or photometric variations, resulting in changes of the texture gradients and poor feature characterization and matching performance.
- Lack of uniqueness between near-identical scene features, causing mismatches between feature sets.

The leading approach to the registration of 3D data [6, 7] requires the data to be triangulated and the nearest neighbors of each point to be known. The data is also regularized and re-triangulated so that the inter-point distance is consistent in

object space. Using the Spin Image concept, the local surface normal of each point is estimated and points in the neighborhood of a point of interest are encoded by reporting their perpendicular distance from the local tangent plane and their in-plane radius. Until recently [11], registration by comparing spin images was performed by means of comparing randomly selected points, or exhaustive comparisons which can take $O(N^2)$ computation time. Spin Images also have several tuning parameters which are left to be determined by the user. A feature detection method is desired which (i) localizes features based on uniqueness in multi-field data sets, (ii) encodes the object-space feature scale(s), (iii) is robust to photometric variations, (iv) can find trackable locations in the absence of texture, and (v) consistently samples local geometry in the construction of its descriptor. Establishing such a feature detection approach is the main goal of this dissertation. It is recognized that such a descriptor has broad utility and likely will prove to be a basic contribution to computer vision. It is noted that typical scenes may yield hundreds of detected features, whereas if the goal is proximity navigation, as few as three features in two or more overlapping measurement sets are required to determine relative translation and rotation. The residuals for the heavily redundant, rigorously linear (via the OLTAE algorithm) least squares registration process provides a strong statistical basis for accepting the features with smallest residuals and rejecting spurious or poorly localized features with large residuals.

B. Background

1. A Brief History of Spacecraft Proximity Operations

Spacecraft rendezvous and docking has a rich history and was a necessary capability which enabled the more ambitious space missions including the accomplishments

of the Gemini and Apollo programs. In fact, the Apollo program was built around rendezvous and docking operations in lunar orbit. Approximately 57 shuttle missions have called for proximity operations to be performed [12], however, most of these were performed with human-in-the-loop control. There are many future applications where autonomous proximity operations will be required and many reasons why autonomous operations will be preferred over manual, human in the loop processes. An explosion of technology and successful flight demonstrations has occurred in the last 5 years. In 2005 the Air Force Research Laboratory's XSS-11 spacecraft represented the first successful series of autonomous proximity operations. In that same year the ill-fated DART spacecraft also launched with similar ambitions. Whereas XSS-11 performed successful circumnavigations of the upper stage of its launch vehicle, DART's unsuccessful mission was to autonomously rendezvous with a dead communications satellite and demonstrate its Advanced Video Guidance Sensor. In 2007 the DARPA Orbital Express mission demonstrated a series of successful proximity operations including rendezvous, approach, station keeping, docking, and fuel transfer. At the same time the Naval Research Laboratories SUMO/FREND spacecraft project was developing technologies and ground-based experiments for autonomous grappling of spacecraft. Among these technologies was TriDAR, a scanning lidar system and 3DLASSO, machine vision software for relative navigation and tracking. Recent years have also seen the development and flight of so called Autonomous Rendezvous and Proximity Operation (ARPO) Systems including the ARCTUS and the Jules Verne which has successfully performed supervised autonomous proximity operations with the International Space Station. Even ambitious student satellite programs are beginning to explore the problem including DRAGONSAT and AGGIESAT, a university nano-SAT program in which students at Texas A&M University and the University of Texas at Austin are involved in flight testing necessary component technologies

with the end goal of performing an autonomous docking between the two nano-SATs. While a variety of proximity operations with various levels of autonomy have been conducted, there is ample evidence that the reliability and versatility of available methods are not adequate to meet anticipated needs. Of immediate relevance, the Land, Air, and Space Robotics (LASR) laboratory at Texas A&M University has developed a *6DOF* space proximity operations research laboratory where advanced vision sensors, algorithms, and their utility in enhancement of closed loop control of proximity operations are experimentally evaluated. This laboratory enabled the experiments reported herein.

2. Developments in *2D* and *3D* Data Processing

While the interest in and need for autonomous spacecraft proximity operations has been developing, the necessary mathematical tools have seen rapid development in the fields of image processing, *3D* data processing, differential geometry, and computer vision. The mathematical framework utilizing concepts from differential geometry, as mapped onto machine vision, saw its major developments due to do Carmo in 1976 [13]. Since those early developments, several examples of applying differential geometry to develop machine interpretation of *3D* manifolds and use of the signs of the Gaussian and Mean Curvatures to classify surface data emerged in the 1980's [14–16]. During the same decade, developments in corner detection and feature tracking were enabling early vision and motion estimation techniques [17, 18] and the mathematical developments of scale space analysis also saw their beginnings [19, 20]. In the 1990's, corner features were generalized to interest points detected by so called blob detectors where information in their localities was compared by means of correlation techniques and the comparison of feature vectors [21]. Also during this time frame we saw the emergence of descriptors theoretically invariant to translation and

rotation [22], similarity transformations [2, 3], and affine transformations in the early 2000’s [23–25]. Since the late 1990’s and early 2000’s, the landmark contributions of 2D and 3D image processing have seen widespread use in computer vision, robotics, visual servoing, and simultaneous localization and mapping. A discussion of scale space analysis, the Scale Invariant Feature Transform (SIFT) [2, 3], Spin Images [6, 7], and recent related developments that are especially relevant to this dissertation is provided in chapter 2.

While significant and successful applications are widespread, it is fair to say that there is a “religious” aspect to the use of existing methodologies because all require significant tuning and heuristic insight to achieve good results on difficult scenes. The tuning and heuristic issues make “fair comparisons” difficult, and also suggests that there is significant room for further improvements in methodology.

C. Overview and Approach

1. Hypothesis

This dissertation develops a new framework for generalized feature detection and tracking in opto-LIDAR fusion systems and argues that model acquisition, processing, registration, and relative navigation can be performed accurately using the developed framework. By extending and combining the methods of 2D and 3D data processing, and by fusing the geometric measurements of Light Detection and Ranging with the texture information acquired by a co-registered camera, features may be localized and tracked in ways not possible to the sensors in isolation. Finally by exploiting the directly measured geometric information, the descriptors of isolated features may be populated by means of a process which samples the object space, resulting in significantly improved performance.

2. Method

Figure 1 illustrates the general framework for feature detection and tracking developed in this dissertation. First, a relative calibration is done to determine the sensor extrinsic parameters, i.e. the relative rotation and translation between the measurement frame of the LIDAR, and the image plane coordinate system of the camera. After this alignment, the texture information from the camera is co-registered with the 3D measurements of the laser scanner. Each individual LIDAR measurement, if visible to the point of view of the camera, is textured with a grayscale value indicating the intensity of visible light in the image. A second texture value provided by many LIDAR sensors is the return light intensity, a grayscale value indicating the energy of the returning photons from the pulsed LIDAR measurement. The intensity variation over the object is a measure of the wavelength specific albedo of the surface. The diffusion equation is applied to the co-registered geometry and intensity data resulting in a multi-resolution analysis of 3D geometry, and 2D texture. Feature points are localized in regions containing significant diffusion flow. This occurs when the measured surface is locally curved in 3D or when the observed texture exhibits large gradient variations. Once these features are localized, a characteristic dimension is derived and assigned to the feature which depends on the range to the feature center and the angular extent of the feature. Using this characteristic dimension to define a sampling neighborhood, the local surface geometry and texture information is encoded in an object space feature descriptor. Finally features are compared and matched in subsequent scans and the best matches are used to estimate the 6 DOF relative sensor navigation solution using a rigorously linear algorithm. In a given set of matched features, there are typically more correspondences than the minimum required for the Optimal Linear Translation and Attitude Estimator (OLTAE) least squares process.

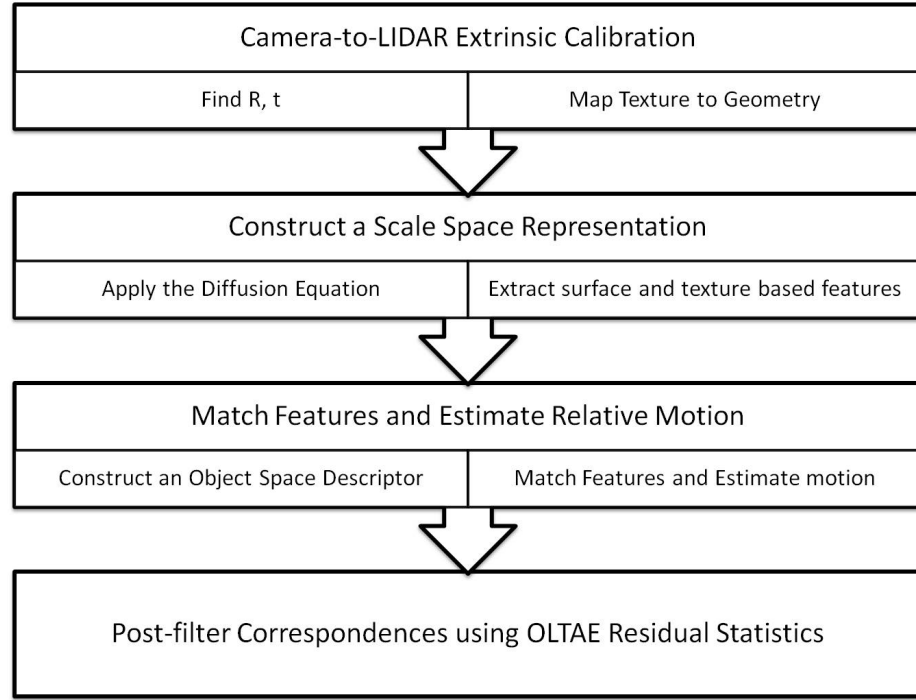


Fig. 1: General feature detection and tracking framework for opto-LIDAR fusion systems

The residuals determined by OLTAE for each correspondence are used to filter the set of matches and remove any remaining poorly localized or mismatched features.

3. Contributions

This dissertation makes the following important contributions to the field of aerospace engineering:

- A meshless approach to combined Scale Space analysis of LIDAR and camera measurements.
- A combined multi-field, multi-scale (MFMS) feature detection process which defines its descriptor in the object space and is robust to noise and lighting variations.

- Statistical characterization of the resulting feature's OLTAE residuals to select the most consistent and accurately localized set of MFMS interest points.
- The connection between surface evolution by mean curvature flow and scale space analysis of images embedded on $2D$ manifolds is demonstrated and applied to raw $3D$ data.
- A software pipeline which allows interactive scale space analysis of range data and camera imagery as well as enables the visualization of the extracted features and constructed models.
- Experimental results on real data demonstrate these MFMS features capability to enable simple examples of spacecraft proximity operations.

4. Overview of Dissertation

This dissertation is organized as follows:

- Chapter 2 provides a review of the literature relating to $2D$ and $3D$ feature extraction.
- Chapter 3 explains the measurement models for each sensor along with an approach to relative calibration and data fusion.
- Chapter 4 describes the process of feature localization in multi-field data.
- Chapter 5 introduces our novel process for generating point descriptors with known characteristic scale along with a post-filtering process based on the residuals generated by the Optimal Linear Translation and Attitude Estimator (OLTAE), a novel, rigorously linear motion estimator .

- The computer software pipeline is discussed in chapter 6 along with an example of registration of the Hubble Space Telescope (HST) model.
- Chapter 7 discusses the experimental validation of the new concepts and computational framework.
- The dissertation is concluded and future work is discussed in chapter 8.

CHAPTER II

BACKGROUND AND LITERATURE REVIEW

There is a significant literature regarding the processing of $2D$ and $3D$ data and the extraction of trackable features. This chapter discusses the most relevant historical developments that enabled, informed, and motivated this dissertation. The general analysis of $3D$ geometry (provided by applying concepts from differential geometry and its applications to surface scans) is discussed along with methods of $3D$ point feature localization and description. Important developments in $2D$ feature tracking are discussed in the context of camera to scene localization and estimation of scene geometry. The Scale Invariant Feature Transform is discussed in detail along with its recent application to range images. The strengths and weaknesses of this technique with regards to $3D$ localization and motion tracking are explained. The important technique of $3D$ point feature description by Spin Images is also introduced along with opportunities for the integration of $2D$ and $3D$ interest point detection and description. The chapter concludes by suggesting a path for improving methods documented in the current literature by combining the analysis of texture and geometry, as well as regularizing the way point based descriptors are populated with respect to the measured $3D$ geometry.

A. Developments in Image Feature Extraction

1. Basic Image Structures and Gradients

A digital image $I(x, y)$, is a matrix representation of pixel response data measured by a Charged Coupled Device (CCD) or Complementary Metal Oxide Semiconductor

(CMOS) which typically reports the digitized photo detection measure of energy on the focal plane, resulting from the projection of light through an optical system. Each matrix element known as a *pixel* reports a numerical analog to digital (A/D) value indicating the collected photons on the corresponding sensor element over the exposure time. While details of the measurement model are reserved for chapter 3, we mention here that calibration of a camera generally generates a function which applies a nonlinear correction to the original image. This correction maps the data such that it corresponds to a pinhole camera model with a co-linearity condition where a point in object-space, the principle point of the camera, and the calibration-corrected image point lie on the same straight line. In the process of projecting the outside world geometry to the image plane, information on the third dimension, i.e. depth from the camera is lost. This means that a single image can not discern the range to 3D points without additional information. Early image processing techniques focused on the isolation of precise mathematical primitives such as points or lines. Localizing these primitives relies heavily on the calculation of image intensity gradients.

$$\nabla I(x, y) \equiv \begin{bmatrix} I_x \\ I_y \end{bmatrix} = \begin{bmatrix} \frac{\partial I(x, y)}{\partial x} \\ \frac{\partial I(x, y)}{\partial y} \end{bmatrix} \quad (2.1)$$

Image gradients are 2D image-space vectors with a magnitude M and direction α which indicate the change in image intensity in the horizontal x and vertical y directions.

$$M(x, y) = \sqrt{I_x^2 + I_y^2} \quad (2.2)$$

$$\alpha(x, y) = \arctan \left[\frac{I_y}{I_x} \right] \quad (2.3)$$

Table 1: Arbitrary image pixel and its eight nearest neighbors.

$I(x - 1, y - 1)$	$I(x, y - 1)$	$I(x + 1, y - 1)$
$I(x - 1, y)$	$I(x, y)$	$I(x + 1, y)$
$I(x - 1, y + 1)$	$I(x, y + 1)$	$I(x + 1, y + 1)$

The image partial derivatives in the above equations are usually approximated by finite differences over the neighborhood of a chosen pixel. Table 1 illustrates the immediate neighborhood of a pixel and its 8 neighbors. One method for approximating image partial derivatives is given by computing central differences at each pixel. An algorithm which implements the central difference derivative approximation in equations 2.4 and 2.5 at every pixel should be reasonably accurate at computing image partial derivatives of a smooth pixel response, but would not be very robust in the presence of noise. It is preferred to use operators which compute image quantities as a function of an area.

$$\left[\frac{\partial I(x, y)}{\partial x}\right] \simeq \frac{1}{2} [I(x + 1, y) - I(x - 1, y)] \quad (2.4)$$

$$\frac{\partial I(x, y)}{\partial y} \simeq \frac{1}{2} [I(x, y + 1) - I(x, y - 1)] \quad (2.5)$$

Often in image processing however image gradients are approximated by use of spatial operators which implement the above equations or similar operations over a neighborhood. Examples include the Sobel, Prewitt, and Roberts operators and all of them represent differently weighted finite difference estimates of the image gradients over the 3x3 neighborhood of an image pixel and its 8 nearest neighbors [26]. The key to choosing a derivative approximation operator is knowing the desired accuracy of the derivative approximation and the level of computational complexity allowed. The Sobel operators shown in figure 2 are an example of a computationally simple oper-

-1	0	1	-1	-2	-1
-2	0	2	0	0	0
-1	0	1	1	2	1
$\frac{\partial I(x,y)}{\partial x}$			$\frac{\partial I(x,y)}{\partial y}$		

Fig. 2: The Sobel X and Y derivative operators.

ator since all of its elements are integers and the operator is only 3×3 . The resulting gradient magnitudes do not however, represent the technically correct partial derivative values of the image intensity function but rather are a weighted area measure of the intensity change. For many applications, only the gradient magnitude pattern and gradient direction are of interest, not the exact magnitude. For cases where the gradient magnitude is more important other operators are used.

These finite difference approximations to the image x and y derivatives are sensitive to noise and are usually applied after an image has been smoothed by applying a low pass filter. The simplest way to apply smoothing to an image is to use a spatial filter, which replaces a pixel value by an average of its original value and its neighbors. For example, a 3×3 mean filter would take the average value of a pixel and its 8 nearest neighbors. Applying the filter to each pixel in an image is done by a process known as convolution. Convolution approximates the convolution integral by computing the sum:

$$I'(x, y) = \sum_{-i=-a}^a \sum_{j=-b}^b w(i, j) I(x - i, y - j) \quad (2.6)$$

It is perhaps more common that a pixel is replaced by a Gaussian weighted average of its neighborhood, in fact the circular Gaussian density function can be shown to be a good approximation to the first Bessel function of the first kind used to model the

passage of light through an optical system with a circular aperture. Romney [27] also shows that one can derive the Gaussian as the necessary representation of a point in $2D$ space from a set of fundamental principles:

- Linearity - the measurement is done in a linear fashion.
- Uniformity - there is no preference for location in the image, all locations are observed by the same aperture function.
- Orientation Invariance - uniform preference for orientation in the image, structures of any orientation should be equally observable.
- Scale Invariance - no preference for scale, structures of any observable scale should be measured in the same way.

If we look at the Fourier Transform of equation 2.6 we obtain:

$$I'(\omega_x, \omega_y) = I(\omega_x, \omega_y)G(\omega_x, \omega_y) \quad (2.7)$$

which is the typical array product in frequency domain filtering. Since the desired operator is required to be isotropic, it is only important to consider the length of the frequency vector $\|\omega\| = \sqrt{\omega_x^2 + \omega_y^2}$. The property of scale invariance requires $G(\sigma_1\omega)G(\sigma_2\omega) = G(\sigma_1\omega + \sigma_2\omega)$. The general solution is given by:

$$G(\sigma\omega) = e^{(\alpha\sigma\omega)^n} \quad (2.8)$$

The dimensions are independent and separable, and it is evident that the calculation of the magnitude of the frequency vector is a 2-norm, so we set n to 2, and for $\alpha^2 = -\frac{1}{2}$ since in the limit as $\|\omega\| \rightarrow \inf$ $G(\omega, \sigma) = 0$ we may obtain by the Inverse Fourier Transform:

$$G(\omega, \sigma) = e^{-\frac{1}{2}\sigma^2\omega^2} \quad (2.9)$$

Or, in the spatial domain:

$$G(x, y, \sigma) = -\frac{1}{2\pi\sigma^2}e^{-\frac{x^2+y^2}{2\sigma^2}} \quad (2.10)$$

Which is the definition of the Gaussian kernel. This kernel is also the so-called Green's Function for the linear, isotropic diffusion equation. To obtain stable estimates for image derivatives, it is common to use a Gaussian smoothing process combined with a derivative operator. Since both processes are linear, it is possible to combine the processes by convolving an image with a Gaussian Derivative operator, which can be computed by sampling the partial derivative expressions of the Gaussian function. Examples of Gaussian Derivative operators are shown in figure 3. The standard and perhaps most reliable method for obtaining image gradients in computer vision uses the application of Gaussian Derivative operators, in fact, many useful detectors based on image geometry are derived from these operators. Following the early developments of edge detection in image data, an edge being defined as a set of pixels belonging to a boundary over which the image gradient magnitude exceeds a threshold, the special and unique image locations representing large gradients in all directions were identified as corners in image data [28].

2. Corner Detection

One of the earliest corner detection routines is attributed to Morevec whose 1980 thesis regarding stereo vision techniques for crude object avoidance for an autonomous rover included an interest operator which defined a neighborhood about a pixel and compared the sum of squared differences between neighborhoods of neighboring pixels. Image locations displaying low self-similarity were candidates for corner features.

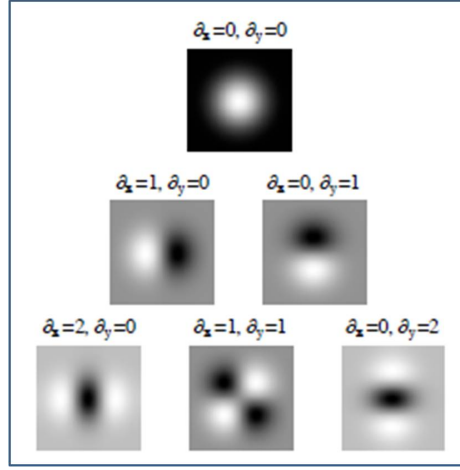


Fig. 3: The Gaussian Kernel and Partial Derivative Operators up to 2nd order.

A more elegant approach to corner detection was presented in 1988 by Harris and Stephens [18]. Instead of comparing windows over a predefined neighborhood about an image pixel, they define the structure tensor over a point neighborhood and introduce a Gaussian weighting. By evaluating the eigenvalues of the structure tensor they were able to classify edge responses from bland regions and corners. For all participating pixel members of a Gaussian weighted neighborhood, the weighted sum of squared differences is given by equation 2.11.

$$SSD(x, y) = \sum_i \sum_j w(i, j) (I(x + j, y + i) - I(i, j))^2 \quad (2.11)$$

Here (i, j) are the vertical and horizontal displacement indices of the pixel neighbors. The weighting function $w(i, j)$ is a gaussian weighting function chosen to match the size of the pixel neighborhood. The displaced pixel $I(x + j, y + i)$ intensity can be approximated by using the Taylor Series expansion as:

$$I(x + j, y + i) = I(i, j) + I_x(i, j)x + I_y(i, j)y \quad (2.12)$$

and by substitution of 2.12 into 2.11 we obtain:

$$SSD(x, y) = \sum_i \sum_j w(i, j) (I_x(i, j)x + I_y(i, j)y)^2 \quad (2.13)$$

which is a weighted quadratic function of the image partial derivatives and (x, y) displacements of the pixel neighbors. This can be represented more generally in vector matrix form by:

$$SSD(x, y) = \begin{bmatrix} x & y \end{bmatrix} A \begin{bmatrix} x & y \end{bmatrix}^T \quad (2.14)$$

where A is the structure tensor and is given by:

$$A = \sum_i \sum_j w(i, j) \begin{bmatrix} I_x^2 & I_x I_y \\ I_x I_y & I_y^2 \end{bmatrix} \quad (2.15)$$

The eigenvalues of the matrix A , α and β , provide the proportions of the principle curvatures of the local autocorrelation function. Simply put these developments provide a direct means of evaluating the local curvature of image data. Figure 4 shows the classification of image response as a function of the eigenvalues of A . The response function was developed to quickly compare the eigenvalues of the structure tensor and determine the degree to which the data was curved and therefore represented a corner. It is given by:

$$R = \det(A) - kTr^2(A) = \alpha\beta - k(\alpha + \beta) \quad (2.16)$$

Here, k is an empirically determined constant used to scale the relative size of the eigenvalue sum compared to the product of the eigenvalues. By applying a threshold to the response function of 2.16 instead of directly computing and comparing the eigenvalues, corner feature strength can be determined without having to compute the more costly square roots involved in the eigenvalue decomposition. This approach

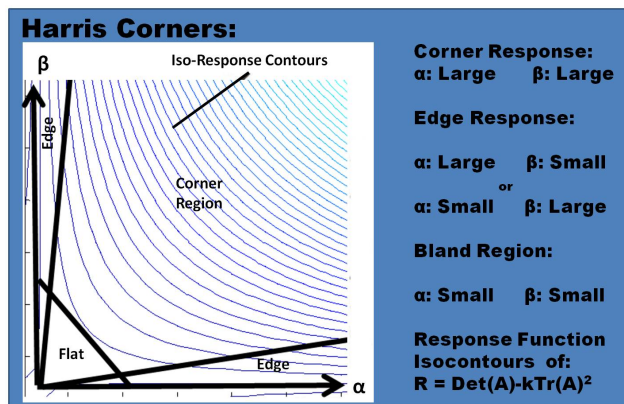


Fig. 4: The distribution of image structures as a function of the eigenvalues of the structure tensor.

works for images of any resolution, and has seen wide application. The next logical step was to extend this framework to multi-resolution image analysis.

3. Multi-Resolution Image Analysis: The Scale Space

Interest points may occur at all scales in image data. The selection of the parameters regarding the extent of a point neighborhood as well as the distance between samples over which finite differences are used to compute gradients directly affect the resulting feature locations extracted from an image. Varying these parameters, and sampling the image at multiple scales so that features can be tracked through different sampling frequencies are among the desired outcomes of applying scale space theory to image data. A substantial existing literature with well developed mathematical principles lies at the heart of scale space theory, only the main developments that motivate the literature on recent feature detectors will be discussed here. The interested reader is referred to work of Witkin [20] and Koenderink and Van Doorn [19]. Applications of Scale Space Theory in computer vision are also documented in the text by Lindeberg [29].

Observable physical phenomena exist at many length scales. This fact is readily

known as we use several different resolutions and units of measurement to describe what we see with our instruments. Celestial length scales are many orders of magnitude larger than objects we experience every day, while those objects are themselves orders of magnitude larger than atomic structures. Each measurement device can be thought of as having a pass band over these length scales through which features are observable. In image processing the key issue with regard to scale is the number of pixels required to capture the image-space “size” of feature, sub-features, or collections of features that emerge in the various feature detection approaches. The lower limit in spatial scale is bounded by the physical resolution of the measurement device while the upper limit is bounded by the physical aperture. “Size” has to be generalized and thought about differently when considering the fusion of image space measurements with range measurements from a LIDAR sensor and will be discussed later. Witkin developed scale space theory and applied it to 1D signals. He observed that the zero crossings of a function’s second derivative at each scale form a so-called nesting tree, and could be used to characterize the signal by developing a hierarchical relationship between features at different scales. A scale space is defined by embedding the original signal into a one parameter family of derived signals. This is done by convolving the original signal with an operator whose scale parameter can be varied to control the spatial participation of the original signal in the convolved signal. As it applies to images, this convolution operation is denoted by equation 2.17.

$$L(x, y, \sigma) = g(x, y, \sigma) * I(x, y) \quad (2.17)$$

Here the function $g(x, y, \sigma)$ represents the 2D Gaussian function and its standard deviation σ represents its characteristic scale.

$$g(x, y, \sigma) = \frac{1}{2\pi\sigma^2} e^{[\frac{(x-x_o)^2}{2\sigma^2} + \frac{(y-y_o)^2}{2\sigma^2}]} \quad (2.18)$$

The ordinate pair (x_o, y_o) could be interpreted as origin of the gaussian operator in image coordinates. This origin is swept over the domain of the image through the process of convolution. The resulting family of derived signals are successively smoothed representations of the original signal. The extrema of the image derivative with respect to the scale σ identify points of interest or features in scale space.

The selection of the Gaussian kernel was no accident. Koenderink [19] introduced the notion of causality, where for grayscale images, no spurious image structure should occur as the result of applying an operator. This means that every gray level of a coarse scale should find its source at a finer one. By applying this constraint along with the requirements that the operator be isotropic and homogeneous, that is, it should be spatially unbiased and consistent over sampling scales, he determined that the operator must satisfy the diffusion equation. These requirements are necessary to ensure that applications of the kernel do not cause spurious maximum and minimum that are artifacts causing false features and not extrema due to elements of the original signal. The linear isotropic diffusion equation is given by:

$$d_t L(x, y, t) = \frac{1}{2} \nabla^2 L = \frac{1}{2} (d_{xx} + d_{yy}) L \quad (2.19)$$

Where d_t represents the derivative operator with respect to time, and d_{xx}, d_{yy} are the second order partial derivative operators with respect to x , and y respectively. The “time” variable in this context is the scale parameter of the Gaussian operator, i.e. $t = \sigma^2$. So diffusion can be thought of as controlling the participation of the information from neighboring pixels into $L(x, y)$. The Gaussian kernel satisfies the diffusion equation making it the unique kernel for generating a scale space. The

standard techniques for generating a scale space increase the dimensionality of the problem of corner detection. Since Gaussian convolution simulates the localization process of sampling at various spatial scales, it is also simulating the effect of obtaining the same image with a lower resolution. In fact combining re-sampling at a scale consistent with the size of the Gaussian operator results in a very common data structure: the Gaussian Pyramid.

4. Image Pyramids

Image pyramids are widely used efficient data structures for multi-resolution image analysis. The construction of an image pyramid involves two processes which may be performed simultaneously. The first is the convolution of the image with an operator, usually a Gaussian or Laplacian. The second operation is re-sampling. The Gaussian pyramid was originally developed for machine vision and image compression. The space required to store an image pyramid is theoretically 4/3 of the size of the original image. The REDUCE function [30] is shown in equation 2.20. An example image pyramid is shown in figure 5.

$$g_k(i, j) = REDUCE(g_{k-1}) = \sum_{m=-2}^2 \sum_{n=-2}^2 w(m, n) g_{k-1}(2i + m, 2j + m) \quad (2.20)$$

The Gaussian Pyramid is an example of a low pass pyramid. A more interesting type of image pyramid is the band pass pyramid where higher resolution image information is encoded by its difference from a lower resolution image representation. In order to take the difference between different resolution levels of the Gaussian Pyramid, the EXPAND function is required:

$$g_{l,k} = EXPAND(g_{l,k-1}) = 4 \sum_{m=-2}^2 \sum_{n=-2}^2 w(m, n) g_{l,k-1}\left(\frac{i-m}{2}, \frac{j-n}{2}\right) \quad (2.21)$$

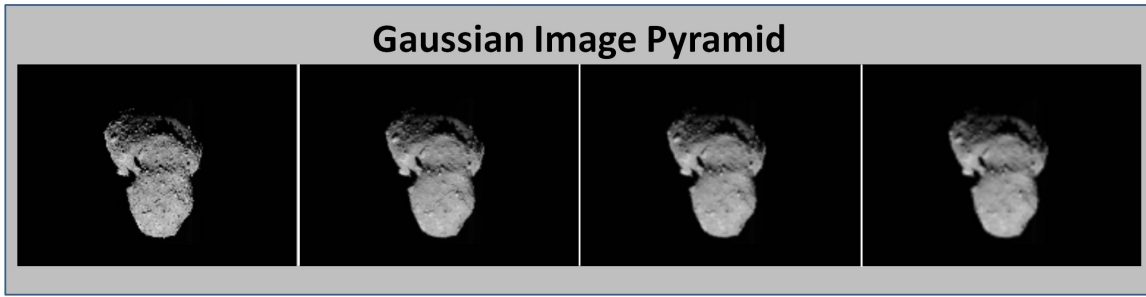


Fig. 5: Example Gaussian Pyramid: The image is repeatedly low pass filtered and re-sampled by a factor of two.

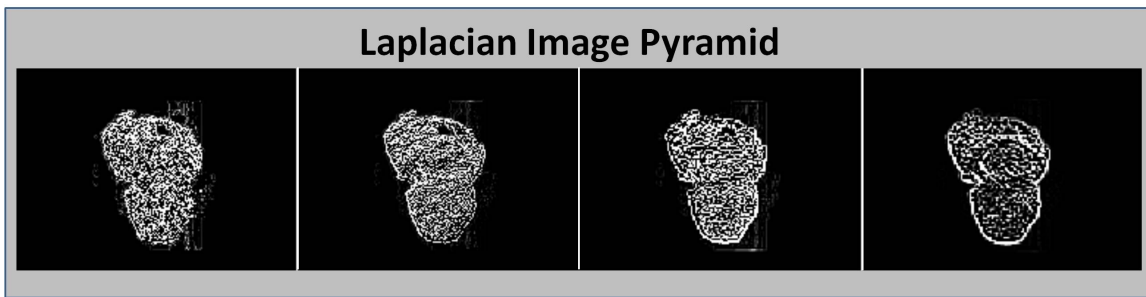


Fig. 6: Laplacian Image Pyramid built by computing differences between original and expanded levels of the Gaussian Pyramid. For visualization purposes the absolute value is shown for each level.

Here the arguments $\frac{i-m}{2}, \frac{j-n}{2}$ must be integers to be included in the summation. By taking the difference between an original level in the Gaussian Pyramid with an expansion of the next pyramid level, the error images L_k which make up the Laplacian pyramid may be generated. The corresponding Laplacian Pyramid to Figure, 5 is shown in Figure 6. To see that this is a band pass we mention here that this approach to building a Laplacian Pyramid is an approximation to convolving the image with a series of Laplacian of Gaussian filters which is the isotropic second derivative of the Gaussian functions. The discrete Laplacian of Gaussian operator is show in figure 7.

Localization of image features in an image pyramid must take into account the scale factor between the arbitrary level of the pyramid and the original image resolution. Most image features are described by a center location in image space, a

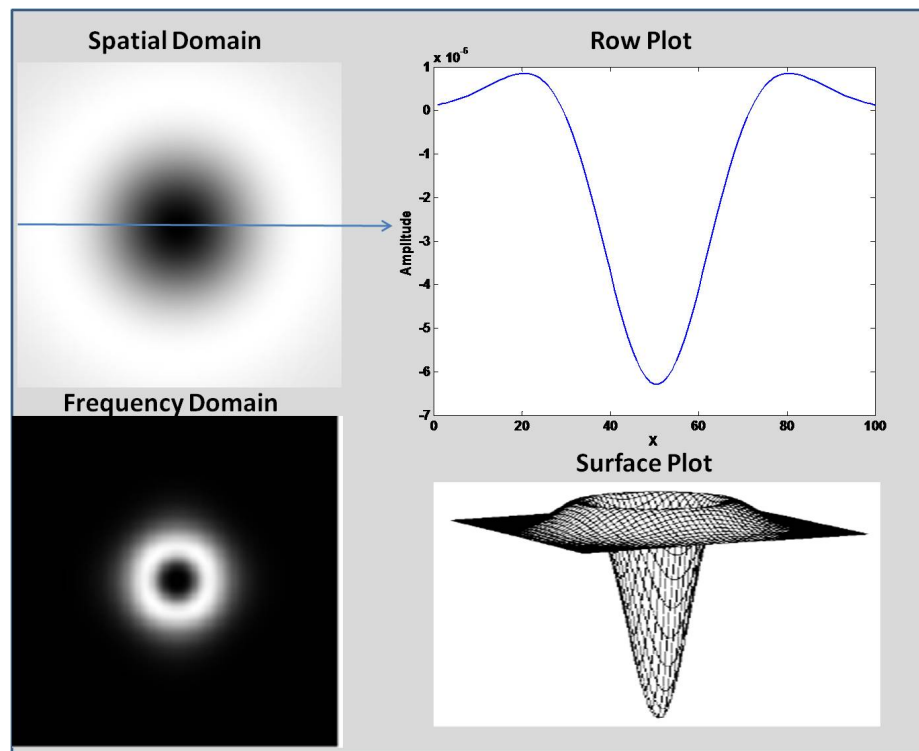


Fig. 7: The Laplacian of Gaussian operator is the result of applying the discrete isotropic second derivative filter to the Gaussian Operator. Its representation in spatial and frequency domain is shown where it represents a pass band filter.

characteristic scale, and a descriptor. The next section details the evolution of the image descriptor generation process for $2D$ data.

5. Invariance Principles and Feature Descriptors

One of the earlier and more famous works which applied the Harris corner detection method to computer vision was that of Zhang [21]. In this work Harris corners were used as feature locations for estimating the epipolar geometry between cameras in a stereo pair. Knowing the exact epipolar geometry between two cameras reduces the dimensionality of the search for point based correspondences from two to one (i.e. a line search). Zhang isolated corner features in both images of the stereo pair and then centered a correlation window on each one. Each corner was then compared to all corners contained in a subregion of the corresponding image inside of which the corresponding point was expected to lie. The correlation score was computed for each comparison and candidate matches whose score was greater than a preset threshold were listed. A relaxation process was then used to disambiguate multiple potential matches for a single corner. Since scale was not taken into account in this work the window size used for correlating corners was a free parameter. Nevertheless, this represented one of the first times a descriptor had been attached to a point based features for the purposes of matching between images. In this case the descriptor was the local sub-image with the corner at its center.

Comparing point based features in this way is problematic for many reasons. First, since the descriptor contains the exact gray values in the vicinity of the point feature, a perfect match is required to contain exactly the same gray values, in the same structure. This requires the perception of the gray scale to be the same from both points of view. Scenes which present this type of photometric behavior are called Lambertian and it should be noted that Lambertian behavior is very rare in space

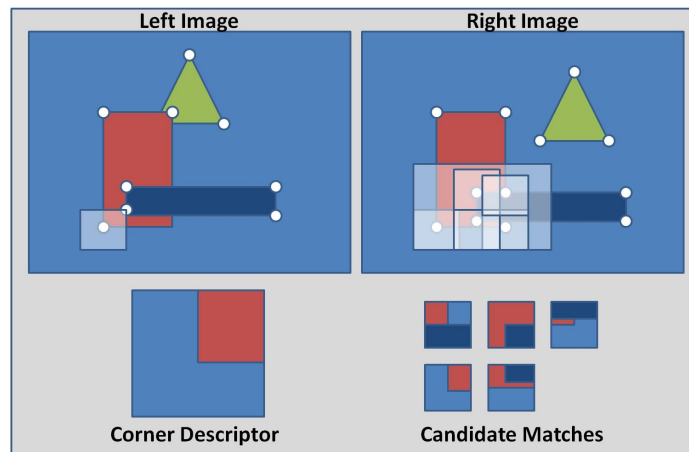


Fig. 8: Candidate corner feature matches in a stereo pair with local area descriptors compared by correlation.

scenes where only a small subset of candidate features may satisfy this constraint. Slight out of plane rotations can change this gray scale distribution and may cause poor correlation in an area that might be the projection of the correct geometric feature causing a false negative response. For a feature to be successfully tracked through a class of transformations of the image data, its descriptor must be “relatively invariant” to that class of transformations. One important logical step is to encode the local image information contained in the neighborhood of a point based feature in a manner where the descriptor is invariant to a single axis rotation about the direction normal to the image plane. A second desired attribute is invariance to scale changes. The Scale Invariant Feature Transform of David Lowe [2,3] addresses both of these attributes, rotational invariance and scale invariance; this approach assigns an orientation to feature points based on the dominant gradient direction and populates feature descriptors by sampling in a reference frame aligned with this feature-defined orientation. In fact this framework for the extraction of features based on scale space extrema, orientation assignment, and the development of scale invariant descriptors represents a significant step forward in image based feature tracking. The next section

explains this algorithm in detail and outlines opportunities for its improvement.

6. The Scale Invariant Feature Transform

The Scale Invariant Feature Transform (SIFT) is one of the most recognized frameworks for multi-scale image feature detection and tracking. In his landmark paper Lowe claimed to have developed a method of extracting features which were invariant to scale and rotation and exhibited “robust matching” performance when image features were subject to affine distortions, the addition of noise, and changes in illumination. The main applications of SIFT-features were intended for were image matching and retrieval, as well as object recognition. Lowe argues that a strength of his technique is that his multi-scale features are well localized in the scale space. This provides estimates for the image position and scale of SIFT features. His algorithm can be broken down into four key processes:

Scale-Space Extrema Detection: A Gaussian Scale space is generated and implemented as a Difference of Gaussian (DoG) Pyramid. Extrema are identified in this data structure by means of non-maxima suppression.

Keypoint Localization: Once coarse locations for extrema are determined in the scale space, a local Taylor-Series (Gauss-Newton algorithm) is used to localize features to sub-pixel and sub-level accuracy in (x, y, σ) .

Orientation Assignment: Each localized feature is assigned an orientation, based on the dominant intensity gradient orientations in the point neighborhood.

Keypoint Descriptor Population: The point neighborhood is sampled with respect to the feature orientation at the correct level in the scale space. A histogram of magnitude weighted gradient orientations is used to populate a 128-D

vector.

Lowe showed that developing a scale space by approximating the Laplacian of Gaussian pyramid by a Difference of Gaussian exhibits similar performance to the scale normalized Laplacian used by Lindeberg [29]. The Difference of Gaussian is given by:

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y) \approx L(x, y, k\sigma) - L(x, y, \sigma) \quad (2.22)$$

Where $G(x, y, \sigma)$ is the Gaussian kernel of equation 2.18, the $*$ indicates the convolution operator. The partial derivative of the image data with respect to the scale parameter σ can then be computed by finite difference approximation.

$$\frac{\partial G}{\partial \sigma} = \sigma \nabla^2 G \quad (2.23)$$

$$\sigma \nabla^2 G \approx \frac{G(x, y, k\sigma) - G(x, y, \sigma)}{k\sigma - \sigma} \quad (2.24)$$

$$(G(x, y, k\sigma) - G(x, y, \sigma)) \approx (k - 1)\sigma^2 \nabla^2 G \quad (2.25)$$

This approximation shows us that the finite difference approximation to the scale derivative of the image actually represents the implementation of the diffusion equation. The same diffusion equation which models the flow of heat or information in thermodynamics. It also shows that the difference of neighboring levels of the Gaussian Pyramid contains the necessary normalization required for scale invariance. Once the scale space is constructed, maxima and minima are localized in the Difference of Gaussian pyramid by comparing each admissible feature location in the pyramid to its eight neighbors in its level, and its nine neighbors in the levels immediately above

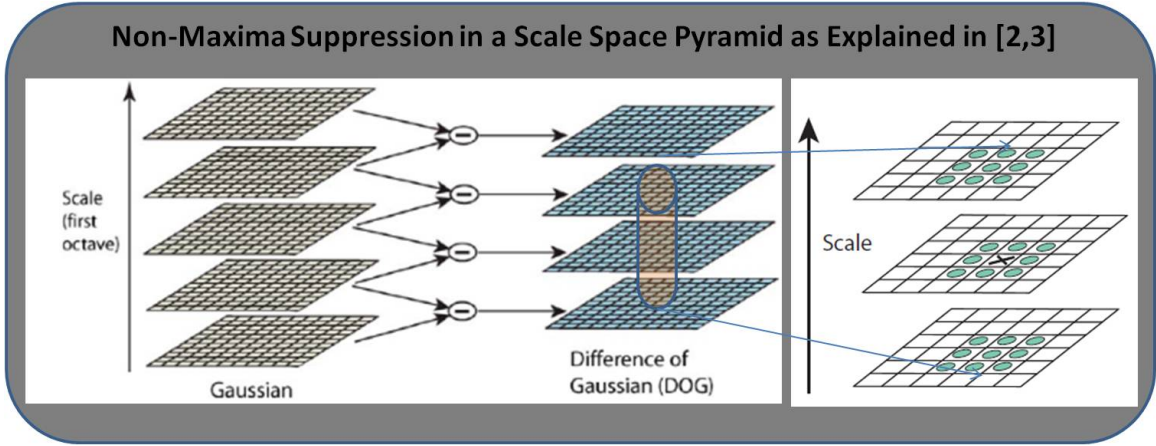


Fig. 9: Non-Maxima Suppression of the Scale Space: Each pyramid location is compared to its 26 neighbors and required to be greater than or less than all of them by a threshold.

and below. If the value $D(x, y, \sigma)$ at that location is greater than or less than all of its neighbors by a preset threshold, it is retained as a keypoint location. This introduces the first tuning parameter of the SIFT algorithm. The contrast threshold restricts feature points to be centered at corners with sufficient gradient magnitude.

Following the coarse (pixel-accurate), feature localization, a sub-pixel interpolation process is applied to each feature location, approximating the feature location by Taylor Series Expansion of the scale space Difference of Gaussian function $D(x)$.

$$D(x) = D + \frac{\partial D}{\partial x} + \frac{1}{2} x^T \frac{\partial^2 D}{\partial x^2} x \quad (2.26)$$

Here x denotes the vector $x = [x, y, \sigma]^T$, the derivative values are approximated by finite differences in the $3 \times 3 \times 3$ neighborhood of the feature point, and the sub-pixel displacement vector is approximately solved for by setting the first derivative of the above function to zero, yielding the Gauss-Newton local minimum estimate (which can be iterated, but usually isn't):

$$\hat{x} = -[\frac{\partial^2 D}{\partial x^2}]^{-1} \frac{\partial D}{\partial x} \quad (2.27)$$

Since features are required to be consistently well localized, edge features, features whose local Hessian matrix have only one strong eigenvalue must be removed. If they are not, experience indicates that subsequent estimation routines will suffer from the localization error associated with the direction with smallest eigenvalue. Lowe borrows the technique from the Harris Corner measure and considers the trace and determinant of the local Hessian matrix to impose a “cornerness” criterion on features. By considering the ratio of the square of the trace to the determinant of the Hessian matrix and writes this constraint as a function of the eigenvalue ratio, $r \equiv \frac{\alpha}{\beta}$:

$$\frac{Tr(\mathbf{H})^2}{Det(\mathbf{H})} = \frac{(\alpha + \beta)^2}{\alpha\beta} = \frac{(r + 1)^2}{r} \quad (2.28)$$

The above expression increases with r , and therefore, for a given value r , a feature showing sufficient “cornerness” will have a value less than the right hand side of 2.28. Once the set of features satisfying the constraints above are localized, a canonical orientation is assigned to the feature. The magnitude and orientation of every sample in a local neighborhood of each feature is computed by the expressions below:

$$m(x, y) = \text{sqrt}[(L(x + 1, y) - L(x - 1, y))^2 + (L(x, y + 1) - L(x, y - 1))^2] \quad (2.29)$$

$$\theta(x, y) = \arctan \left[\frac{L(x, y + 1) - L(x, y - 1)}{L(x + 1, y) - L(x - 1, y)} \right] \quad (2.30)$$

A weighted orientation histogram is then generated using the magnitude of each gradient to weight its contribution. The peaks of this histogram are then used to identify the dominant gradient orientation. For distributions with multiple local maxima in the 360 degree histogram, a duplicate feature is generated for peaks whose

value are within 80 percent of the global maximum. This represents a redundancy in the SIFT framework. The vector $\mathbf{x} = [x, y, \sigma, \theta]^T$ represents a feature frame, i.e. the necessary information required to define the local reference frame of a SIFT feature. Local textural information in the scale space is then encoded in a feature descriptor vector by means of a systematic sampling process aligned with this reference frame. It is this process which identifies the unique local fingerprint of the image data at this point. It will be shown later in this dissertation that consistent sampling in image space turns out to be a restrictive process which can be vastly improved by sampling not in the image space, but rather in the object space, with the associated samples taken along the object space surface being measured.

The SIFT feature vector consists of a 4×4 array of gradient orientation histograms taken from a 16×16 set of samples from the appropriate level of the scale space. These entries to the orientation histogram are weighted by their magnitude as well as a Gaussian function fit over the entire feature neighborhood. Each orientation histogram is divided into 8 orientation bins and the $4 \cdot 4 \cdot 8 = 128$ values populate the feature descriptor. The process is illustrated below over an 8×8 region.

Feature descriptors are then matched in subsequent images by taking the inner product between the feature vectors of a feature of interest and a candidate match. In an effort to be partially invariant to illumination changes, the descriptor value is normalized to be a unit vector, then any components with value greater than 0.2 are set to 0.2 and the vector is re-normalized. The tuning value 0.2 is a tolerance frequently used in SIFT implementations. The reasoning behind this is that the magnitude of the image gradients change with local illumination change but the gradient orientations are somewhat unaffected. In this way the feature descriptor can still be matched over small variations in image intensity. SIFT features have proven very useful for image retrieval, and have been used successfully in motion tracking

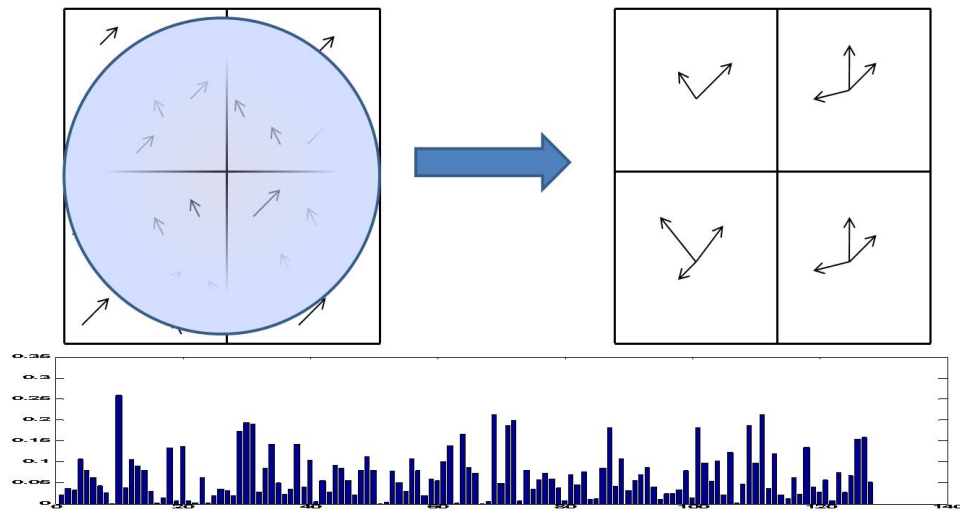


Fig. 10: Local sampling of gradient orientations occurs in the appropriate pyramid level associated with the feature scale, and with respect to the reference orientation of the feature.

and structure from motion applications where displacement between frames are small. The various approximations and decisions implicit in Lowe’s SIFT algorithm result in a fairly widely applicable approach. However, the actual degree of robustness to the features to scale and rotation variations and robustness to scene and lighting variations is difficult to predict prior to applying the algorithm to specific images. In truth, there is substantial room for improvement.

Efforts to improve upon SIFT’s utility and computational efficiency have included PCA-SIFT [31], the Speeded-Up Robust Features (SURF) [32], among many others including the exciting work of T.W.R. Lo who has adapted the SIFT technique to range images. One of the elusive issues is the accuracy with which the feature is effectively located in image or object space. The accuracy is known to be a function of “everything” (the texture of the scene, the illumination, the pose of the sensor, occlusions, etc. as well as rotation and scale variations all of which

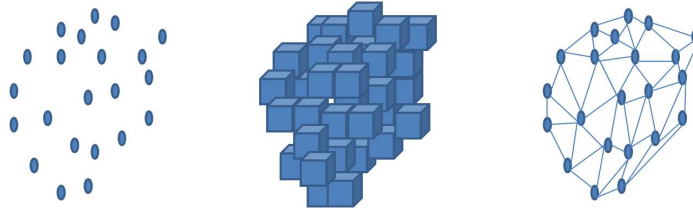


Fig. 11: An example of $3D$ data representations, Left: a point cloud, Middle: Voxels representing volume data, Right: A surface mesh resulting from a Delaunay Triangulation

are application-dependent). Thus, all feature detection methods can benefit from *a posteriori* statistical analysis to characterize errors and reject spurious features. The approach taken herein is to use the residuals from the least squares OLTAE algorithm to provide this needed insight.

B. Developments in $3D$ Feature Extraction

Three dimensional data has many forms and methods of acquisition. The most common types include raw point clouds which are simply a list of the coordinates of a collection of 3-vectors, volume elements or voxels which can be thought of as an occupancy grid of a partitioned volume of $3D$ space, and surface meshes which include information regarding a derived connectivity between $3D$ points. A simple example of these data types is shown in figure 11.

Voxel data is the data type that is most similar to image data. A volume of $3D$ space representing the domain for the measured data is partitioned into cuboid elements of constant size. Each volume element represents a location relative to a $3D$ sensor, and may take on a value indicating occupancy, or some other scalar indicating intensity for example. Voxel based renderings have in fact been referred to as $3D$ images. Point cloud data is typically the standard raw measurement form for $3D$ data acquisition systems. No assumptions are made with respect to the adjacency

of measurements contained in a point cloud representation. Usually some processing is required to determine this information. One method, the Delaunay Triangulation determines the adjacency between points in a point cloud by associating groups of 4 points by constraining the tetrahedron composed of the edges connecting each point pair. Each admissible triangle must come from a tetrahedron which contains no other data point within its circumsphere. Though there are efficient methods for computing meshes, there can be ambiguous solutions, and the process is computationally intensive as the number of measurements increases. This motivates the need for a meshless approach to scale space analysis of $3D$ data and the use of raw point clouds as the representation of choice for $3D$ data in this dissertation. Three dimensional data may be generated passively by stereo-vision techniques or actively by RADAR, SONAR, or LIDAR which measures the time of flight from the sensor to scene points with range vector direction measured by the two angular variables (azimuth and elevation in the sensor frame). It is well known that range sensors generate spurious range measurements, so the use of raw point clouds must anticipate the necessity for data filtering/smoothing. The developments presented in this section apply in general to $3D$ data, however we will focus on the concept of a “range camera” or “range image” which represents an array of range measurements taken by a scanning LIDAR or other $3D$ acquisition device.

1. Smoothing $3D$ Data

Raw LIDAR measurements are noisy with the variance of the noise being dependent on the particular sensor, the object being imaged, as well as atmospheric conditions in non-vacuum applications. It is often desired to determine local geometric parameters in $3D$ datasets, however, such computations are sensitive to noise. As is the case in image processing, the approach to determining gradients of any mea-

sured field variable, for example, is usually enhanced by a smoothing step. In very much the same spirit, $3D$ data can be smoothed to determine reliable estimates of surface normals and curvature values. The level of smoothing is dictated by some estimate of sensor noise in the particular application. When processing $2D$ data, a window was defined as a domain for local computations. Due to the discrete nature of digital images this window was square, but an isotropic weighting was applied to minimize these effects. One common way to define a neighborhood about a point in $3D$ space is to specify a radius in the object space from a reference point and determine all points in the data set which are within the specified distance to the reference point. This characteristic distance must be carefully chosen to ensure that the correct number of neighboring measurements are related. Too few neighbors and the estimates can be inaccurate due to small sample statistics, too many, and each computation is too intensive. Clearly the level of point cloud sparsity relative to measurable surface geometry variations is a key issue. If many points are so close together that they are essentially the same physical point, a simple average will suffice to give the best estimate. A method is needed to specify the number of neighbors desired, evaluate the distances to each data point and retain the desired number, taking into account the measurement error and the local density of the point cloud. This method can be problematic as it is not a strict definition of a locality in $3D$ space. Depending on the point distribution, very large distances can be used to compute estimates for geometric values which are intended to describe a small region of space.

Just as in the smoothing of image data, smoothing can be applied to scalars defined on $3D$ data sets by replacing the coordinates of a point by an average of its current value and weighted operation on the values contained in the neighborhood. This averaging process is very similar to the process of convolution. In practice, smoothing the shape of the measured surface, the $3D$ position of each measurement is

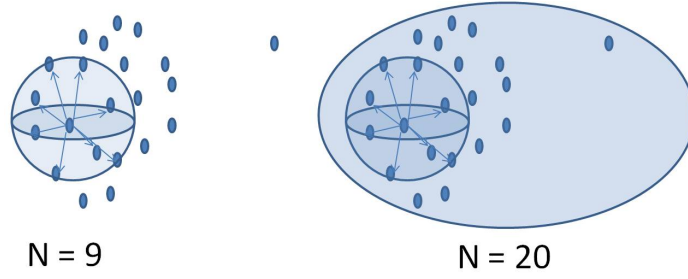


Fig. 12: An example of 3D neighborhood definitions, Left: a spherical radius with 9 interior points, Right: $K = 20$ nearest neighbors containing 1 point at a large distance with the $K = 9$ neighborhood shown for comparison.

frequently displaced by a surface evolution process. Surface evolution can be modeled by the following equation:

$$\frac{\partial P}{\partial t} = F(P) \quad (2.31)$$

Integrating the surface evolution equation we can observe the application of a surface evolution process such as the simplest linear function:

$$P_{k+1} = P_k + F(P_k)\Delta t \quad (2.32)$$

Here P_k represents the point cloud locations of all measurements before the application of the surface evolution process, $F(P_k)$ represents the application of a local smoothing/interpolation operator on a subset of points in the point cloud, and Δt represents a step in the diffusion process. Recall that in image diffusion the variable t was related to the scale of the diffusion operator. The diffusion displacement can be defined as the change in the surface resulting from the application of a diffusion operator:

$$\Delta P_k = P_{k+1} - P_k = F(P_k)\Delta t \quad (2.33)$$

This change in the surface over time is implemented by applying a diffusion operator

to the data. A very common operator used in this setting is the so-called umbrella operator shown below:

$$L(P) = \frac{1}{n} \sum_{i=1}^n P_i - P \quad (2.34)$$

where P_i is the i 'th of the n neighbors around P . The use of the umbrella operator is the most basic case of surface evolution by Laplacian smoothing. The degree to which smoothing occurs for each step Δt is usually controlled by the parameter λ and it is also common to weight the importance of each neighbor based on their distance to the point being smoothed. The surface evolution equation for Laplacian smoothing along with the expression for the weighted umbrella operator are given in equations 2.35 and 2.36.

$$P_{k+1} = P_k + \lambda L(P_k) \quad (2.35)$$

$$L_w(p) = \frac{1}{\sum w_i} \sum_{i=1}^n w_i P_i - P \quad (2.36)$$

Again a common weighting used for the weighted umbrella operator is the Gaussian weighting. For surface meshes, the neighborhood can be defined as an “ n -ring” of neighbors. An “ n -ring” of neighbors are all points that require only n edges be traversed to reach point P . Simple smoothing algorithms simply set n to 1. For this case the choice of inter-point distance used to compute the Gaussian weighting used in the weighted umbrella operator is inconsequential. For larger values of n it becomes important to determine if the Euclidian distance is appropriate, or if a surface based geodesic distance is necessary. An example of the application of Laplacian smoothing is shown in figure 13.

Laplacian smoothing as defined can cause surface measurements to drift tangentially along the surface causing a regularizing effect on the measurements. This means

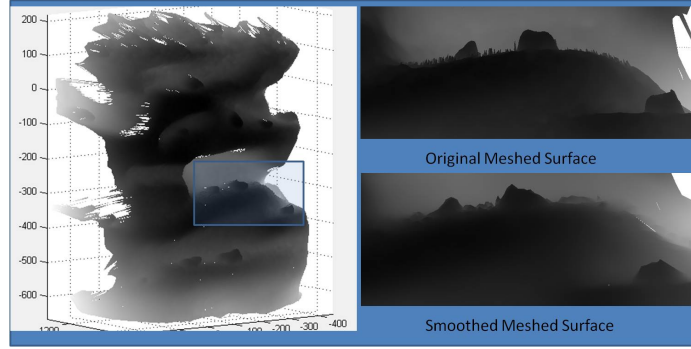


Fig. 13: An example of Laplacian smoothing applied to surface data

that for successive application of the umbrella operator, the object-space distribution of inter-point distances is not necessarily preserved. It has been shown that even for points measured on a plane, where zero motion is desired, points will tend to drift until evenly distributed. This characteristic is not desirable. A second process used for the smoothing of surface data is known as Mean Curvature Flow.

To avoid the undesired tangential drift which occurs with the use of the umbrella operator and Laplacian smoothing, mean curvature flow uses the following formulation of the surface evolution equation:

$$P_{k+1} = P_k - H\hat{n}\Delta t \quad (2.37)$$

where H is the mean curvature at each point in the data set and \hat{n} is the unit normal to the surface at each point. Per point estimates for the mean curvature values and surface normal directions can be determined in several ways including principle component analysis as well as computing approximations to surface derivatives and using the techniques of differential geometry. Implicitly, we know the accuracy of the measurements, their spatial density, and the scale associated with underlying observable features must be considered, at least heuristically in establishing good smoothing algorithms.

2. Differential Geometry

Differential geometry [13] is concerned with the rigorous geometric analysis of curves and surfaces. This discipline has wide ranging applications, including general relativity, electromagnetics, and Lagrangian and Hamiltonian mechanics. We are concerned here with the developments which stem from the application of differential geometry to the geometric modeling of poorly known, discretely measurable surfaces. Consider the a surface in 3-space parameterized in the following way:

$$\vec{r}(u, v) = [x(u, v), y(u, v), z(u, v)]^T \quad (2.38)$$

where \vec{r} is the vector from the origin to an arbitrary point in $3D$ space, and (u, v) represent two orthogonal directions in a $2D$ parameter space which describe the embedding of the $3D$ data. This is exactly the situation explained in chapter 3 where (u, v) represent steps in elevation and azimuthal angle respectively. It is assumed that the measured surface is at least twice differentiable, in which case the first two fundamental forms of differential geometry apply. As is also discussed in chapter 3, this assumption need only be satisfied locally within the domain of the smoothing operator which leads to the operator needing to be at least a 5×5 window in the parameter plane. Of course it is implicit that the deviation of the measured surface from this assumption may lead to over-smoothing of sharp/discontinuous geometry. The first fundamental form is given by:

$$\mathbf{I}(u, v, du, dv) = d\vec{r} \cdot d\vec{r} = E du^2 + 2F du dv + G dv^2 \quad (2.39)$$

where, E, F, and G are given by $\vec{r}_u \cdot \vec{r}_u$, $\vec{r}_u \cdot \vec{r}_v$, and $\vec{r}_v \cdot \vec{r}_v$ respectively. The first order partial derivatives of the position vector with respect to the embedding parameters u and v are given by r_u and r_v , and represent a basis for the local tangent plane

of a point. The first fundamental form measures the change in $d\vec{r} = r_u du + r_v dv$ with respect to a change in du and dv , and is considered an intrinsic invariant. This means that the value of the first fundamental form will be the same for a surface regardless of its embedding in $3D$ space. For analytical manifolds which may be consistently sampled, this useful fact results in point based descriptions that are invariant of viewpoint. However, in the case of a LIDAR, the sampling of the surface is a function of the relative position and orientation of the surface with respect to the LIDAR, and the first fundamental form is not necessarily invariant of sampling. The redundancy of sampling with some modern sensors, however helps to compensate for lack of uniformity. The second fundamental form of differential geometry describes the extrinsic properties of the surface embedding in $3D$ -Space. It is given by:

$$\mathbf{II}(u, v, du, dv) = -d\vec{r} \cdot d\vec{n} = Ld_u^2 + 2Md_ud_v + Nd_v^2 \quad (2.40)$$

where L , M , and N are given by $\vec{r}_{uu} \cdot \vec{n}$, $\vec{r}_{uv} \cdot \vec{n}$, and $\vec{r}_{vv} \cdot \vec{n}$ respectively. The r_{ii} are the second order partial derivatives of the position vector and \vec{n} is the surface normal vector:

$$\vec{n} = \frac{r_u \times r_v}{\|r_u \times r_v\|} \quad (2.41)$$

Two additional equally important and widely used surface parameters in differential geometry are the Gaussian (K) and Mean (H) curvature values:

$$H = \frac{1}{2} \frac{EN + GL - 2FM}{EG - F^2} \quad (2.42)$$

$$K = \frac{LN - M^2}{EG - F^2} \quad (2.43)$$

In terms of the principle curvatures (k_1, k_2) in the parameter plane:

Table 2: Classification of surface types based on the Mean and Gaussian Curvature Values.

	$K(i, j) > 0$	$K(i, j)=0$	$K(i, j) < 0$
$H(i, j) < 0$	Peak	Ridge	Saddle Ridge
$H(i, j) = 0$	Undefined	Flat	Minimal Surface
$H(i, j) > 0$	Pit	Valley	Saddle Valley

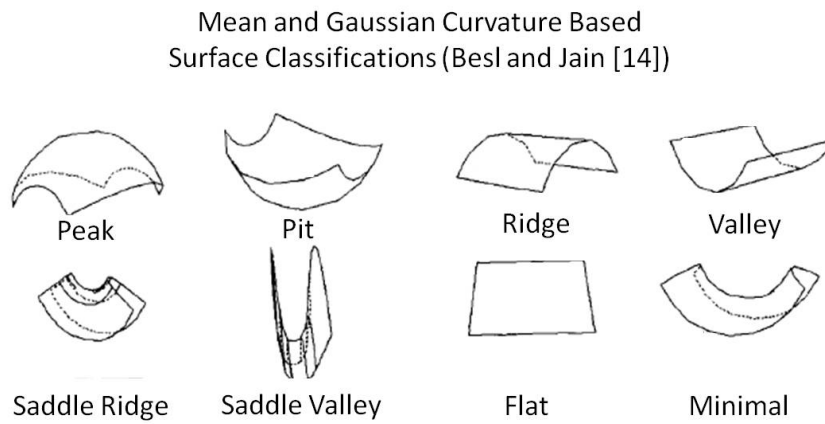


Fig. 14: Surfaces types classified by Mean and Gaussian Curvatures as indicated by Table 2.

$$K = k_1 k_2 \Rightarrow k_2 = \frac{K}{k_1} \quad (2.44)$$

$$H = \frac{k_1 + k_2}{2} \Rightarrow k_1 = H \pm \sqrt{H^2 - K} \quad (2.45)$$

The classifications of Table 2 have been used by Besl and Jain [15] to segment surfaces into 8 types. These surface types range from locally convex (peak) to locally concave (pit) and their classification with respect to their Gaussian and Mean curvature values is provided in Table 2; these are illustrated graphically for reference in figure 14.

The common denominator in the equation for both mean and Gaussian curvature

is the expression $EF - G^2$ which can be shown to be the determinant of the positive definite matrix:

$$\begin{bmatrix} E & F \\ F & G \end{bmatrix} = \begin{bmatrix} \vec{r}_u \cdot \vec{r}_u & \vec{r}_u \cdot \vec{r}_v \\ \vec{r}_u \cdot \vec{r}_v & \vec{r}_v \cdot \vec{r}_v \end{bmatrix} \quad (2.46)$$

The matrix of equation 2.46 is commonly called the Riemannian metric and is defined at each surface point. Note that the Riemannian metric is somewhat analogous to the structure tensor used for corner detection in image processing. This similarity indicates that using local curvature extrema in range data will be analogous to corner detection in imagery. Thus the cornerness criteria of $2D$ is readily extended to the parameter plane but here it indicates to what degree a point on a surface is displaced from its neighbors. If at a point we may locally describe the surface by single valued functions of x and y , we may re-write the surface parameterization in the form:

$$\vec{r} = [u, v, r(u, v)]^T \quad (2.47)$$

Using this form, the expressions for r_u , r_v , r_{uu} , r_{vv} , and r_{uv} can be simplified to:

$$\vec{r}_u = [1, 0, f_u]^T; \quad (2.48)$$

$$\vec{r}_v = [0, 1, f_v]^T; \quad (2.49)$$

$$\vec{r}_{uu} = [0, 0, f_{uu}]^T; \quad (2.50)$$

$$\vec{r}_{vv} = [0, 0, f_{vv}]^T; \quad (2.51)$$

$$\vec{r}_{uv} = [0, 0, f_{uv}]^T; \quad (2.52)$$

Inserting these expressions into the expressions for Gaussian and mean curvature we obtain:

$$H = \frac{(1 + f_v)f_{uu} + (1 + f_u^2)f_{vv} - 2f_u f_v f_{uv}}{2(1 + f_u^2 + f_v^2)^{3/2}} \quad (2.53)$$

$$K = \frac{f_{uu}f_{vv} - f_{uv}^2}{(1 + f_u^2 + f_v^2)^2} \quad (2.54)$$

In the case that the surface is a fronto-parallel plane, these curvature equations reduce to the curvature and the determinant of the hessian respectively. The determinant of the hessian was used to evaluate candidate extrema in the difference of Gaussian scale space for effective localization. Note that the expression $(1 + f_u^2 + f_v^2)^2$ is equal to determinant of the Riemannian metric of the surface $EG - F^2$ which for a fronto-parallel plane ($f_u = 0, f_v = 0$) is equal to 1. In this way, we can relate the expression for an area element dA defined at a point to (d_x, d_y) .

$$dA = \sqrt{EG - F^2} dx dy \quad (2.55)$$

Notice that the square of the surface area of an infinitesimal surface element is proportional to the assumption of a locally orthogonal plane at a point, and the size of individual surface elements grows with increasing surface gradients. This is an important fact and can be a large source of localization error. *3D* surface points displaying high values of mean curvature make good candidates for *3D* feature locations. Similar to how Harris Corners which display large eigenvalues, represent locations where large changes in the texture occur for small displacements. It is at these sharp curvature locations where we define geometric features. Before point based correspondences were used to register *3D* point clouds, exhaustive search techniques like the Iterative Closest Point method of Besl and McKay [33] or Chen and Medioni [34] were used, requiring N^2 comparisons where N is the number of points in the model. These methods require a good initial guess for accurate results because they are prone to converging to local minima and can therefore lead to an incorrect registration. One

very successful method used for creating point based descriptors in 3D point clouds is the Spin Image technique introduced by Andrew Johnson [6, 7].

3. 3D Point Based Registration by Spin Images

Spin Images can be defined at every point in a 3D data set, given that a means exists for identifying a surface normal at each point. This is usually accomplished by nearest neighbor indexing or adjacency information in a 3D mesh. An *Oriented Point* is defined by its cartesian coordinates and its surface normal vector. The surface normal is usually computed by fitting a plane to the point of interest and its “n-ring” neighbors. An alternative method is to take the eigenvector associated with the least eigenvalue of the covariance matrix of a *symmetric neighborhood* about a particular point. In this case a symmetric neighborhood is defined by involving all neighbors of a point which are less than a specified distance away in the computation of the normal. The distance metric used can be Euclidian or geodesic. In both cases it is important to have a regularized mesh, and in his dissertation, Johnson provides a method for controlling mesh resolution. The five degree of freedom basis is defined by the 3D coordinates of a measured point, and the azimuth and elevation values which define the direction of the surface normal in the measurement frame. An example of an oriented point is illustrated in Figure 15.

A Spin Image projects the participating points into a cylindrical coordinate system with origin at the point of interest. The computed coordinates are the axial distance (α) from the normal line and the perpendicular displacement (β) from the local tangent plane. These are the radius and height in cylindrical coordinates respectively. This technique can be both global, and local depending on the chosen parameters of the Spin Image. Constraints can be applied regarding the extent of the local neighborhood, the sampling frequency in α and β , and the support angle, i.e.

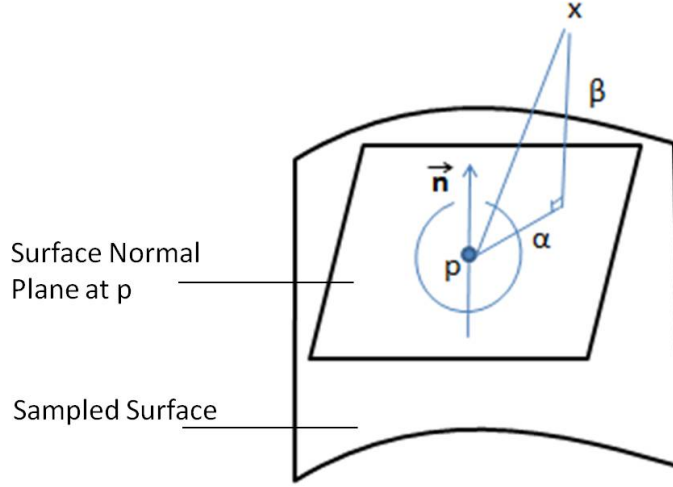


Fig. 15: An example Oriented Point consisting of a position and direction in space. In this basis only the in-plane radial distance α and the axial distance from the local tangent plane β are needed to describe a position in this frame.

the allowable deviation in the surface normal of a participating point from the normal of the interest point. The equations for computing the local cylindrical coordinates of a Spin Image for a point of interest p and an arbitrary neighboring point x are given below:

$$\alpha = \sqrt{\|\vec{x} - \vec{p}\|^2 - (\vec{n} \cdot (\vec{x} - \vec{p}))^2} \quad (2.56)$$

$$\beta = \vec{n} \cdot (\vec{x} - \vec{p}) \quad (2.57)$$

Spin Images can be very effective in providing a good preliminary registration of two meshed models. All of the techniques available to 2D template matching, such as normalized cross correlation can be used to compare two Spin Images. However, model registration based on Spin Images has historically relied on exhaustive search or random sampling techniques to achieve registration. This can be very time consuming, especially for large Spin Images. Recently, Ho and Gibbons [11] applied

the technique of defining Spin Images at mesh locations exhibiting high Koenderink Curvedness and saw vastly improved efficiency. Additional variations of the Spin Image technique have also employed multi-resolution analysis [9], and texture [8]. The application of scale-space to the Spin Image technique is still an open problem. The SIFT technique has been extended to range images with some success and represents a competing point based descriptor. Other point based descriptors are also described in [35, 36].

4. Applying the Scale Invariant Feature Transform to Range Data

Within the last few years, Lo and Seibert [4, 5] have applied the SIFT technique of Lowe [2, 3] to range images. In their application, the range images were generated by using a stereo vision system with projected structured light. The data contained in the segmented range values of the measured object are first normalized to have unit standard deviation and zero mean in their z-coordinate. The Gaussian and Mean curvatures of the surface data are then computed by image gradients. The principle curvatures are then determined by equations 2.44 and 2.45 and a variant of the shape index is computed for each location in scale space by:

$$s = \frac{2}{\pi} \arctan \left[\frac{k_2 + k_1}{k_2 - k_1} \right] \quad (2.58)$$

Keypoints are then localized in the scale space by the same method as Lowe [2, 3] but a new descriptor is populated using shape index values. This provides a means of extracting scale space extrema from 3D surface data, which can allow for feature points to be associated with a point cloud or mesh without being a member of the original set of measured points. In this work, the 3D smoothing of the surface data is not considered, and therefore the resulting localization of features in the object

space can suffer increasing inaccuracies with increasing scale. A framework which further exploits $3D$ information to account for inconsistent sampling of an object and describes both geometric and photometric extrema in the object space would be even more powerful. Applying the diffusion equation to noisy surface data and captured image data along with developing a descriptor based on mean curvature flow will result in an effective means for localizing object-space features based on geometric or textural properties. The following chapters outline a framework to achieve this end.

CHAPTER III

SENSOR CHARACTERIZATION AND SYSTEM CALIBRATION

Information on the geometry and surface texture of a proximal body can be measured by a variety of methods. These methods can be passive in the case of camera observations or active for sensors such as Light Detection and Ranging (LIDAR) devices. While many other sensors such as Doppler RADAR or flash LIDAR also provide 3D data, this dissertation is focused on the fusion of rapid scanning LIDAR (typically millions of measurements per second) and High Definition camera imagery. An active sensor uses energy to measure the world around it. In the case of a LIDAR, typically, a pulse of monochromatic light is generated and directed toward the object to be measured. The round trip time of flight is recorded by an internal clock which is triggered at the time of the out going pulse, and triggered again when a photo-detector senses a sufficient number of photons in that wavelength returning from the scene. More generally, the wave form of the sent and received light pulse may be convolved with the sent pulse to derive the relative phase and thereby obtain higher accuracy LIDAR measurements. The resulting range measurement is calculated as the product of half the round-trip flight time and the speed of light. The resulting output of a scanning LIDAR is a 3D point cloud, a set of measured points which report the measured range along a line of sight from the system origin. In contrast to the active measurement approach of the LIDAR, a camera is a passive optical system. Light in this case passes through a lens or optical train, and incident energy is integrated in the pixels of a Charged Coupled Device (CCD) or a Complementary Metal Oxide Semiconductor (CMOS) pixel array. In this case, the reflected energy from the object of interest must be the source of the measured photons. The imaging device collects

photons over an integration period, usually a few tens of milliseconds, and reports the total received energy of each element in the array. The sections that follow will explain both of these sensor models in greater detail, introduce the current and future sensors used in the Land Air and Space Robotics laboratory, and explain a method for co-registering their outputs after each sensor has been individually calibrated.

A. Measuring Geometry by Light Detection and Ranging

Measurements made by Light Detection and Ranging systems most commonly report the distance to a scene point along a line of sight vector. An individual measurement is commonly reported as a vector magnitude, $\|r\|$ and a unit vector direction \hat{r} . A list of these measurements taken from a stationary LIDAR system may be converted to another form. Consider a spherical coordinate system located at the origin of the LIDAR, in the case of a scanning LIDAR, the origin is the center of projection located at the point of intersection of the two spin axes on the plane of the scanning mirror.

The range vector direction is defined by the azimuth (θ), and elevation (ϕ) angles which may be determined from the components of the range unit vector:

$$\theta = \arctan \left[\frac{r_x}{r_z} \right] \quad (3.1)$$

$$\phi = \arctan \left[\frac{r_y}{r_z} \right] \quad (3.2)$$

where r_x, r_y, r_z are the horizontal, vertical, and heading directions of the unit range vector respectively. The selection of the components in this manner is consistent with a camera coordinate system which chooses the z-direction to be away from the camera along the bore sight. This geometry is a slight variation of the standard azimuth, elevation geometry which is given by standard spherical coordinates:

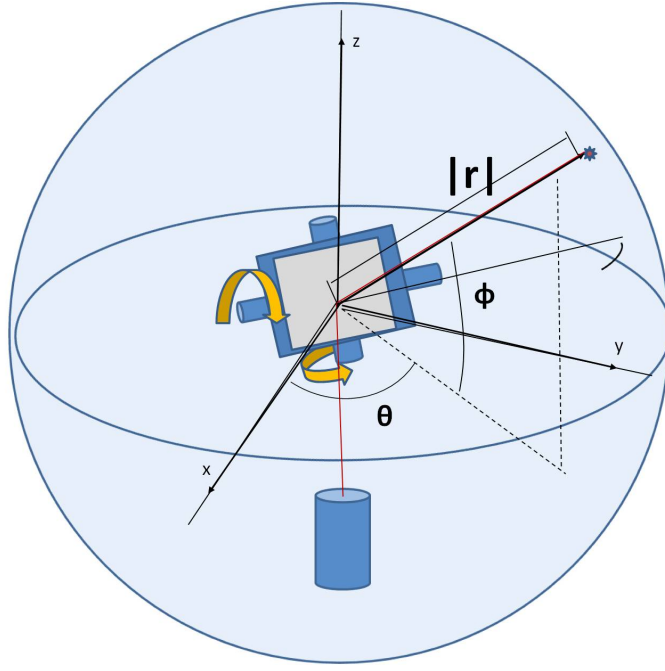


Fig. 16: A two axis scanning mirror as the origin of a spherical coordinate system.

$$\theta = \arctan \left[\frac{r_y}{r_x} \right] \quad (3.3)$$

$$\phi = \arcsin r_z \quad (3.4)$$

$$r_x = \cos \phi \cos \theta \quad (3.5)$$

$$r_y = \cos \phi \sin \theta \quad (3.6)$$

$$r_z = \sin \phi \quad (3.7)$$

In this case the z-axis is the direction associated with $\phi = 0$ and $\theta = 0$ and the x and y axes are orthogonal. The resulting equations for the orthogonal unit vector components are given by:

$$r_x = \frac{\tan \theta}{\sqrt{1 + \tan^2 \theta + \tan^2 \phi}} \quad (3.8)$$

$$r_y = \frac{\tan \phi}{\sqrt{1 + \tan^2 \theta + \tan^2 \phi}} \quad (3.9)$$

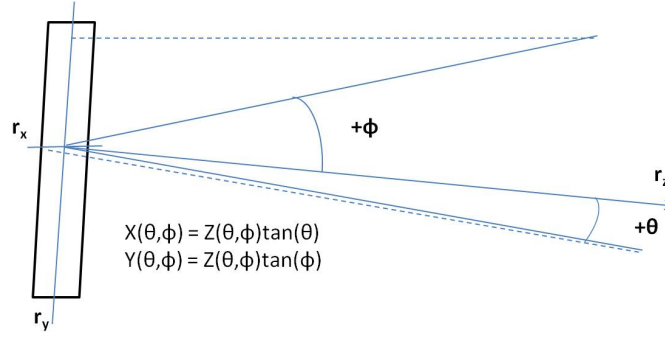


Fig. 17: This diagram illustrates the slight variation from traditional spherical coordinates to orthogonal axis coordinates.

$$r_z = \frac{1}{\sqrt{1 + \tan^2 \theta + \tan^2 \phi}} \quad (3.10)$$

These orthogonal axis angular coordinates are convenient to use in conjunction with the camera coordinate system convention where the z-axis is the direction of the bore sight, and the x and y axis are the orthogonal displacements from that direction. In both coordinate systems the z-axis points in the direction of increasing depth from the sensor and displacements in the image plane can be thought of as angular displacements from the center line of sight direction. Figure 17 illustrates this geometry.

For a two axis scanning mirror, it is easy to represent a set of measurements as a rectangular array which reports the range as a field $r(\theta, \phi)$. Each element in the array represents a consistent step $d\phi$, $d\theta$ and breaks down the scanner field of view into a specified number m rows and n columns. Range data in this form is called a range image. An example range image of an asteroid model is shown in figure 18. In contrast to a disparity image obtained by stereo vision techniques, the range to the 3D points represented by the gray scale value in each pixel are directly measured, and the measured azimuth and elevation angle data is part of the original measurement. For a range image derived by stereo techniques, the 3D point is estimated, and the angular accuracy is limited by the resolution of the camera. Many LIDAR systems also report a gray scale value indicating the amount of energy from the original light

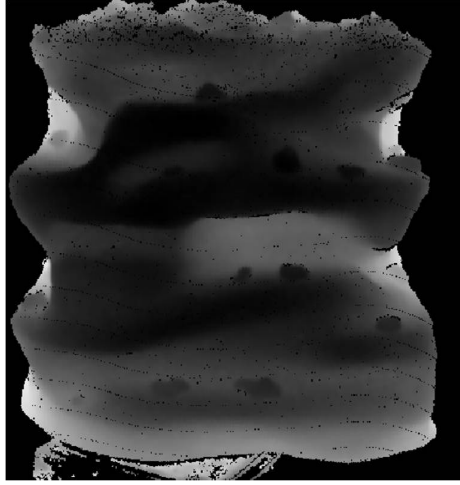


Fig. 18: An example range image, a $2D$ array of range values taken by a stationary scanner.

pulse that returned to the photo detector. This gray scale is referred to as the Return Light Intensity and is directly related to the quality of a LIDAR measurement. The Light Return intensity is a function of the interaction of the laser spot with the measured surface and therefore is sensitive to the incidence angle of the measurement as well as material properties such as surface texture, absorption, and diffusivity. An example image of this texture field is shown for the asteroid model in figure 19.

The Return Light Intensity has seen little use in practice but has been used as an indicator of measurement confidence as well as a texture field. In this dissertation, the Return Light intensity is used primarily as a texture field for feature extraction. The second texture field is provided by the passive observation of the visible light as measured by a camera.

B. Passive Field Measurements from Camera Imagery

Cameras are passive observers. Photons from observable light sources, including reflected photons from surfaces are projected through a lens or optical train to a CCD



Fig. 19: Return Light Intensity Data for the Asteroid Range Image.

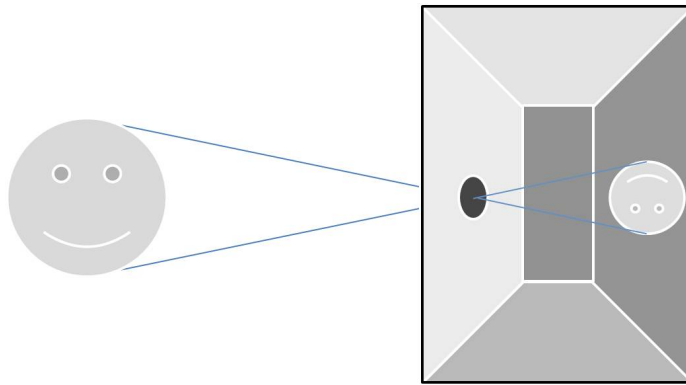


Fig. 20: A simple representation of a “Camera Obscura.”

or CMOS device. The geometry of this process can be quite complex, but through a process known as calibration, data obtained by many camera systems can be mapped to a common geometric model, the pinhole camera. The pinhole camera model as its name suggests describes the passage of light through a point or small circular aperture. In Latin, camera means vaulted or chambered room and the early cameras or “Camera Obscura” were literally dark rooms with light passing through a small hole and projecting to a screen [37].

The pinhole camera model relates Object space points $\vec{x} = [x, y, z]^T$ to image

points $[i_x, i_y]^T$ by:

$$i_x = f \frac{x}{z} + p_x \quad (3.11)$$

$$i_y = f \frac{y}{z} + p_y \quad (3.12)$$

where f is the focal length of the camera, the distance of the center of projection of the lens to the imaging plane, and the principle offsets (p_x, p_y) indicate the intersection of the camera principle axis with the image plane. We can conveniently represent this projection in matrix form using homogeneous coordinates $\vec{X} = [x, y, z, 1]^T$:

$$\begin{bmatrix} fy + p_x z \\ fy + p_y z \\ z \end{bmatrix} = \begin{bmatrix} f & 0 & p_x & 0 \\ 0 & f & p_y & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \begin{bmatrix} x \\ y \\ z \\ 1 \end{bmatrix} \quad (3.13)$$

Notice here that the original relationships of equations 3.11 and 3.12 are obtained by dividing equations 1 and 2 by row 3. A more general form of observation equation can be obtained if we consider a camera whose measurement frame and origin are displaced with respect to a reference coordinate system. In this way, an arbitrary point would have coordinates:

$$\vec{x} = \begin{bmatrix} x - x_c \\ y - y_c \\ z - z_c \end{bmatrix} \quad (3.14)$$

where $[x, y, z]^T$ are the cartesian coordinates of an arbitrary point in the reference coordinate system, and $[x_c, y_c, z_c]^T$ are the cartesian coordinates of the camera system origin. Considering also that the camera may be rotated with respect to the reference

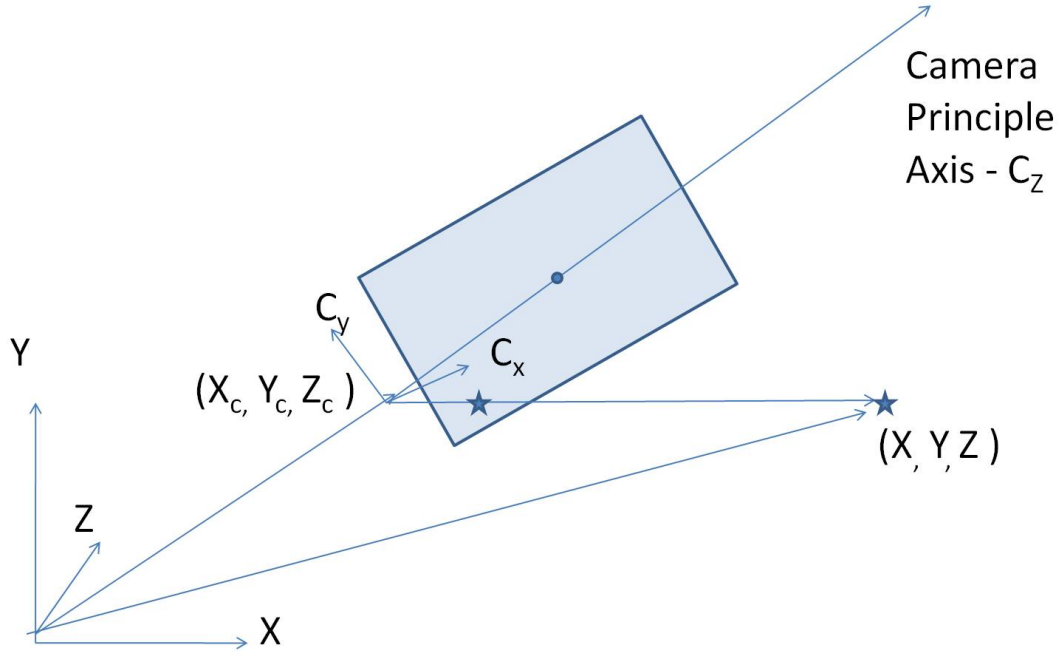


Fig. 21: The Colinearity equations describe the pinhole camera with respect to an arbitrary reference frame.

coordinate system the following co-linearity equations express the requirement that an arbitrary scene point, the camera center of focus, and its projection to the image plane all lie on the same ray.

$$\begin{bmatrix} i_x \\ i_y \end{bmatrix} = \begin{bmatrix} p_x + f \left[\frac{R_{1,1}(x-x_c) + R_{1,2}(y-y_c) + R_{1,3}(z-z_c)}{R_{3,1}(x-x_c) + R_{3,2}(y-y_c) + R_{3,3}(z-z_c)} \right] \\ p_y + f \left[\frac{R_{2,1}(x-x_c) + R_{2,2}(y-y_c) + R_{2,3}(z-z_c)}{R_{3,1}(x-x_c) + R_{3,2}(y-y_c) + R_{3,3}(z-z_c)} \right] \end{bmatrix} \quad (3.15)$$

Calibrating a camera usually requires the determination of the intrinsic and extrinsic parameters. The intrinsic parameters can be described by the left 3x3 matrix of the projection matrix in 3.13.

$$K = \begin{bmatrix} f & s & p_x \\ 0 & f & p_y \\ 0 & 0 & 1 \end{bmatrix} \quad (3.16)$$

The intrinsic parameters include the focal length f , the principle offsets (p_x, p_y) , and the skew parameter s which is zero for most cameras today. Once these parameters have been determined the projection matrix of equation 3.13 may be simply written as:

$$\vec{x} = K[I_{3 \times 3} | 0_{3 \times 1}] \vec{X} \quad (3.17)$$

In this form we can observe that the remaining identity matrix and zero vector in the projection matrix describe a camera which has its coordinate system aligned with the reference coordinate system and whose origin is co-located with the origin of the reference coordinate system. In general, a camera is rotated and translated with respect to some frame of interest in a particular application, and the determination of the extrinsic parameters, the rotation and translation of the camera coordinate system with respect to the frame of interest is called an extrinsic calibration. For the application of this dissertation, it is desired to determine the cameras position and orientation with respect to the LIDAR coordinate system.

C. Camera to LIDAR Extrinsic Calibration

The standard approach to camera calibration involves the measurement of world scene points in the coordinate system of a calibration object. Common calibration objects include one, or multiple planes with an array of high contrast features in known locations. For example, a chess board with black and white squares of known

size is a frequent choice. In a systematic way, a scene point is associated with its image and a list of image to scene point correspondences is built:

$$\vec{x}_i = P\vec{X}_i \quad (3.18)$$

Using the condition that the vector cross product $\vec{x}_i \times P\vec{X}_i = 0$, we can obtain two independent equations for each point correspondence using two rows of the matrix relationship below.

$$\begin{bmatrix} 0^T & -z_i\vec{X}_i^T & y_i\vec{X}_i^T \\ z_i\vec{X}_i^T & 0^T & -x_i\vec{X}_i^T \\ -y_i\vec{X}_i^T & x_i\vec{X}_i^T & 0^T \end{bmatrix} \begin{bmatrix} P_1^T \\ P_2^T \\ P_3^T \end{bmatrix} = 0 \quad (3.19)$$

Here the P_i^T terms are the 4-vector rows of the projection matrix, the $[x_i, y_i, z_i]$ are the components of the image point and the X_i^T terms are the homogeneous coordinates of the scene point represented as the row vector $X_i^T = [X_i, Y_i, Z_i, 1]$. At least six point correspondences are required to solve the linear system of equations $Ap = 0$ where A is the $2nx12$ matrix of independent rows of equation 3.19. The solution for P is unique to within a multiplicative scalar, provided that the rank of the coefficient matrix is 11.

Obtaining a solution for the elements of the projection matrix P does not result in a matrix which distinguishes the intrinsic and extrinsic parameters. Decomposing the resulting projection matrix into the form of equation 3.17 can be difficult. Luckily, it is possible to perform a restricted camera estimation subject to the constraints imposed by the values expected for the intrinsic parameters. For a camera with known intrinsic parameters, the relative pose between the camera image and any other coordinate system may be determined by a non-linear least squares process which aligns multiple line of sight measurements.

1. The Vis-Nav Solution

The Vis-Nav Sensor has seen much use in proximity operations of autonomous systems. Using an array of uniquely identifiable beacons, the problem of determining the relative pose of a camera is solved by determining the location of the camera origin in the beacon coordinate system and the elements of the rotation matrix between the camera coordinate system and beacon coordinate system by non-linear least squares. Beginning with a measurement model common in attitude estimation:

$$\hat{b}_i = A\hat{r}_i \quad (3.20)$$

The body fixed measurements b_i are the image points found by determining the centroids of each individual beacon. Using the measured image space coordinates of the centroid and the known focal length, the line of sight vector to the beacon in the camera coordinate system can be expressed by:

$$\hat{b}_i \equiv \frac{1}{\sqrt{f^2 + x_i^2 + y_i^2}} \begin{bmatrix} -x_i \\ -y_i \\ f \end{bmatrix} \quad (3.21)$$

Likewise, the line of sight to each individual beacon may be modeled by:

$$\hat{r}_i \equiv \frac{1}{\sqrt{(X_i - X_c)^2 + (Y_i - Y_c)^2 + (Z_i - Z_c)^2}} \begin{bmatrix} X_i - X_c \\ Y_i - Y_c \\ Z_i - Z_c \end{bmatrix} \quad (3.22)$$

The least squares estimates for the coordinates of the camera center, and the camera orientation parameters are found by using the Gaussian Least Squares Differential Correction (GLSDC) which seeks to minimize the cost function $J = \frac{1}{2}\Delta b_i^T W \Delta b_i$.

Here, $\Delta b_i = \hat{b}_i - \tilde{b}_i$ where $\tilde{b}_i = f(x)$ is the predicted measurement given the evaluation of the current estimate of the state variables $x = [X_c, \sigma_c]^T$. Here $X_c = [X_c Y_c Z_c]^T$. The attitude state variables σ_c are typically chosen to be the Modified Rodriguez Parameters (MRP's) which parameterize the direction cosine matrix A as shown in equation 3.23:

$$[A(\sigma_c)] = [I_{3 \times 3}] + \frac{8[\sigma_c \times] - 4(1 - \sigma_c^T \sigma_c)[\sigma_c \times]}{(1 + \sigma_c^T \sigma_c)^2} \quad (3.23)$$

Starting with the initial guess that the camera frame is aligned with the beacon frame the differential correction can be written as a function of the Jacobian of the measurement model, a weighting matrix based on the inverse of the covariance of each measurement, and the measurement error Δb_i :

$$\Delta x = (H^T W H)^{-1} H^T W \Delta b_i \quad (3.24)$$

The Jacobian matrix may be computed by $H = [\frac{\partial f}{\partial X_c} \frac{\partial f}{\partial \sigma_c}]$ where the vector $f(x)$ is given by $[A(\sigma_i)] \frac{[X_i - X_c]}{\|X_i - X_c\|}$ and the partial derivatives are computed by:

$$\frac{\partial f}{\partial X_i} = \frac{1}{\|X_i - X_c\|} [A(\sigma_i)] (r_i r_i^T - I_{3 \times 3}) \quad (3.25)$$

$$\frac{\partial f}{\partial \sigma_i} = \frac{4}{(1 + \sigma_i^T \sigma_i)^2} [f \times] ((1 - \sigma_i^T \sigma_i) [I_{3 \times 3}] - 2[\sigma_i \times] + 2\sigma_i \sigma_i^T) \quad (3.26)$$

Using the initial estimate of perfect alignment, the normal equations of 3.24, and evaluating the cost function until convergence results in accurate estimates for the camera pose with respect to a reference coordinate system. In order to determine these parameters for a camera in the reference frame of a Light Detection and Ranging device, the beacons of the above discussion must have specific properties. The approach proposed in this dissertation is to use a set of calibration objects which allow for easy localization of a characteristic point in both the 3D coordinates measured

by the LIDAR and the image space of the camera. One choice of targets are small white styrofoam spheres of known radius. Styrofoam has excellent diffuse reflection across a wide color spectrum. In order to manufacture accurate point to point correspondences as in the methods introduced above, the center of the best fit sphere computed for each object is associated with the line of sight defined by the centroid of the spheres image in the camera. Figure 22 shows the geometry of this type of correspondence, it should be noted that associating the center of the best fit sphere to the image centroid can provide more accurate point to point correspondences than trying to associate a single LIDAR measurement with a location in the camera image. Care should also be taken to ensure that the centroid of the image intensity function is the line of sight through the center of the sphere which can easily be done by locating a light source near the camera.

Determining the correct subset of LIDAR measurements to associate with each calibration object is not trivial. If the LIDAR data is in the form of a list of 3-vectors, methods for determining nearest neighbors can be used to determine groups of points which have a variance in their 3D position comparable to the radius of the calibration sphere. Another, more interactive method is used here. First, a 2D array of LIDAR measurements is computed by converting the range measurements into the form $r(\phi, \theta)$ where θ and ϕ are the azimuth and elevation angles determined from the unit line of sight vector by equations 3.1 and 3.2. The nominal steps in azimuth and elevation $\delta\theta$ and $\delta\phi$ are determined by dividing the field of view dimensions by the number of columns, and the expected number of measurements per column respectively. In the case of the Deltasphere 3000 scanner shown in figure 23 [38], these parameters are provided in the header of the scan file.

An example Return Light Intensity image of a calibration scene is shown in figure 24 alongside the camera image of the same scene. Since the strength of the

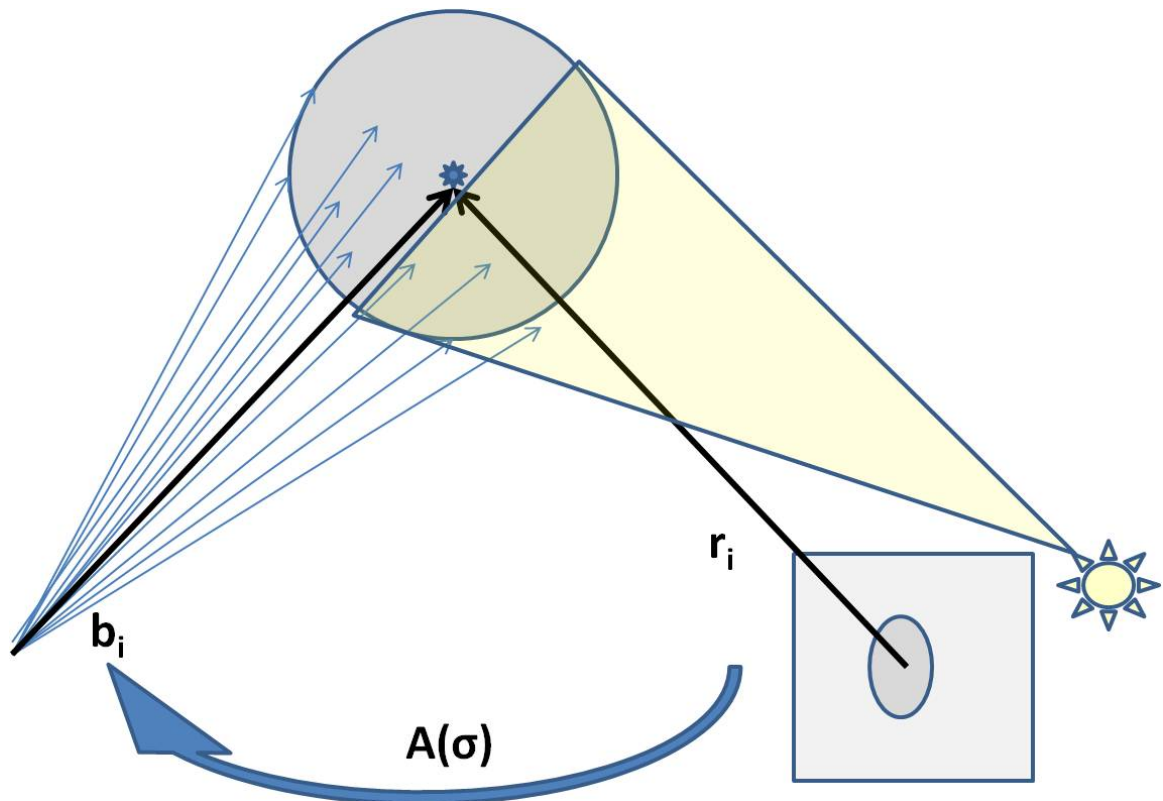


Fig. 22: $2D$ - $3D$ point correspondence by associating the center of a best fit sphere to the centroid of the spheres image.



Fig. 23: The Deltasphere 3000 scanner operates by pulsing a laser against a mirror which rotates along the horizontal axis, the entire apparatus is rotated by a motor along the vertical axis resulting in a raster scan range image.

return light intensity is dependent on the cosine of the incidence angle of a particular LIDAR measurement, the locations of the sphere which obtain a maximum value should indicate lines of sight which are nominally through the center of mass of the sphere.

Manual association of each sphere in the Return Light and Camera images can easily be achieved using a mouse input function such as Matlab's `ginput` function.

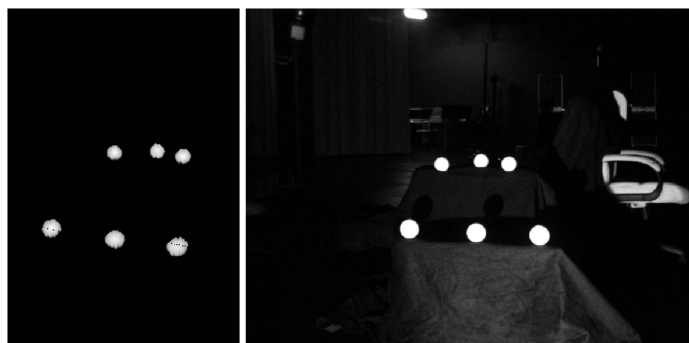


Fig. 24: Left: A Return Light Intensity image of six calibration spheres. Right: The Visible Light image taken by the camera for which a calibration is sought.

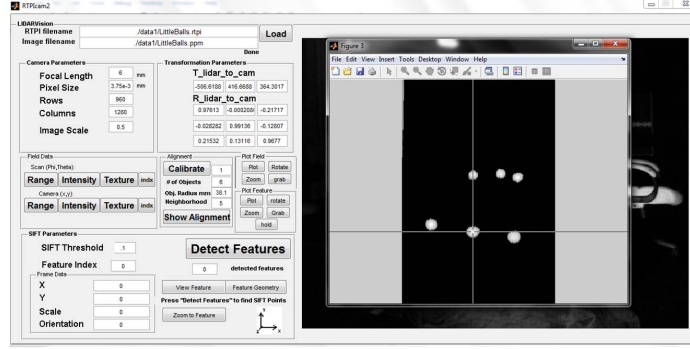


Fig. 25: The object association phase of the calibration process allows the user to manually identify each calibration object in each image and associate a neighborhood with each data set over which the best fit sphere and image centroid is determined.

Figure 25 shows this process implemented in a software routine which requires the user to click the center of the sphere in each image and specify a $2D$ neighborhood over which pixel data will be used in the computation of the $2D$ and $3D$ characteristic points.

Once the $2D$ and $3D$ neighborhoods of data for each object have been associated, the $3D$ points are determined by one of two methods. The first method is to find the center of the best fit sphere of radius equal to the radius of the calibration object. This process can again be solved by GLSDC. In this case, we seek to minimize the cost function $J(x) = \frac{1}{2} \sum_i [\|x_i - \bar{x}\| - r]^2$ where the x_i are the individual $3D$ measurements in the neighborhood, \bar{x} is the estimate of the sphere center, and r is the known radius. This process works well for spheres that are large with respect to the localization error of the measurements. If this method is ineffective, the measurement for which the cosine of the incidence angle attains a maximum value may be chosen as the line of sight direction, and the $3D$ characteristic point may be determined by adding the radius of the calibration sphere to the measurement in the direction of the line of sight.

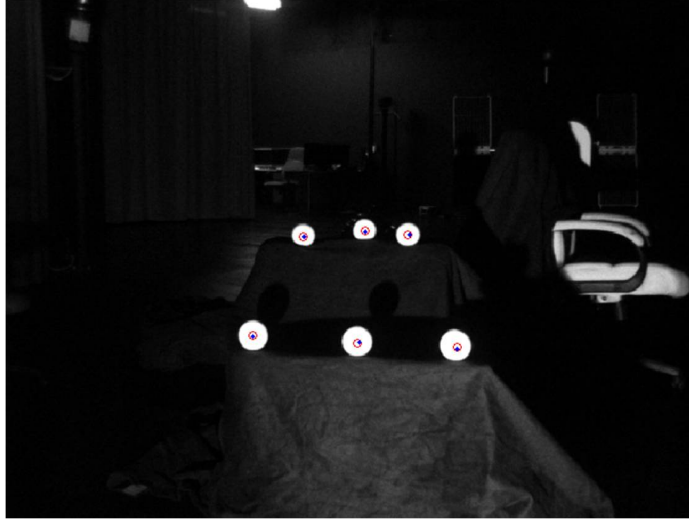


Fig. 26: Alignment of LIDAR and Camera using Calibration spheres and the Vis-Nav approach. The blue asterisks indicate the location of the $2D$ characteristic points, the red circles display the projected locations of the $3D$ -points through the co-linearity equations which result from the calibration.

$$\hat{X}_i = \tilde{X}_i + r\hat{l}_i \quad (3.27)$$

Determining the centroid of the camera image of each sphere may be done by computing the center of mass of each neighborhoods intensity function if care has been taken to be sure that the image of the sphere is under the correct lighting conditions. An alternative method to centroiding is to threshold the image and determine the center of the best fit circle to each 2-d neighborhood. Once the $2D$ and $3D$ characteristic points have been determined, the Vis-Nav solution to the extrinsic calibration problem applies directly. An example of the alignment using calibration spheres and the Vis-Nav calibration approach is shown in figure 26.

Once the extrinsic parameters of the camera are determined with respect to the LIDAR frame a visible light image from the perspective of the LIDAR may be generated by determining an intensity value for each LIDAR point visible to the camera by interpolation. For a camera which is only very slightly displaced with

respect to the LIDAR, nearly all points will project to a unique location in the camera and the process of determining the associated visible light value consists of computing the camera coordinates by the colinearity equations of 3.15 and interpolating the image data over a small neighborhood about that point. A common approach is to compute the Gaussian weighted average of the (for example) local 3×3 neighborhood.

$$VL(\theta, \phi) = \sum_{i=-1:1} \sum_{j=-1:1} G(i+2, j+2) I(\lceil \frac{i_y + p_y}{d_y} \rceil + i, \lceil \frac{i_x + p_x}{d_x} \rceil + j) \quad (3.28)$$

Here $G(i, j)$ is a 3×3 matrix of values sampled from the $2D$ Gaussian kernel. For a camera with a larger displacement, certain scene geometries can cause occlusions to occur. Occlusions occur when more than one point in the $3D$ data set project to the same pixel location in the camera, or when a subset of the points line of sight is blocked (occluded) by other points in the scene. In order to correctly apply the visible light texture to the data set, it is necessary to determine occlusions occur. A simple method is to determine the range of each $3D$ point in the camera frame, and sort the $3D$ data with respect to this value. The image locations and interpolated values for each unique data point are then computed. An array the same size as the camera image is used to indicate if a camera image pixel has been used in a previous interpolation. Each time an interpolation is performed, the associated 3×3 neighborhood of the occupancy image takes the value of the range between the camera and the interpolated data point. This array or occupancy image is queried each time a location is determined for a LIDAR measurement, if it is zero at the pixel the current measurement projects to, the interpolation proceeds. If the value is not zero, then the range value populating the occupancy image is compared to the range of the current data point and if the difference is larger than a threshold, the data point is not interpolated. Figure 27 shows the result of this process on a calibrated range image of

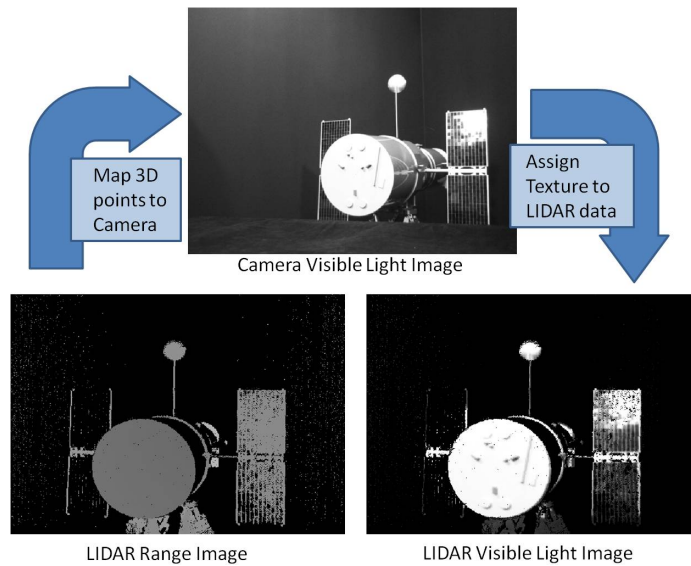


Fig. 27: Alignment of LIDAR and Camera using Calibration spheres and the Vis-Nav approach. The blue asterisks indicate the location of the 2D characteristic points, the red circles display the projected locations of the 3D-points through the co-linearity equations which result from the calibration.

the 1/10th scale Hubble Space Telescope model at the Texas A&M University Land Air and Space Robotics Laboratory.

Data in this form represents the data Multi-Field Multi-Scale Features were designed for. A fusion system consists of a time of flight measurement device, in this case a LIDAR, externally calibrated with a texture measuring device. This process could be repeated with an array of cameras of any type. Nothing prevents this process from being applied to an infra-red or hyper-spectral camera. Increasing the number of available textures will simply increase the number of fields to be searched for features. As is shown in figure 27 the LIDAR system does not guarantee a fully populated image. In many cases pixels in a range image may be empty due to insufficient return light. It is important to determine if this is a local event or if the LIDAR measurement truly missed the object being measured. This problem is addressed by use of a hole filling process which is explained in the next chapter.

CHAPTER IV

SCALE SPACE ANALYSIS OF MULTI-FIELD DATA ON MEASURED MANIFOLDS

Following an accurate extrinsic calibration, the range data sensed by a time of flight scanner and the texture data associated with each sensor is sufficiently co-registered. Traditional approaches to generating a multi-resolution representation of the image data involve building a scale-space. A widely adapted approach to generating a scale-space is to construct a 1-parameter family of signals derived by convolving the original signal with a Gaussian operator. In two dimensions we have:

$$L(x, y, t) = g(x, y, t) * I(x, y) \quad (4.1)$$

The result of a Gaussian convolution is a blurred version of the original image data where each element of $L(x, y, t)$ at (x, y) is a Gaussian weighted average of its (x, y) neighbors where t is the scale parameter or variance of the Gaussian operator:

$$g(x, y, t) = \frac{1}{2\pi t} e^{-\frac{(x^2+y^2)}{2t}} \quad (4.2)$$

An example of a scale space was shown in chapter 2, the Gaussian pyramid is formed by successively convolving an image with a Gaussian (with $t = \sqrt{2}$, for this special case). In a traditional scale-space representation the re-sampling process of the Gaussian pyramid is not performed. The method proposed in this dissertation will also preserve all original measurements and therefore will represent a traditional scale space method. A key difficulty in directly applying Gaussian convolution to a range image involves the meaning of image elements containing zero values. In an intensity image, this can mean that an element did not collect any photons, and the zero value

is considered part of the image texture. In a range image however, this can have significant undesired consequences when elements containing a zero value are involved in an average of geometric measurements. A range measurement is based on the round trip time of flight to a scene point. A successful range measurement is reported when a sufficient amount of energy returns to the photo-detector. If the amount of energy collected is less than a pre-specified threshold over a delay associated with the maximum expected measurable range, the range measurement is recorded as a failure and a zero pixel is reported. Physically, this can occur for many reasons. If the maximum time of flight is exceeded, it can mean that no scene point is close enough to cause a measurement, or that the projected energy was reflected away from the receiver which occurs often with high glancing angles and non-Lambertian surfaces. Consider the range image of the asteroid model shown in figure 28. Smoothing this data set by applying a Gaussian operation as in image processing results in the zero element pixels causing significant false gradients in the surface which can be seen in the representation at the right. This occurs because the zero-valued pixel is replaced by a Gaussian weighted average of the neighboring range values, and the zero range value which should not be involved in the computation. Smoothing with an operator which does not take into account that neighbors may not be populated with measurements can cause significant distortion artifacts. The most significant distortion occurs at the object boundaries, which can be expected. This affect has been well addressed by Brady and Ponce [39] and requires more involved smoothing techniques like the computational molecules of Terzopolous [40].

In image processing, convolution is implemented by performing a series of array products with a spatial filter, in this case a Gaussian, and summing the result. For each array product whose image members are all non-zero, the following process is equivalent to convolution.

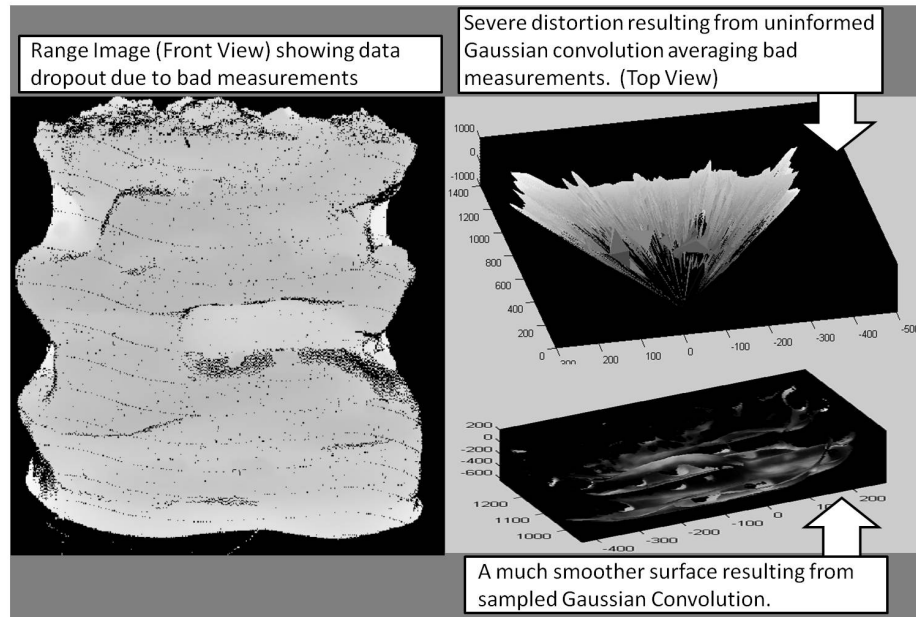


Fig. 28: Left: A range image of a model asteroid with zero pixels representing bad range measurements. Right-Top: A surface model showing the result of Gaussian convolution. Right-Bottom: The same surface textured by Return Light Intensity values achieved using sampled Gaussian Convolutions.

1. Identify for all participating Image elements $f(x, y)$ the (x, y) coordinates containing non-zero members.
2. Set all locations in the Gaussian operator, $G(x, y)$, corresponding to zero valued elements to zero.
3. Re-scale the filter values by $1/\text{sum}(G(x, y))$.
4. Compute the array product and sum the result.

For array products which do contain elements with zero values, this process is an approximation to the linear process of Gaussian convolution. It is similar to the process described in smoothing 3D data by means of the weighted umbrella operator, but the process described above does not use the measured 3D information and just performs a smoothing on the range values. In this case, the weights which are com-

puted for the Gaussian operator are defined assuming unit step distances between the pixels, and the range values are being blurred much the same way as the scalar fields of return light and visible light intensities. This approach is undesirable for a number of reasons. First, blurring the range data as a scalar field is not an effective 3D analysis and does not correctly smooth the noisy LIDAR data. Replacing a range value, which is the magnitude of a measured position vector, by an average of the range magnitudes over an area is not the same as surface evolution, and depending on the way the values are weighted, can cause significant distortions. Second, surface smoothing, and therefore, the smoothing of any associated scalar data is subject to tangential drift when points are not regularly sampled on the object. This drift is caused by poorly selected weights and can be minimized by careful design of the operator. Only for very specific surface types and poses does the condition of consistent object space sampling occur for a particular sensor. It is desired that the surface evolution process applied to the measured point cloud data both apply the diffusion equation to the measurements based on some local assumption of surface properties, and evolve the data to a representation of a progressively smoother surface in such a way that the sampling frequency along the surface is not changed. In chapter 2, it was shown that surface evolution by mean curvature flow contained only displacements in the direction of the underlying surface normal. Mean curvature flow is a surface area minimizing flow, and we will show here that it can be achieved by designing an umbrella operator constructed using the process described above, with modifications which take into account the measured shape of the surface. Recall that a surface evolution process may be described by the expression in equation 4.3.

$$\frac{\partial P}{\partial t} = F(P) \tag{4.3}$$

The expression in equation 4.3 can be thought of as the rate of change in the surface data or the diffusion flow rate cause by applying the process to the points P represented by $F(P)$. In chapter 2, this process was defined for mean curvature flow using the discrete equation in 4.5.

$$P_{k+1} = P_k - H\hat{n}\Delta t \quad (4.4)$$

By writing 4.5 in the form of 4.3 we can show that the diffusion flow rate is given by the curvature normal, $H\hat{n}$.

$$\frac{P_{k+1} - P_k}{\Delta t} = -H\hat{n} \quad (4.5)$$

The curvature normal, is defined in differential geometry another way:

$$H\hat{n} = \frac{1}{2} \frac{\nabla A}{A} = \text{div}(\hat{n}) \quad (4.6)$$

It is shown in [41] that the computation of the curvature normal for a surface mesh can be achieved by the following equation:

$$H\hat{n} = -\frac{1}{4A} \sum_{j=N_1} (\cot\alpha_i + \cot\beta_j)(x_j - x_i) \quad (4.7)$$

The expression in equation 4.7 is shown here to make two points. First, the operator for a mesh which implements mean curvature flow as a surface evolution process is in the form of an umbrella operator. The umbrella operator in its general form was given in equation 2.34 and evolves each point in a surface by defining its future location to be the result of an averaging operation on its original position and the positions of its neighbors. The umbrella operator was first presented here to introduce surface evolution by Laplacian smoothing. Laplacian smoothing however, can have undesired properties like tangential drift. The main point here is that with the correct selection

of weights, the Laplacian smoothing process results in the mean curvature flow and correctly evolves the signal without aliasing the spatial position of the points as a function of scale. The second point made here is that the expression in 4.7 is also the derived expression for a mesh for an operator known as the Laplace beltrami operator [41, 42]. The Laplace-Beltrami Operator is the generalization of the Laplacian operator for Riemannian manifolds. If we let f denote a twice differentiable Riemannian manifold, we may express the Laplace-Beltrami Operator as:

$$\Delta f = \text{Div}(\nabla f) \quad (4.8)$$

The Laplace-Beltrami operator is most commonly implemented using a specifically weighted umbrella operator:

$$\Delta f(P) = \frac{1}{\sum_{i \in N_1} w_i} \sum_{i \in N_1} w_i (P - P_i) \quad (4.9)$$

Here the w_i are the weights associated with the distance between each point $(P - P_i)$. This expression is typically only evaluated for the 1-ring neighbors N_1 , but it can be shown that if one uses the geodesic distance to define the weighting, the equation 4.9, can be extended past the 1-ring neighbors. In [41], the authors introduce a normalized curvature normal operator and express the umbrella operator to be:

$$H\hat{n} = -\frac{1}{\sum_{j \in N_1} (\cot\alpha_i + \cot\beta_j)} \sum_{j \in N_1} (\cot\alpha_i + \cot\beta_j)(x_j - x_i) \quad (4.10)$$

This approximation to the Laplace-Beltrami operator is an extension of the so-called graph Laplacians which use weights to convey the adjacency information of surface points. It is well known in the literature, [43] that a point in a point cloud or mesh can be evolved by mean curvature flow of the surface from which it was sampled by applying the Laplace-Beltrami operator.

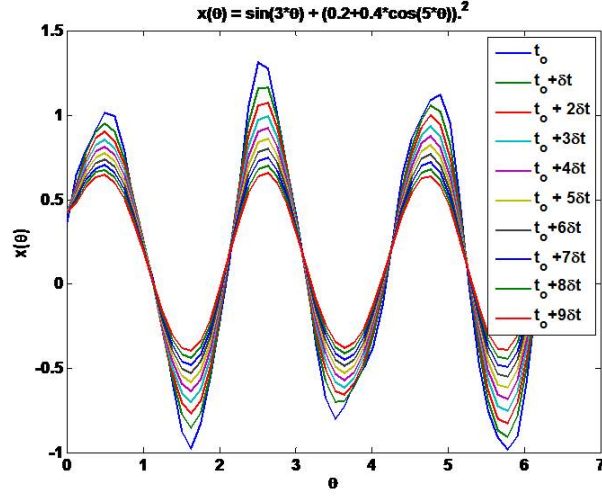


Fig. 29: A uniformly sampled 1D signal smoothed by the finite difference approximation to the Laplacian

To illustrate the importance of selecting the correct weights when applying an umbrella operator to build a scale space of data, consider the approximation of the Laplacian on a 1-D curve. We seek to build a 1-D scale space which was shown by Witkin to provide extrema in a nesting tree. This example addresses figures 29 to 31 where the original function is the periodic signal with largest amplitude and successive applications of a Laplacian operator result in progressively lower amplitude signals. Consider the signal $x(\theta) = \sin(3 * \theta) + (0.2 + 0.4\cos(5\theta))^2$ on the interval $[0 : 2\pi]$ sampled in steps of $\pi/25$. The 1-D discrete approximation to the Laplacian can be determined by finite differences to be $L(x_i(t)) = \frac{1}{2}(x_{i-1} - x_i) + \frac{1}{2}(x_{i+1} - x_i)$ and the signal may be evolved by $x_i(t + 1) = x_i(t) + L(x_i(t))$. The results after each of 10 iterations are shown in figure 29.

If the sampling in θ is allowed to vary instead of being consistently stepped, successive convolutions do not correctly smooth the data. If we corrupt the theta values by $\theta = \theta + \frac{d\theta}{4}N(0,1)$, the result of the same successive smoothing process using the standard Laplacian operator is shown in figure 30. Notice the smoothing artifacts

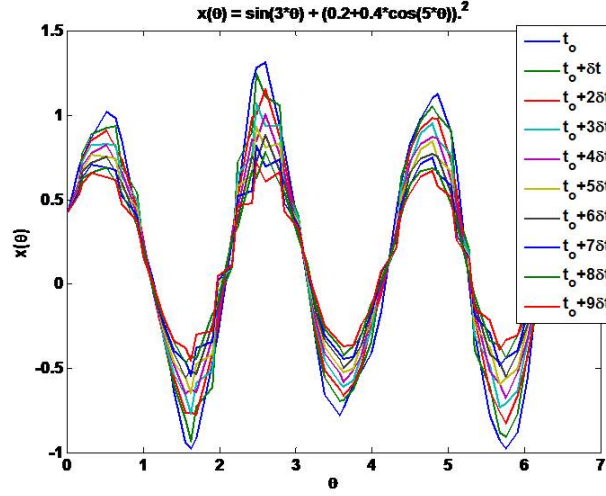


Fig. 30: A non-uniformly sampled 1D signal smoothed by the finite difference approximation to the Laplacian

caused by this ill-informed smoothing process which make it difficult to distinguish each individually smoothed version of the original signal which should lie neatly one within the other.

The reason for the poor smoothing performance is the equal weighting assigned to each nearest neighbor regardless of the actual distance $\theta - \theta_i$. When the Laplacian operator is adapted to take into account the actual inter-point distance we have the following approximation:

$$L(\theta_i) = \frac{1}{w^- + w^+} (w^- x(\theta_{i-1}) + w^+ x(\theta_{i+1})) - \theta_i \quad (4.11)$$

where $w^- = \frac{1}{\theta_i - \theta_{i-1}}$ and $w^+ = \frac{1}{\theta_{i+1} - \theta_i}$ are the weights associated with the inverse of the inter-point distances between the first nearest neighbor in the positive and negative directions. We note here that the first nearest neighbors represent the 1D analog of the 1-ring neighbors on a mesh, or the 8 nearest neighbors to a pixel in an image, and that this approximation to the Laplacian is a weighted umbrella operator. The result of applying the distance corrected Laplacian operator to the same non-uniform

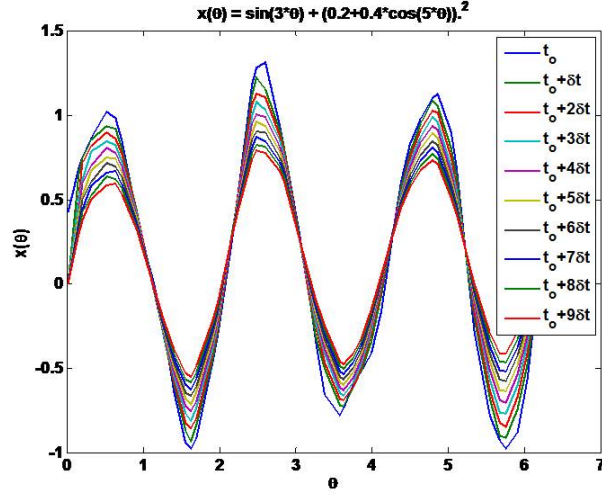


Fig. 31: A non-uniformly sampled 1D signal smoothed by the distance corrected finite difference approximation to the Laplacian

sampling of our example is shown in figure 31, notice here that the desired nested structure is preserved even though the inter-sample distance is not constant.

The Laplace-Beltrami operator works in very much the same way as the 1D example for points sampled on a surface. Defining an umbrella operator with weights defined as in 4.10 evolves a surface mesh by mean curvature flow, as all point displacements are only in the direction of the surface normal. The method developed in this dissertation evolves a measured surface by mean curvature flow, however does not require that the measured points be meshed. In order to evolve the measured point cloud by mean curvature flow, it is necessary to find a meshless approximation to the Laplace-Beltrami operator. Fortunately, it has been demonstrated in [42] that a good approximation to such an operator is an umbrella operator with Gaussian weights defined by the geodesic distance.

In Euclidian space, the Laplace-Beltrami operator takes the familiar form of the standard Laplacian:

$$\Delta f = - \sum_i \frac{\partial^2 f}{\partial x_i^2} \quad (4.12)$$

On a manifold however we may define the Laplace-Beltrami operator as a function of samples from the manifold and the Riemannian metric tensor G :

$$\Delta f = \frac{1}{\sqrt{\det(G)}} \sum_j \partial_j \left(\sqrt{\det(G)} \sum_i G^{-1} \partial_i f \right) \quad (4.13)$$

which reduces to the expression of equation 4.12 in the case of a plane, on which the metric tensor G is the identity matrix. The standard approximation to equation 4.13 for a point cloud is given by the *graph Laplacian* which defines weights between a point in a point cloud and its neighbors to be $w_{ij} = e^{-\frac{\|x_i - x_j\|^2}{4t}}$ where $\|x_i - x_j\|^2$ represents the geodesic distance between the i^{th} and j^{th} point. The corresponding graph Laplacian matrix L_n^t is given in equation 4.14.

$$(L_n^t)_{ij} = \begin{cases} -w_{ij}, & i \neq j \\ \sum_k w_{ik}, & i = j \end{cases} \quad (4.14)$$

This matrix indicates the influence of data point i on neighbor j and along the diagonal, the weights are summed to account for the contribution of points in a neighborhood. In this way, it is possible to apply the graph Laplacian to functions defined on the points sampled from a manifold by computing:

$$L_n^t f(x) = f(x_i) \sum_j e^{-\frac{\|x_i - x_j\|^2}{4t}} - \sum_j f(x_j) e^{-\frac{\|x_i - x_j\|^2}{4t}} \quad (4.15)$$

Here $f(x)$ represents a scalar function defined at the points x . This fact will be useful when evolving surface data with associated textures, since the surface corrected application of the diffusion equation can also be used to diffuse associated texture. The graph Laplacian has been shown to converge to the Laplace-Beltrami operator

for point sampled surfaces. This theorem as well as the rigorous connection of the Laplace-Beltrami operator to the Gaussian in geodesic coordinates is provided in [42].

We may begin to understand the connection between the Laplace-Beltrami operator and the geodesic Gaussian by first observing its connection to the heat diffusion equation. As is well known in the image processing literature [2, 3, 27, 42], the linear isotropic diffusion equation can be determined by convolution with a Laplacian operator or closely approximated by computing the Difference of Gaussians. By extension we should expect the Laplace-Beltrami operator, which is a generalization of the Laplacian to a surface, to also be related to the heat equation which describes the flow of heat on a manifold. A sufficiently differentiable function $f(x, t)$ satisfies the heat equation if:

$$\frac{\partial}{\partial t}f(x, t) + \Delta f(x, t) = 0 \quad (4.16)$$

The solution to a differential equation in this form is a member of the semi-group of heat operators \mathbf{H}^t and given an initial distribution $f(x, 0)$, the distribution at time t may be given by $f(x, t) = \mathbf{H}^t f(x, 0)$. It has been proven [42], that solutions take the form of the 1-parameter family of Gaussian functions.

$$H^t(x) = (4\pi t)^{-\frac{k}{2}} e^{-\frac{\|x-x_i\|^2}{4t}} \quad (4.17)$$

We may use the heat equation to approximate a Laplace operator for a given application. We may write:

$$\Delta f(x) = -\frac{\partial}{\partial t}u(x, t)|_{t=0} = -\frac{\partial}{\partial t}H^t f(x)|_{t=0} = \lim_{t \rightarrow 0} \frac{1}{t}(f(x) - H^t f(x)) \quad (4.18)$$

Using the result from the graph Laplacian of equation 4.15 we can obtain an approximation for the limit

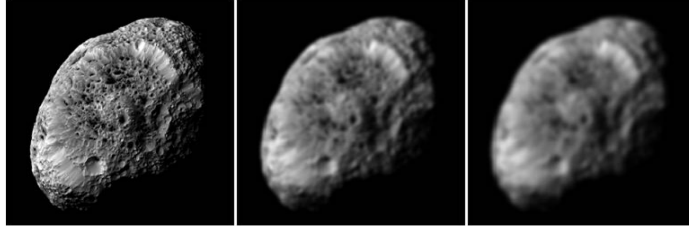


Fig. 32: Asteroid image after 1, 5, and 10 applications of Laplacian smoothing.

$$\Delta f(x) = \frac{1}{t(4\pi t)^{\frac{k}{2}}} L_n^t f(x) \quad (4.19)$$

Much like in the $1 - D$ example if the inter-point distances used in 4.15 account for the distribution between points, the application of a graph Laplacian as an approximation to the Laplace Beltrami operator will evolve a point cloud subject to the mean curvature flow of the underlying surface. Just as the Laplace-Beltrami operator converges to the isotropic Laplacian in the case that the surface is a plane, the surface based distance should converge to a euclidian distance in the case of a plane, and grow as surface curvature increases. This is exactly the behavior of the geodesic distance. Further, the geodesic distance may be approximated by a sum of the measured euclidian distances up to third order [42].

To further illustrate the benefit of extending the Laplacian operator to an approximation of the Laplace-Beltrami operator which takes into account the surface based distances between points, consider the following $2D$ example. An image can be thought of as a data set $(X(u, v), Y(u, v), I(u, v))$ where the coordinates (X, Y) are in consistent steps. First, an image of an asteroid is used as a texture $(I(u, v))$ for a planar surface which is sampled in constant steps. For this ideal sampling structure, the techniques of image processing and surface processing are equivalent, operators constructed under the assumption of constant spacing correctly evolve the data. An example of applying Laplacian smoothing multiple times is shown in figure 32.

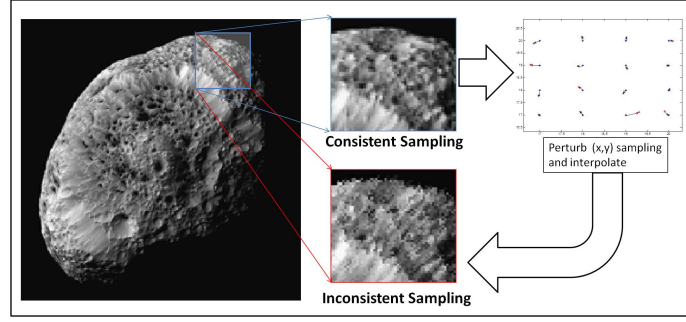


Fig. 33: Noisy Non-uniform $2D$ sampling of an asteroid image simulating the observation of the image surface by a LIDAR.

To simulate the case that this texture data lies on a fronto-parallel plane and is sampled by a LIDAR, the original sample locations are perturbed by an amount less than 0.5 pixels in each direction and the texture data is interpolated from the original image. In this case, the inter-point distance on the $X - Y$ plane is not constant. The difference in the appearance of the image due to inconsistent sampling is illustrated in figure 33, notice the texture is appears corrupted due to an inconsistent sampling being represented in a structured grid.

In this way the original image signal is distorted due to uneven sampling. If the gray values were placed in the continuous locations of the samples, the texture would look like the original image, with some holes due to overlapping and missed pixels. The process of blurring this image benefits from using operators which consider the actual inter-point sampling distance. After all successive blurring processes, the graph Laplacian achieves superior results. Figure 34 shows that the sum of the per pixel error between each blurred image and the original scene is less for the graph Laplacian which correctly weights the contribution of each nearest neighbor.

Consider the form of the textured LIDAR data described in chapter 3. Each “frame” is a $M \times N$ array of measurement sites where M and N are the expected number of measurements in the azimuth and elevation angles given the field of view

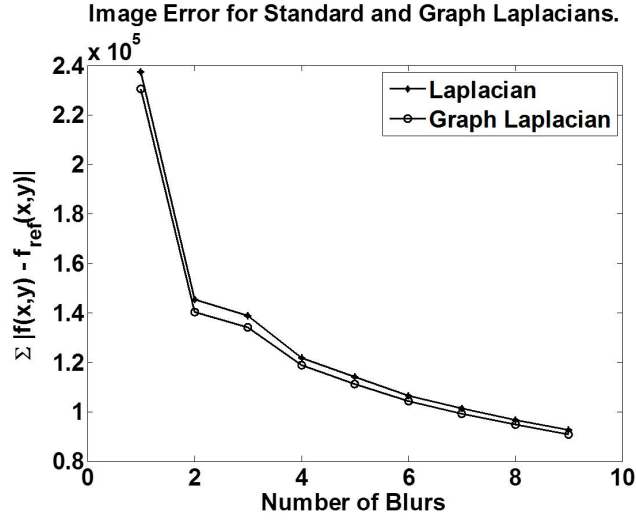


Fig. 34: The sum of the per pixel error between the consecutive Laplacian smoothings of a uniformly sampled signal and the Laplacian and Graph Laplacian smoothing of the non-uniformly resampled signal. The graph Laplacian achieves lower errors over all blurs due to correct nearest neighbor weighting.

of the LIDAR and the angular sampling rates. Each element location in the array can be considered a “pixel” in the sense that it is an image or array element. However associated with each array element are the 3D position of associated LIDAR measurement and associated textures. We can represent the data being smoothed in the 2D example, and the data contained in a textured LIDAR array of measurements by the triple $[\mathbf{X}(\mathbf{u}, \mathbf{v}), RLI(u, v), VL(u, v)]$. In the 2D example, the spatial coordinates of the measurements were constrained to lie on the $X - Y$ plane. In the general case, the vectors $\mathbf{X}(\mathbf{u}, \mathbf{v})$ are the cartesian coordinates measured by the LIDAR. Each array element is nominally $d\phi$ by $d\theta$, but for points measured by an equal angle sampling device like a 2-axis scanning LIDAR the distance between two arbitrary points on a measured surface is not characterized by the angle between them. In fact the fronto-parallel configuration is the only case where equal angle sampling of a plane is isotropic, and the only geometric structure where this is locally true for all measurement directions in the field of view is a spherical shell displaced from the LIDAR at a

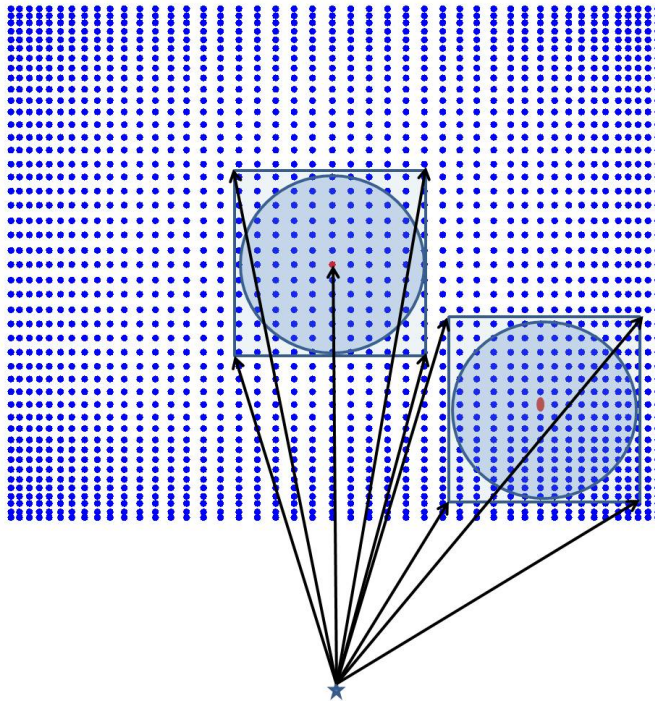


Fig. 35: Equal angle sampling of a fronto-parallel plane by a two axis scanning LIDAR. The central point, whose measurement direction is orthogonal to the plane observes a symmetric distribution in its neighbors while every other point for the same operator observes an asymmetric distribution.

distance equal to its radius of curvature. Figure 35 shows the case of a fronto-parallel plane measured by a LIDAR. Notice that the only symmetric distribution of points can be found in a circular region about the bore sight.

Directly applying a Gaussian operator as in the scale space methods of image processing, or applying an isotropic operator such as the Laplacian to this data will incorrectly weight the nearest neighbors in all cases but the central point. In the case of a spherical shell, the equal angle sampling will result in consistent sampling of the object space surface. However, a weighting defined on a distance based on the subtended angle between a point and its neighbors is proportional to the Euclidian $3D$ distance for the case that the range to the two points is equal, but as the ratio between the ranges grows the weighting based on the angle between a point and

its neighbors ceases to accurately reflect the actual distance between two points and tends to weight too highly points that are far away. Finally the Euclidian $3D$ distance is only a distance along a manifold if that manifold is planar. This fact is not of any consequence for operators that consider only the 1-ring neighbors, but for operators that are extended to multiple rings, the outer rings will be weighted by a chordal distance instead of a cumulative distance along the surface.

For an array of measurements representing a point cloud we do not have any assumptions of connectivity, aside from assuming that the line between an arbitrary point in the point cloud and its eight neighbors in the array is an admissible path along the surface. Figure 36 illustrates this array and displays as a graph the directions of admissible connectivity. Also explained in the same figure, is the 7×7 operator used to define a graph Laplacian in the form of an umbrella operator which uses a meshless approximation of geodesic distance for each point in the operator array. Being a 7×7 array, the neighborhood of a fully populated operator, that is when the operator is being applied to area of the measurement array which has all non-zero elements, is analogous to a 3-ring neighborhood. It is important to consider that the knowledge of the measurement order with respect to the scanning process is used to enforce the previous statement. There are certainly some configurations where a meshing process, without the knowledge of the true scanning order might change the order of the measurements.

Computing the geodesic distance for each element in the operator proceeds as follows:

1. 1-ring Neighbors: The geodesic distance is the Euclidian $3D$ distance $\|x_i - x_j\|^2$
2. 2-ring Neighbors: Compute the sum of Euclidian $3D$ distances over the $n \leq 3$ admissible paths to the 2-ring neighbor. Admissible paths to 2-ring neighbors

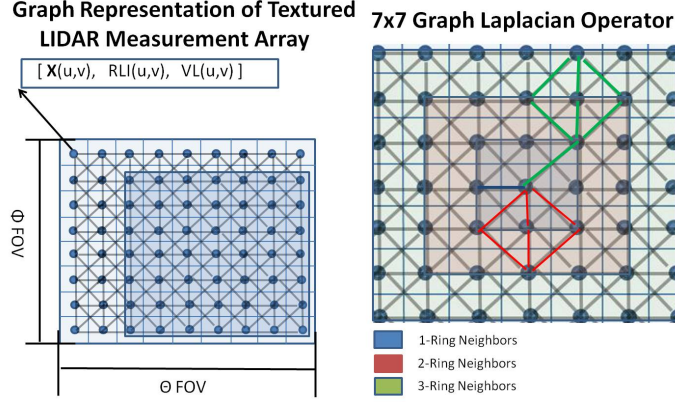


Fig. 36: Left: Graphical representation of the textured LIDAR measurement array. Lines indicate assumed admissible paths along the surface. Right: A 7x7 graph Laplacian Operator.

take only 2 steps between nodes to reach the neighbor.

3. 3-ring Neighbors: Given that the minimum path distance is known to the 2-ring neighbors, determine the minimum distance between the $n \leq 3$ admissible paths from the nearest 2-ring neighbors to each 3-ring neighbor.

The weights used to populate the operator are computed from the standard Gaussian equation using the computed geodesic distances.

Consider the form for the graph Laplacian operator given in equation 4.15. For a fully populated operator the summation in the first term goes to 1. We obtain the following approximation to a Laplacian on the surface:

$$L_n^t f(x) = f(x_i) - \sum f(x_j) e^{\frac{\|x_i - x_j\|^2}{4t}} \quad (4.20)$$

This approximation to the Laplacian is in the form of a Difference of Gaussian operator which was the same approximation used in the Scale Invariant Feature Transform of Lowe [2]. As will be shown in the next section, the application of the heat equation to a manifold may be executed in the same way as the image diffusion used in SIFT.

The approximation of the scale normalized Laplacian achieved by computing the difference of successive Gaussian convolutions may be extended to manifolds measured by LIDAR systems by defining the distance used in the Gaussian weighting to be an arc length along the surface. The generalization of applying the diffusion equation to a point sampled surfaces requires an operator which when applied to the surface data evolves 3D data by mean curvature flow. This operator is implemented in the form of an umbrella operator with Gaussian weights defined by the geodesic distance. Just as the diffusion equation is implemented in the SIFT algorithm by computing a Difference of Gaussian scale space, the diffusion equation can be applied to measured manifolds by computing a Difference of Geodesic Gaussian scale space.

A. Geodesic Gaussian Convolution for Simultaneous Surface Evolution and Texture Analysis

Consider the 2D example given in the previous section. Each measurement was made on the plane, however the inter-point distance was not restricted to be multiples of some constant step. This configuration is not unlike how a scanning LIDAR measures a fronto-parallel surface. The main difference is that many scanning LIDAR systems tend to sample in equal angle increments. A scanning LIDAR provides an array of range values and either a unit vector direction or the azimuth and elevation angles. The Deltasphere 3000 scanner used in this dissertation provides a list of range values with the measured azimuth and elevation angles. Also provided in the header of a LIDAR “frame” of measurements are the minimum and maximum values for the azimuth and elevation angles as well as a nominal number of measurements in each row and column. The following equations are used to derive a step $d\phi$, $d\theta$ used to build the measurement array.

$$d\phi = \frac{\phi_{max} - \phi_{min}}{n_{rows}} \quad (4.21)$$

$$d\theta = \frac{\theta_{max} - \theta_{min}}{n_{cols}} \quad (4.22)$$

While the Deltasphere LIDAR utilizes uniform (ϕ, θ) motions, other sensors may have more complicated foveal scan patterns that are highly non-uniform in (ϕ, θ) . Modifications to the methods of this dissertation will be needed to accommodate such highly non-uniform range measurements. It is expected however that a non-linear map can be defined between the complex scan patterns and $2D$ the array representation used in this dissertation.

The first step in performing a scale space analysis on the Deltasphere LIDAR measurements is to construct the measurement array from the list of LIDAR measurements in the same form as explained in chapter 3. The measured ϕ and θ values for each measurement are mapped to the discrete (u, v) coordinates of the measurement array defined by:

$$u = \lceil n_{cols} - \frac{\phi - \phi_{min}}{d\phi} \rceil \quad (4.23)$$

$$v = \lceil n_{rows} - \frac{\theta - \theta_{min}}{d\theta} \rceil \quad (4.24)$$

This mapping also accounts for the scanning directions of the LIDAR system. Once the (u, v) coordinates of each measurement is determined, the images $r(u, v)$, $\theta(u, v)$, and $\phi(u, v)$ as well as the associated textures of the return light intensity $RLI(u, v)$ are constructed. The final texture value is determined by projecting the measured $3D$ point using the colinearity equations of chapter 3 to the image plane of the camera and computing the associated texture $VL(u, v)$. Once this process is complete the two texture fields and the $3D$ measurements are able to be analyzed like the planar

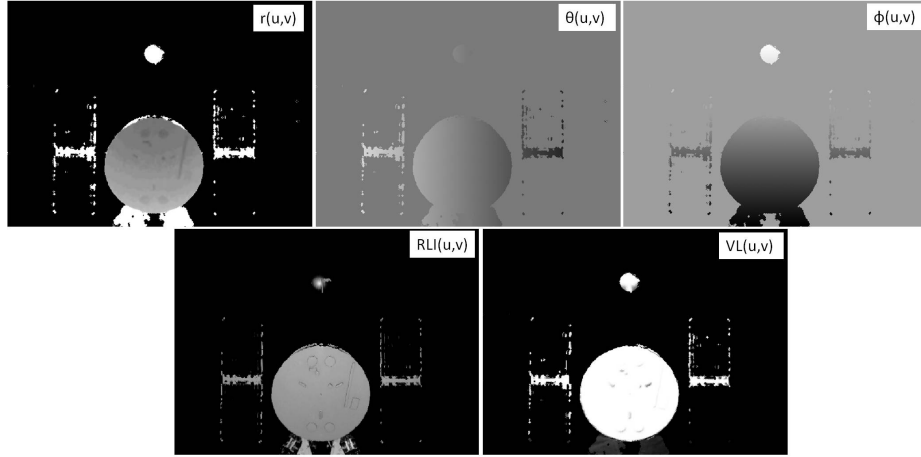


Fig. 37: Measurement array values of range, azimuth angle, elevation angle, return light, and visible light textures.

example. Figure 37 shows an example data set representing a textured scan of the HST model used for proximity operations experiments at the Land Air and Space Robotics laboratory at Texas A& M university.

Generalizing the scale space analysis to surface data and associated texture fields requires the geodesic Gaussian operator as demonstrated in the last section. In contrast to the Euclidian distance, the geodesic distance is associated with an arc length along the measured manifold. The same operator used to analyze the planar example can apply directly to the data shown in figure 37. Applying the geodesic Gaussian operator in order to evolve the surface data by mean curvature flow requires that the surface be parameterized as $\mathbf{X}(\mathbf{u}, \mathbf{v}) = [x(u, v), y(u, v), z(u, v)]^T$ and the operator be applied to each coordinate individually. For the two-axis scanning LIDAR model, Besl and Jain have shown in [14] that the cartesian coordinates are provided by:

$$x(u, v) = \frac{r(u, v) \tan(\theta(u, v))}{(1 + \tan^2(\theta(u, v)) + \tan^2(\phi(u, v)))^{0.5}} \quad (4.25)$$

$$y(u, v) = \frac{r(u, v) \tan(\phi(u, v))}{(1 + \tan^2(\theta(u, v)) + \tan^2(\phi(u, v)))^{0.5}} \quad (4.26)$$

$$z(u, v) = \frac{r(u, v)}{(1 + \tan^2(\theta(u, v))^2 + \tan^2(\phi(u, v)))^{0.5}} \quad (4.27)$$

The surface data is evolved by the diffusion equation:

$$\frac{\partial}{\partial t} f(\mathbf{X}(\mathbf{u}, \mathbf{v}), t) + \Delta f(\mathbf{X}(\mathbf{u}, \mathbf{v}), t) = 0 \quad (4.28)$$

which is implemented by successive convolutions of the defined graph Laplacian based on Geodesic distances such that the surface evolution process of equation 4.5 shown here as:

$$f(\mathbf{X}(\mathbf{u}, \mathbf{v}), t_{k+1}) = f(\mathbf{X}(\mathbf{u}, \mathbf{v}), t_k) - H\hat{n}\Delta t = f(\mathbf{X}(\mathbf{u}, \mathbf{v}), t_k) - \Delta f(\mathbf{X}(\mathbf{u}, \mathbf{v}), t_k)\Delta t \quad (4.29)$$

Here the operator $\Delta f(\mathbf{X}(\mathbf{u}, \mathbf{v}), t_k)$ is approximated using the form of equation 4.9:

$$\Delta f(\mathbf{X}(\mathbf{u}, \mathbf{v}), t_k) = \frac{1}{\sum_{i \in N_3} w_i} \sum_{i \in N_3} (w_i f(\mathbf{X}(\mathbf{u}, \mathbf{v}), t_k)) - f(\mathbf{X}(\mathbf{u}, \mathbf{v}), t_k) \quad (4.30)$$

where N_3 represents the elements of the 3-ring neighborhood operated on by the 7×7 graph Laplacian and the weighting w_i is the weighing defined by the Gaussian function:

$$w_i(x_i, \sigma) = \frac{1}{2\pi\sigma^2} e^{\frac{-D_{G_i}}{2\sigma^2}} \quad (4.31)$$

The geodesic distance D_{G_i} is used here to determine the Gaussian weighting. This is the necessary step to extend the standard Laplacian smoothing of image processing approximated by a difference of Gaussian convolutions to the 3D mesh evolution process of mean curvature flow by means of the graph Laplacian approximation to the Laplace Beltrami operator. The final parameter used to specify this convolution process is the traditional scale parameter of the Gaussian function. In 2D image

processing, the scale parameter is typically set to fit a desired area under the $2D$ Gaussian curve to the selected window size. Large windows are capable of representing operators with larger scale values. In general, scale space analysis allows a bit of freedom in the selection in the operator radius and the associated scale parameter. A common choice is to define a parameterized relationship between the two values for example, let s_x denote the operator radius and σ be the scale or time parameter. The expression $s = \gamma^2 t$ where γ is a parameter chosen on the interval $[1 : 2]$ is a common method for relating the operator scale parameter to the size of the operator. In [2], Lowe chooses the scale parameter for a 7×7 operator to be defined as $\sigma = 2^{1/k}$ where k is the number of successive convolutions required before the effective scale is twice the original value of σ . Regardless of the definition of σ based on the $2D$ neighborhood of the operator, it is important to make the actual scale parameter of the operator based on the $3D$ neighborhood of the data the operator is being applied to. Consider the case where the operator is being applied to a fully populated 7×7 area in the measurement array, the figure below demonstrates the variability in inter-point distance as a function of range. For a planar surface orthogonal to the line of sight direction, the inter-point distance scales by the range to the center point of the operator. The nominal expected $3D$ inter-point distance is given by the arc length subtended by one pixel at the measured range. The scale parameter is made adaptable to the surface sampling frequency by scaling it by this expected arc length value.

$$\sigma_{3D} = r(u, v) s_x \sigma_{2D} \quad (4.32)$$

where s_x represents the mean subtended angle $s_x = \sqrt{d\theta^2 + d\phi^2}$. In this way the geodesic distance based weighting is normalized with respect to the expected value

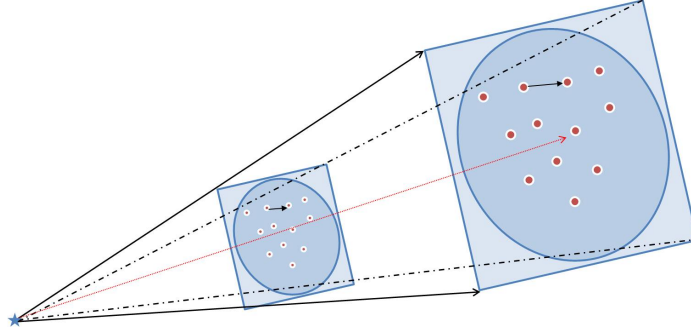


Fig. 38: An example of range based scaling of operator scale.

for the inter-point distance and for the case where the surface is accurately represented locally by a plane orthogonal to the line of sight of the operator origin, the geodesic distance achieves a value of 3σ and converges to the same methods used to define scale for the $2D$ operators.

With the adaptive geodesic distance weighting scheme defined as above, the scale space representation of textured LIDAR data can be built by extending the diffusion equation to the vector $\mathbf{X}(\mathbf{u}, \mathbf{v}) = [X(u, v), Y(u, v), Z(u, v), RLI(u, v), VL(u, v)]^T$ which changes the equation of 4.28 to:

$$\frac{\partial}{\partial t} f(\mathbf{X}(\mathbf{u}, \mathbf{v}), t) + \Delta f(\mathbf{X}(\mathbf{u}, \mathbf{v}), t) = 0 \quad (4.33)$$

In this way the data is evolved by five simultaneous diffusion processes

$$f(\mathbf{X}(\mathbf{u}, \mathbf{v}), t_{k+1}) = f(\mathbf{X}(\mathbf{u}, \mathbf{v}), t_k) - H\hat{n}\Delta t = f(\mathbf{X}(\mathbf{u}, \mathbf{v}), t_k) - \Delta f(\mathbf{X}(\mathbf{u}, \mathbf{v}), t_k)\Delta t \quad (4.34)$$

Once the evolution process has been applied to the textured $3D$ data, the new values for range, azimuth, and elevation angles can be determined by:

$$r(u, v) = \sqrt{x(u, v)^2 + y(u, v)^2 + z(u, v)^2} \quad (4.35)$$

$$\theta(u, v) = \arctan \frac{x(u, v)}{z(u, v)} \quad (4.36)$$

$$\phi(u, v) = \arctan \frac{y(u, v)}{z(u, v)} \quad (4.37)$$

Computing the difference of Gaussian values with respect to the cartesian coordinates results in the components of the curvature normal:

$$H\hat{n}(u, v) = \begin{bmatrix} x(u, v, t_{k+1}) - x(u, v, t_k) \\ y(u, v, t_{k+1}) - y(u, v, t_k) \\ z(u, v, t_{k+1}) - z(u, v, t_k) \end{bmatrix} \quad (4.38)$$

Here the vector magnitude $\|H\hat{n}\|$ is the value for the mean curvature H and is the value used to define the diffusion displacements for the extrema field used to describe geometric features. We define here the fields used for extrema localization as the two texture fields $RLI(u, v)$ and $VL(u, v)$, and the magnitude of the 3D diffusion displacements, or the mean curvature. In this way the features are identified as extrema in the Difference of Geodesic Gaussian scale space in one field describing the extrinsic geometry of the surface and two fields representing scalar functions defined on the surface. The first texture field is weakly correlated to material properties and can be used also as an indicator of measurement confidence. The texture field associated with the camera texture provides a means for detecting motion when it is not geometrically apparent. An example of such a case is a constant curvature surface such as a sphere or plane which is larger than the field of view of the LIDAR. In such a case, without texture information, it is impossible to deduce if the surface has moved with respect to the LIDAR. Figure, 39 shows the result of successive applications of the geodesic Gaussian convolution to the HST spacecraft model. Notice that the surface is successively smoother, while at the same time the texture mapped to the surface has also been blurred. The three fields which represent the diffusion

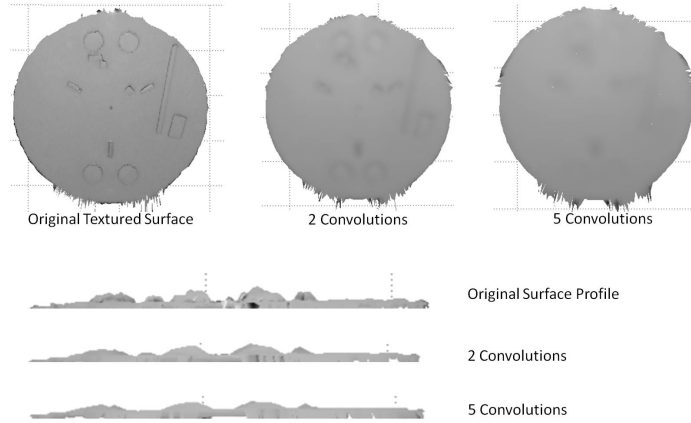


Fig. 39: Successive surface evolution by geodesic Gaussian convolution on the HST model. Top: The back plate of the HST model textured by return light intensity after 0,2, and 5 convolutions. Bottom: The side view of the surface after 0,2,and 5 convolutions.

displacements computed by the Difference of Geodesic Gaussian operator are shown in figure 40 for the same scenario for two and five convolutions.

This process represents a rigorous extension of the difference of Gaussian operator to surface evolution by mean curvature flow for manifolds measured by time of flight devices. The use of the geodesic Gaussian has been demonstrated as a means to simultaneously apply the diffusion equation to the surface data as well as the scalar functions defined on the measured surface. Localizing extrema in the fields associated with the diffusion displacements, either geometric or textural will identify interest points associated with corner like features in either the mean curvature of the surface, or in the texture fields. The next section will explain the process of feature extraction in the multi-field scale space.

B. Feature Extraction in Multi-Field Data

Once the scale space has been constructed, the image stack analogous to the scale space analysis of image data contains the 3D information associated with successively

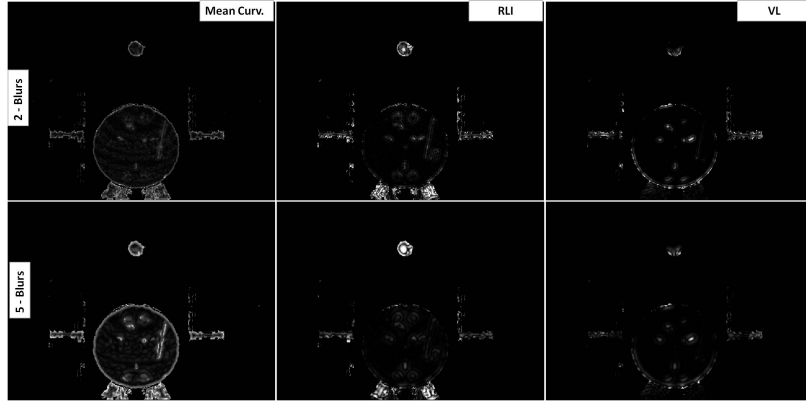


Fig. 40: Image representations of the diffusion displacements computed by the difference of geodesic Gaussians. Top: Mean Curvature, Return Light Intensity, and Visible Light for 2 convolutions. Bottom: The same for 5 convolutions.

smoothed representations of the measured surface as well as the evolution of the scalar texture fields on the surface. In the multi-resolution methods of image processing, a pixel in the difference of Gaussian image stack would be compared to its 26 nearest neighbors, the 8 nearest neighbors of its current level in the stack, and the 9 nearest neighbors in the levels above and below. If this value was either greater than all of these values or less than all of these values by a preset threshold, it was flagged as a tentative location for a scale space extrema. This approach known as non-maxima suppression quickly reduces the image stack to on the order of 10^3 locations in an image which have associated with them their image space locations (x, y) as well as their scale σ . Consider just one level of a scale space. If a pixel is greater than or less than all of its 8 nearest neighbors, and no other level existed, it would be called an extrema. Furthermore consider the case of a LIDAR measuring a symmetric curved surface variation extruded from a planar surface. Figure 41 illustrates this scenario.

In this case, the location associated with a maximum or minimum in mean curvature is displaced along the surface normal direction just like every other point and usually continues to be a maximum until the convolution smooths it out. In our

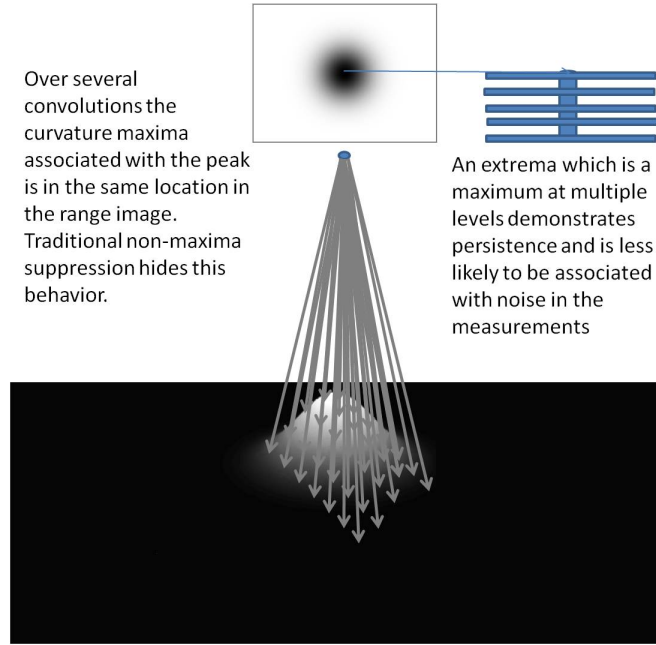


Fig. 41: A persistent geometric feature represented as a surface bump measured by a LIDAR system.

approach to non-maxima suppression, each measurement in the scale space is associated with its immediate neighborhood at its current level in the scale space. The neighborhood is defined however as a function of the current $2D$ scale value. Just as in the Scale Invariant Feature Transform, we define the $2D$ scale parameter to be $\sigma = 2^{\frac{1}{k}}$ where k is the number of successive convolutions desired before the effective scale is double the original value for σ . Therefore, the neighborhood is the local 15×15 region for the first k convolutions since the operator diameter is used to define the radius of the non-maxima suppression neighborhood. For subsequent convolutions, the neighborhood is defined by the effective operator radius for the current octave, so for the next two octaves the neighborhood is 29×29 , and 43×43 respectively. This definition of neighborhood sampling results in detected extrema which are the most significant with respect to the operator domain size. For an array location $DoG(u, v, w, \sigma)$ where w indicates the field of the difference of Gaussian

function, if a position (u, v) is a maximum, or minimum compared to all members of its neighborhood, over multiple consecutive levels of the scale space σ , that location is said to contain a *persistent* feature. Since the particular blurring process is intended to minimize tangential drift of points on an evolved surface, extrema commonly exist in the same location in the range image over several scales. Therefore, persistence is a strong classifier, or criteria for feature confidence. A feature which exists over several scales is associated with a scale range as opposed to a single scale. As long as the scale ranges of features overlap, they are considered admissible candidates for a matching procedure which compares their associated descriptors and perform hypothesis tests to solve the feature correspondence problem. Once a location in the multi-field scale space (u, v, w, σ) or a persistent set of locations $(u, v, w, \sigma_{min} : \sigma_{max})$ is identified as being an interest point, a key-point is generated which contains the object space 3D coordinates, the minimum and maximum 2D and 3D scales, the field the interest point was localized in, and a strength value which indicates a combination of the relative magnitude of the diffusion displacement with respect to its neighbors. The 2D σ values are useful for visualization purposes on the 2D representations of the scale space data. The necessary values for characterizing an interest point in this framework are $[x, y, z, \sigma_{min}^{3D}, \sigma_{max}^{3D}, F_i, S]$. F_i denotes the field from which the keypoint was extracted. An index of 1 indicates the mean curvature values associated with the geometry and the return light and visible light intensities are associated with index values of 2 and 3 respectively. Finally, the strength value, or confidence indicator, is a function based on the corneriness test of [2] and the magnitude of the diffusion displacement associated with the key-point.

$$S = \frac{\det Hess}{trace^2 Hess} (\sigma H \hat{n}(u, v))^2 \quad (4.39)$$

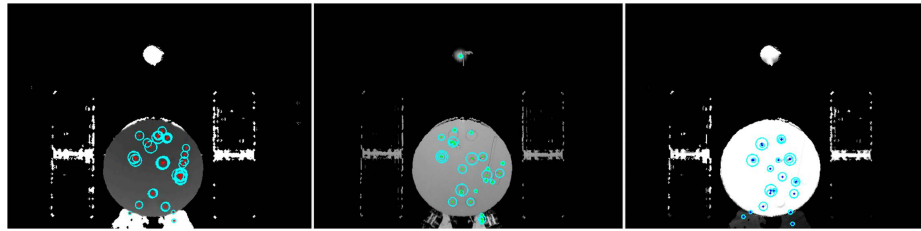


Fig. 42: An example of multi-field multi-scale feature extraction on the HST model. Left: The range field with curvature based features. Middle: The return light intensity field with texture features. Right: The visible light field associated with camera texture. The radius of the identified interest points indicates $2D$ feature scale.

where $Hess$ indicates the local hessian matrix of the difference of Gaussian function, and $H\hat{n}(u, v)$ is the displacement value at the (u, v, w, σ) location in the difference of Gaussian scale space. Only interest points which satisfy the constraint placed on the eigenvalue ratio of the difference of Gaussian values introduced in [2] as well as a constraint on the minimum magnitude diffusion displacement are kept as interest points and fully defined. An example of multi-field multi-scale interest point detection is shown in figure 42 for the HST model.

Each key-point is associated with a descriptor similar to a spin-image [6, 7] which encodes the geometric information of the surface as well as the texture information from each intensity field. The generation of this descriptor is the subject of the next chapter.

CHAPTER V

MULTI-FIELD DIFFUSION FLOW SIGNATURES FOR ROBUST 3D REGISTRATION

Once a set of scale space extrema are detected in each field of the data (e.g. the Curvature Normal field representation of the measured 3D geometry and the two associated scalar fields representing the measured visible light and return light intensity) , the next crucial process is to associate with each location a fingerprint which can be reliably associated with features detected in a subsequent observation of a surface. It is important that whatever descriptor that is used to associate features be robust to relative motion induced transformations and the resulting changes in the sampling of the surface. Furthermore, it is also important for the descriptor to be flexible and allow for good matching performance in the presence of local variations in texture. The descriptor developed by Lowe [2] is robust to monotonic changes in local image intensity, but the descriptor is defined based on uniform sampling in the image space. One reason for the failure of this descriptor to accurately describe 3D shape is the lack of 3D information in the descriptor itself. This is to be expected since it was originally designed for image retrieval and not close range object inspection. The *Spin Image* technique of Johnson [6, 7] is one of the more useful point based descriptors used for registering 3D surfaces. As explained in chapter 2, a spin image descriptor is generated by first defining an oriented point coordinate system at the point of interest. This is done by determining the surface normal direction and then computing the in plane radial distance, α , and the out of plane distance, β , for each point in the data set or neighborhood.

$$\alpha = \sqrt{\|x - p\|^2 - (n \cdot (x - p))^2} \quad (5.1)$$

$$\beta = n \cdot (x - p) \quad (5.2)$$

The performance of the Spin Image technique is sensitive to a number of spin image generation parameters. The first important parameter is the bin size $(\Delta_\alpha, \Delta_\beta)$ which is used to define the storage size of the spin image descriptor. It is common to choose $\Delta_\alpha = \Delta_\beta$ but this is not required. Along with the Spin Image width, W , which is defined in terms of the maximum number of steps $\Delta_\alpha, \Delta_\beta$, the feature support distances $D_{s,\alpha}$, $D_{s,\beta}$ can be defined as in equation 5.3. Once the image size, and bin size parameters are selected, it is easy to visualize the local sampling geometry used to populate the descriptor. Figure 43 illustrates the local cylindrical sampling geometry.

$$D_{s,\alpha} = W\Delta_\alpha \quad (5.3)$$

$$D_{s,\beta} = W\Delta_\beta \quad (5.4)$$

An additional criteria, the support angle A_s defines the minimum allowable deviation in surface normal a neighboring point is allowed to have in order to contribute to the descriptor. It is implemented as a threshold on the inner product of the feature point surface normal and the neighboring points surface normal $\text{acos}(n_a, n_b) \leq A_s$. For all admissible points inside of the support distance from the feature, the discrete locations in the Spin Image array are given by:

$$i = \lfloor \frac{\frac{W}{2} - \beta}{\Delta_\beta} \rfloor \quad (5.5)$$

$$j = \lfloor \frac{\alpha}{\Delta_\alpha} \rfloor \quad (5.6)$$

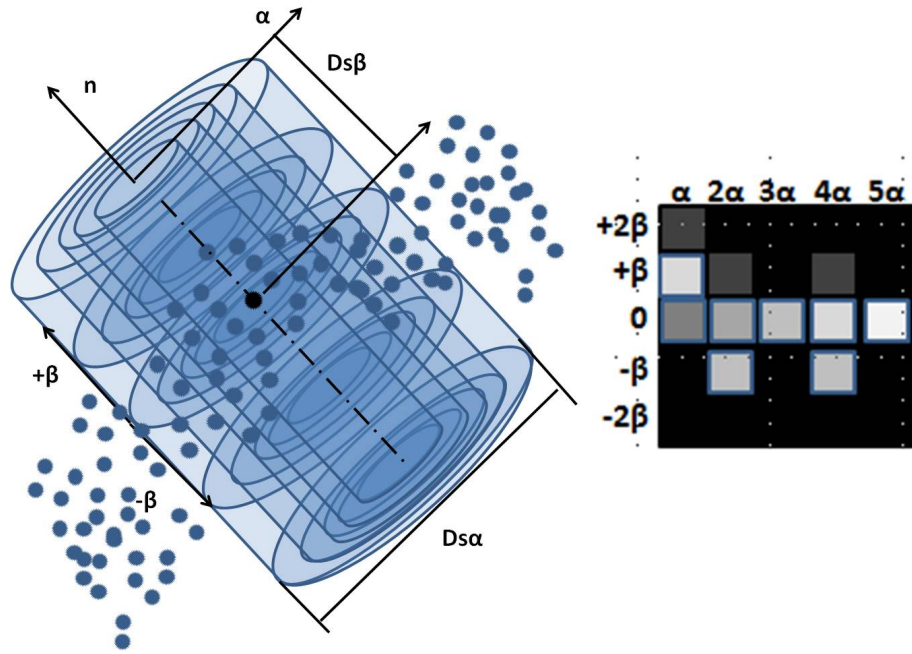


Fig. 43: The local sampling geometry associated with 5 steps Δ_α and ± 2 steps Δ_β

For a simple implementation of the above process, the (i, j) location in the spin image array is incremented by 1 for every point in the neighborhood. In order to account for noise in the point locations, the contribution of each point in the neighborhood is spread out over the local 2×2 element neighborhood in the spin image array by bilinear interpolation. The interpolation weights are determined by:

$$a = \alpha - i\Delta_\alpha \quad (5.7)$$

$$b = \beta - j\Delta_\beta \quad (5.8)$$

Finally, the contributions to each of the 4 local array elements are given by:

$$SI(i, j) = (1 - a)(1 - b) \quad (5.9)$$

$$SI(i, j + 1) = (1 - a)(b) \quad (5.10)$$

$$SI(i + 1, j) = (a)(1 - b) \quad (5.11)$$

$$SI(i + 1, j + 1) = ab \quad (5.12)$$

Johnson [6] recommends selecting the bin size to be between 1 and 2 times the size of the mesh resolution. A significant effort is expended regularizing the mesh built for a set of 3D measurements before the spin image technique is applied. Given that the inter-point distance has been regularized, the spin image representation is consistent for every point in the mesh. This process is an important one because it ensures that the sampling density in a given area about an arbitrary point on a surface mesh is the same, giving the descriptors the best chance of being populated in a similar way. Once spin images are generated for every point in a set of measurements, or for extracted scale space extrema locations, they can be compared by means of normalized cross correlation. The linear correlation coefficient between spin image A and the spin image of a candidate correspondence B is given by:

$$R(A, B) = \frac{N \sum A_i B_i - \sum A_i \sum B_i}{\sqrt{(N \sum A_i^2 - (\sum A_i)^2)(N \sum B_i^2 - (\sum B_i)^2)}} \quad (5.13)$$

Here, N is the number of pixels in the spin image and the index i is a single index which references all pixels in the spin image from 1 to N . The output of this similarity measure is a value between -1 and 1 which indicates if the spin images are completely anti-correlated, or perfectly correlated respectively. Since the spin image was introduced in 1997, there have been several variants of the technique including mult-scale, and textured spin images [8, 9]. Brusco et. al. [8] in applying their textured spin image technique to a globe, or textured sphere demonstrated that for some objects, geometric information alone can be insufficient for registration. The spin image was enhanced with the texture data by increasing the dimension of the spin image to include a coarse sampling of the gray scale or color data associated with each point. For simple geometric shapes, such as a plane or sphere, if the surface is

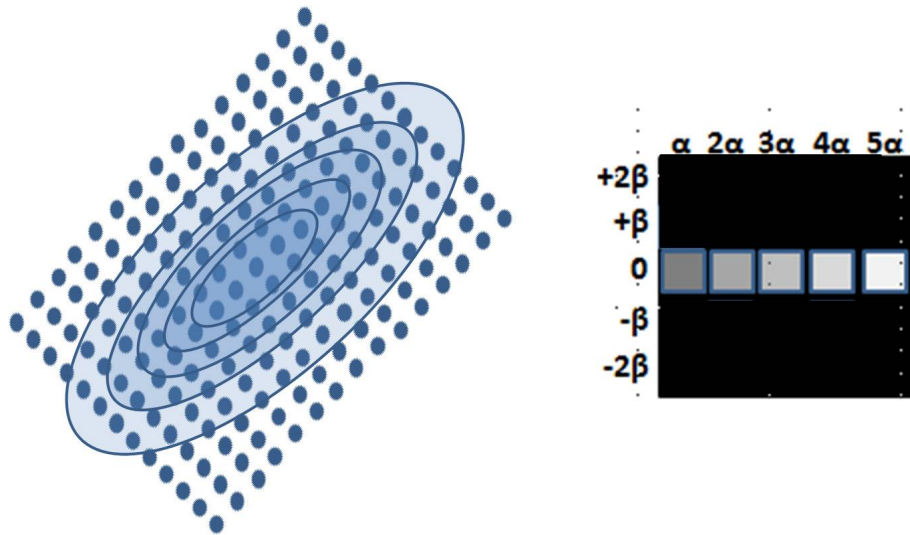


Fig. 44: Spin Image population for a uniform distribution of points on a planar surface.

sampled in consistent steps in the object space, the geometric information encoded by a spin image with any set of generation parameters is not unique, and will not guarantee effective matching performance. For a plane, all samples mapped to the spin image will fall on the α axis, and the pixel values will tend to increase with α .

For this reason, as spin image support distances become smaller, their uniqueness decreases making the possibility of frequent mismatching more likely. In his dissertation, Johnson proposes using a statistical analysis of the correlation coefficients of the set of all possible correspondences between the points on the model and the points on the scene, or current set of measurements. Extreme outliers are those whose correlation scores are at least a distance of $3f_s$ from the population mean. The distance f_s in this case is the distance between the upper and lower fourths of the population. In the case of a sphere or plane however, there are no outliers. Another way for spin images to fail is related to object space symmetry. Since spin images are translation and rotation invariant, they, like all other point based descriptors are subject to confusion in cases of repeating structures. Since spin images can also be global de-

scriptors by selecting a very large support distance, global asymmetries can be used to distinguish similar local structures from each other, however, global descriptors can see other difficulties in cases where subsequent data sets represent partial scans of an object. Finally, the construction of a spin image is sensitive to the estimate of the oriented point surface normal. For this reason, the process of mesh regularization is crucial to ensure that every point has the best chance of a repeatable estimate of the local surface normal. For the case where meshes or point distributions are not regularized, the uncertainty in the estimated direction of the surface normal can be significant. This can be problematic when attempting to apply spin-image descriptors on the fly in proximity operations experiments based on LIDAR data. The proposed point based feature description technique, the *Multi-Field Diffusion Flow Signature* is inspired by the spin image technique, but also exhibits the following qualities which make it useful for a data-driven solution to the 3D-point correspondence problem:

1. No mesh generation or regularization required for descriptor generation.
2. Zero dependence on feature surface normal estimates which can be subject to uncertainty.
3. Enhanced texture information to treat local surface variations and texture variations in the same framework based on the diffusion flow generated during scale space feature detection.

A. Multi-Field Diffusion Flow Signatures

Multi-Field Diffusion Flow Signatures are point based descriptors which encode the local information of multi-field scale space extrema. Like spin images, they can be defined at any point in a set of measurements, but it is most effective to build them

in order to describe the extracted feature points resulting from the process described in chapter 4. Also, much like spin images, a single field Diffusion Flow Signature is a two dimensional array which encodes a radial distance and a normalized vector which represents the diffusion flow velocity of each point in a neighborhood around a feature. In order to avoid the sensitivity of the descriptor population to the local surface normal estimate, and to use the values determined by applying the diffusion equation to the surface and its accompanying textures directly, the values for α and β are given by:

$$\alpha = \sqrt{(X_j - X_i)^2 + (Y_j - Y_i)^2 + (Z_j - Z_i)^2} \quad (5.14)$$

$$\beta_k = \frac{DoG(u_j, v_j, k_i)}{argmax(DoGG(u_N, v_N, k_i))} \quad (5.15)$$

Notice that as opposed to the parameters used to define spin images that there is no dependence on an estimate of the local surface normal. The parameter α is simply the euclidian distance of between the i^{th} surface point and its j^{th} neighbor. The normalized diffusion flow velocity is reported for each of the k fields by the parameter β . For the LIDAR-imagery fusion problems discussed here k ranges from 1 to 3 where 1 denotes the 3D geometry field based on the curvature normal and the remaining indices describe the scalar texture fields. In this way, the x, y, z coordinates refer to the measured 3D cartesian coordinates of the measured surface points and the coordinates (u_j, v_j) refer to the image space coordinates of the j^{th} neighbor in the neighborhood u_N, v_N . Furthermore, the Difference of Geodesic Gaussian values $DoGG(u_j, v_j, k_i)$, are the values of the diffusion flow velocities or the diffusion displacements caused by applying the geodesic Gaussian approximation to the Laplace-Beltrami operator at the surface point associated with the data located at (u_j, v_j) . In this case, the local neighborhood is defined to be a spherical neighborhood around the 3D point

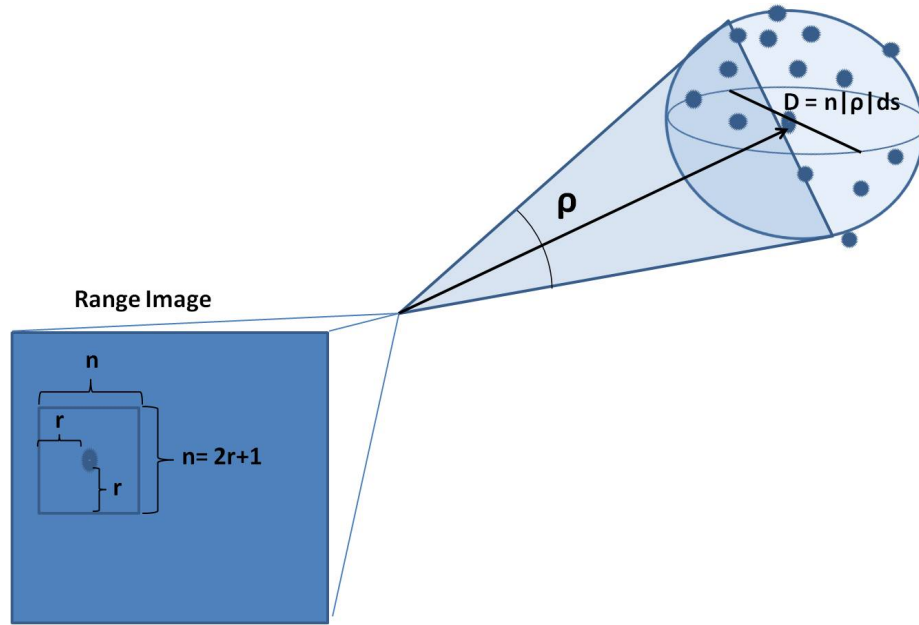


Fig. 45: The 2D and 3D Multi-Field Diffusion Flow Signature Geometry.

of interest. It is derived by first defining an $n \times n$ neighborhood about the 2D pixel location of the extracted feature point where $n = 2r + 1$ and r , is the neighborhood radius in image space. Therefore we have $u_N = u_i - r : u_i + r$ and $v_N = v_i - r : v_i + r$. The geometry is shown in Figure 45.

The 2D neighborhood is used to define an array of candidate nearest neighbors. The 3D neighborhood is defined by a maximum allowable radial distance which is bounded by one half the scaled diameter of the 2D neighborhood. The 2D neighborhood diameter n is scaled by the product of the magnitude of the range to the feature point and the subtended angle of a pixel in the range image. The 3D neighborhood is subdivided into m spherical shells such that the radial bins represent constant steps in radius from the feature point. No distinction is made with regard to a preference in surface normal direction. The second coordinate is computed for each field in the data set. The diffusion flow value for each field, is stored in the Difference of Geodesic Gaussian scale space at each pixel in the neighborhood. In each field, the $n \times n$ neigh-

borhood of each Difference of Geodesic Gaussian extrema is normalized by dividing each pixel member by the value of the element with largest magnitude. This results in an array of values between -1 and 1 . For the particular field from which the scale space extrema is extracted, all members of the neighborhood have values which are a fraction of the features diffusion flow value, however this is not necessarily the case in the other fields. The bounded set of values for each field are sub-divided into n bins exactly as in the process defined by the spin image technique. A bin size is defined and the radial, α , and the diffusion flow, $[\beta_1, \beta_2, \beta_3]$ values specify the Diffusion Flow Signature coordinates:

$$i_k = \lfloor \frac{\frac{W}{2} - \beta_k}{\Delta_{\beta_k}} \rfloor \quad (5.16)$$

$$j_k = \lfloor \frac{\alpha}{\Delta_{\alpha}} \rfloor \quad (5.17)$$

Here k is an index indicating the field. In the examples of this dissertation k takes on the values $1 : 3$ which indicate the $3D$ diffusion flow based on surface geometry, and scalar diffusion flow based on the Return Light Intensity and Visible Light fields respectively. The resulting descriptor can be visualized as three $2D$ arrays of equivalent size which integrate the local diffusion flow geometry as a function of radial distance from the center of the feature point. Figure 46 illustrates the three sets of Diffusion Flow Signature coordinates and the resulting arrays after applying the bilinear interpolation process of equations 5.9 through 5.12.

The Diffusion Flow Signature associated with the geometric information of the surface is perhaps the easiest to visualize. Consider a surface at two different stages of a diffusion process. Recall that each point on the surface is displaced by applying an approximation to the Laplace-Beltrami operator which implements surface evolution by mean curvature flow. Figure 47 shows the surface at times t_k and t_{k+1} . Each point

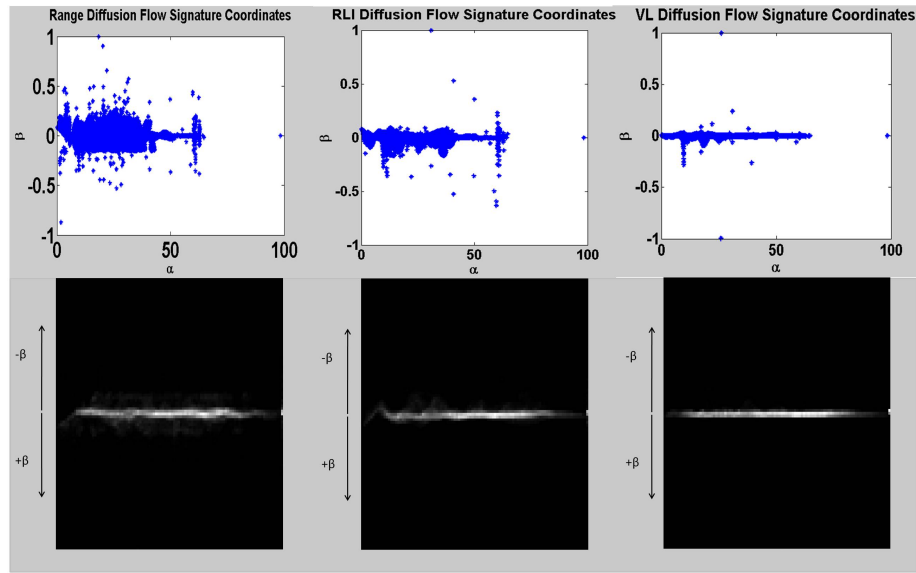


Fig. 46: Top: Diffusion Flow Signature Coordinates for an arbitrary feature with three fields. Bottom: The corresponding 2D arrays comprising the Multi-Field Diffusion Flow Signature.

on the surface is displaced in the direction opposite of its surface normal vector a magnitude equal to its local mean curvature value. These vectors can be defined by taking the difference in position $d = \vec{r}_{k+1} - \vec{r}_k$. The magnitude of d is the magnitude of the diffusion flow vector $d = \|H\hat{n}\|$ and is the value reported in each pixel of the first field of the Difference of Geodesic Gaussian Scale Space. For a constant curvature surface, all entries to the diffusion flow signature will reside in a row. This represents a limited information situation, and in the case of a plane, the row will be the center row exactly like a spin image.

For the scalar fields associated with each texture the diffusion flow values demonstrate the derivative of the texture information with respect to scale. For a feature which has the same Difference of Geodesic Gaussian value for all elements in the descriptor neighborhood, the normalized value will be the same for all pixels, and the top row of the Diffusion Flow Signature for that field will be populated exactly like the center row of a spin image for a plane. If the actual texture value is the same for

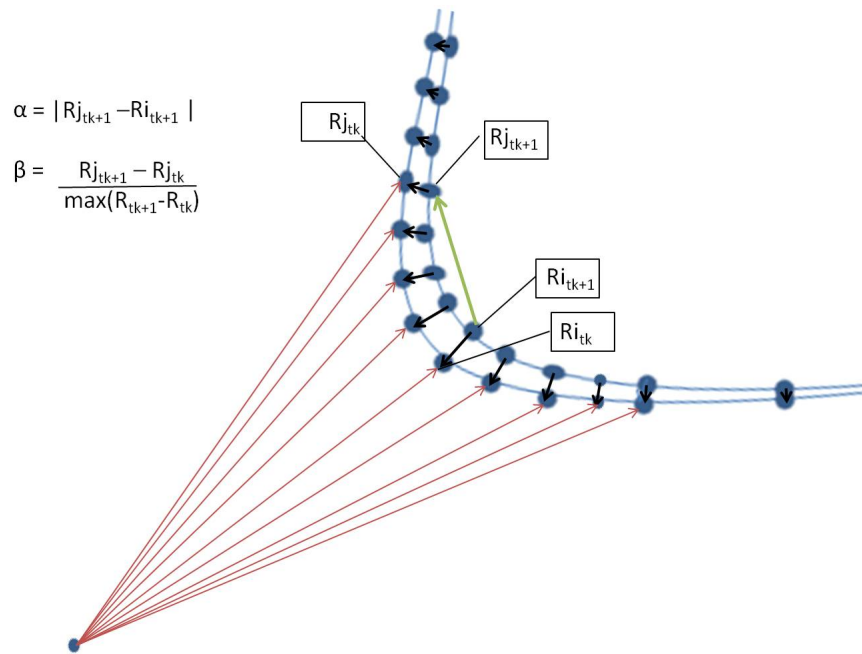


Fig. 47: Surface evolution encoded in a Diffusion Flow Signatures. The signature coordinates describe the euclidian distance between a feature point center and neighboring measurements along with the normalized magnitude of the diffusion flow vectors.

all elements, the DOGG value will be zero, and the center row of the Diffusion Flow Signature for that field will be populated like a planar spin image. In the case of a constant valued plane, all $2D$ arrays in the Diffusion Flow Signature contain entries in the center row which is an indication of a zero information state. This behavior provides a simple means to evaluate candidate descriptors to determine if they contain sufficient information to be unique enough to have a high probability of resulting in a correct correspondence.

By using the information from multiple fields, more information can be conveyed using fewer points in the descriptor, which was also the conclusion in the work of [8]. In the case of a consistently textured plane, this descriptor provides no unique information just like spin images. However, for a textured plane, Diffusion Flow Signatures encode the observable variations which might make a location in a texture

3D data set unique. Since Diffusion Flow Signatures are only constructed at scale space extrema, which are required to be well defined maxima or minima in at least one field, the computed descriptors are never in a configuration which conveys no information. This allows features to be extracted and described on bland surfaces which contain unique local curvature variations, as well as zero information geometric structures which contain unique variations in texture. This fact, combined with the feature extraction process described in the previous chapter is unique to the framework of this dissertation and results in the robust extraction and description of keypoints which provide the best chance of 3D feature tracking and relative motion estimation in spacecraft proximity operations. The next section details the processes of matching Diffusion Flow Signatures and give specific demonstrations of the type of failure modes that can always occur when using local, point based descriptors along with the various tuning parameters available to overcome the various challenges in feature detection, description, and matching in spacecraft proximity operations.

B. Matching Diffusion Flow Signatures: Approach and Remaining Challenges

Due to the capacity of Diffusion Flow Signatures to describe multi-field data, the matching process can be very versatile. In some applications, it may be sufficient to match only the geometric field of the descriptor, in others it may be more important to match the texture information. Independent of the criteria used to define an acceptable match, the approach to evaluating candidate matches involves the computation of the linear correlation coefficient of equation 5.13 for each field. Computing multiple correlation coefficients for a candidate match can allow for a higher degree of discrimination between otherwise similar looking features. The addition of field

information can be the subtle difference between features experiencing several many to one mappings, and sets of unique one to one correspondences. What is desired is a method to determine to what degree each field contributes to the uniqueness of a feature, and to define a weighting which reflects the relative importance of each fields contribution to the descriptor. In the previous section, an intuitive interpretation for each descriptor field was described which demonstrated that poor feature locations would result in descriptors with entries in only a small number of descriptor elements. A field with unique information results in an array that has most of its elements populated. For this reason we define a weighting which reflects the overall fill factor of each field in each descriptor involved in a correspondence evaluation. The fill factor γ is computed by taking the ratio of the number of non-zero bins of the Diffusion Flow Signature to the total number of bins in the array.

$$\gamma_k = \frac{\sum_{i=1}^m \sum_{j=1}^n DFS(i, j, k) > 0}{mn} \quad (5.18)$$

The fill factor γ_k is computed for each of the k fields in the descriptor for a feature and its candidate match. The overall similarity score is then computed by defining the weighted average:

$$\Xi = \sum_k w_k R_k \quad (5.19)$$

Here, R_k is the linear correlation coefficient computed for the k^{th} field of the descriptors and the weights w_k are defined as:

$$w_k = \frac{\gamma_{i,k} \gamma_{j,k}}{\sum_k \gamma_{i,k} \gamma_{j,k}} \quad (5.20)$$

The individual fill factor values are bounded by $[0, 1]$, therefore the product of the fill factors for the i^{th} and j^{th} feature given by $\gamma_{i,k} \gamma_{j,k}$ is also bounded on the same interval.

For the case where all fields have 100% fill factor, the weighted average of equation 5.19 results in an equal weighting between all linear correlation coefficients. The fill factor values are related to the variance of the local diffusion flow displacements in that for bland surfaces (flat plane or constant curvature) or textures, w_k reports a small value, and for large curvature variations or texture variations reports a large value. In this way, features which contain more unique information in one field than the others, the weighting reflects the relative importance of each fields contribution to the descriptor. Comparing features in this way has been shown to very rapidly eliminates potential correspondences which do not have sufficient similarity in well defined fields. Additionally, this matching process can also penalize feature matches which have poorly defined descriptors by requiring individual weights to be above a threshold before they may participate in the weighted average.

Figure 48 demonstrates the matching performance of the described method, feature sets are extracted for three measurement fields on the HST model in two different scans taken at different relative poses. The two lists of features are first sorted in descending order with respect to their $3D$ scale. Features are only given a score if they were localized in the same field. Of the 101 and 107 features extracted from each scene, only 35 satisfied the necessary conditions of the matching process which include thresholds on each individual correlation coefficient and weighting value. The sufficiency condition for candidate feature matches is provided by the residuals of the Optimal Linear Attitude and Translation Estimation algorithm explained in the next section. Given a set of features which contain a majority of correct correspondences which are accurately localized, the rigorously linear motion estimation process provides accurate relative motion estimates, as well as residual information which is a direct statistical characterization of the localization error. These residuals are used as a post-filter to remove any correspondences which are poorly localized either due

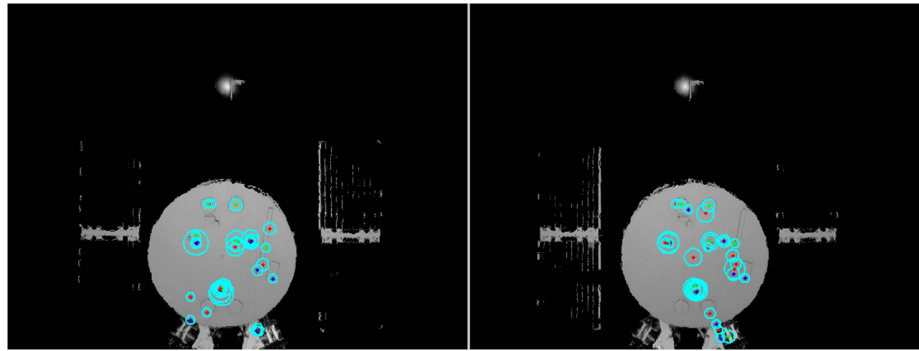


Fig. 48: 35 Matched features from approximately 100 extracted features in two scenes displaced by a 4 degree rotation of the HST spacecraft model.

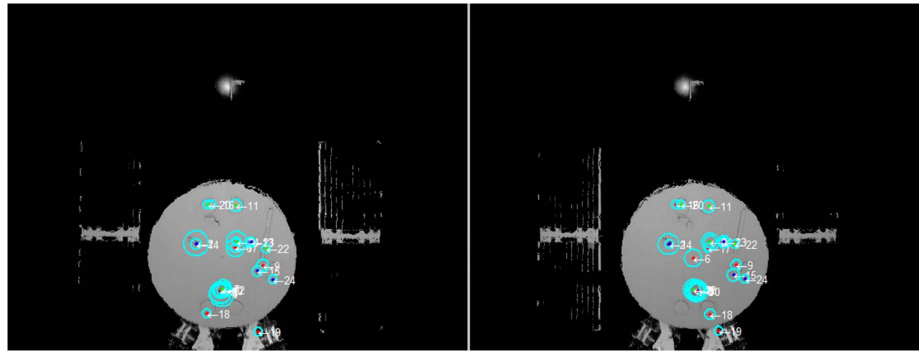


Fig. 49: $N = 24$ remaining feature points following OLTAE Residual Post-Filtering

to mismatches or poor accuracy caused by, local noise or sampling issues, repeating structures, or symmetry. Following the OLTAE residuals based post-filtering, the remaining matches provide an accurately localized set of correct correspondences illustrated by Figure 49.

The following examples demonstrate the common types of problems that exist for all 3D registration techniques based on local and sometimes even global, feature descriptors and motivate the need for effective statistical characterization of localization accuracy to provide the sufficient condition for features to truly be considered correct correspondences. Given that the extracted point based features are associated with local descriptors which are intended to be translation, rotation, and scale

invariant, any sub-structure or data subset which is displaced from the feature being described, but sufficiently similar to the feature of interest has the potential of scoring high in a similarity comparison, and therefore may be the cause of a mismatch. The easiest way to visualize this problem is to think of the global vs. local nature of a descriptor as an aperture problem. Consider the arrangement of white spheres used for the relative calibration between the camera and LIDAR shown in figure 50. If a descriptor is built based on a neighborhood of local data that is not much larger than the radius of the calibration spheres, there is not sufficient information to tell one from the other with great certainty. With a larger area for the descriptor however, it can be possible to distinguish the objects. Typically descriptor neighborhoods are either of a fixed size or include all available data. Since in most applications a single LIDAR scan is only a partial scan of the forward facing surface of an observed object, even “global” descriptors are somewhat local.

Consider now a cylindrical protrusion from a plane, a common structure on the HST model used for testing in the experiments of this dissertation. Figure 51 demonstrates that an isolated local feature identified on a rotationally symmetric structure can match an infinite number of locations on the same structure, or on any other similar structure.

Again, growing the neighborhood of the descriptor can help resolve ambiguities, but does not make this impossible. Features are caused by structure either in the measured surface, or in the texture. In both cases, this structure can display global symmetries, both rotation, and reflective, and for similar features that are sufficiently close together, even expanding the locality of the descriptor to include all measurements may not necessarily disambiguate even the best constructed descriptors. Couple this fact with inaccurate measurements, variations in the local surface sampling frequency, and other sources for sensor noise or physical sources for varia-

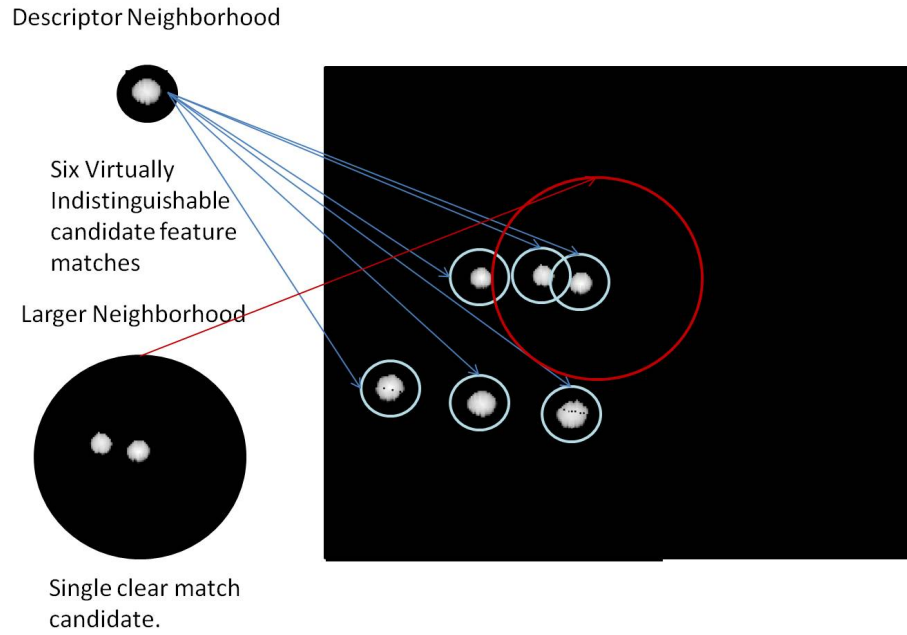


Fig. 50: Example feature matching for descriptors based on two different size localities. The smaller neighborhood feature does not have sufficient data to discriminate between the candidate locations for its match.

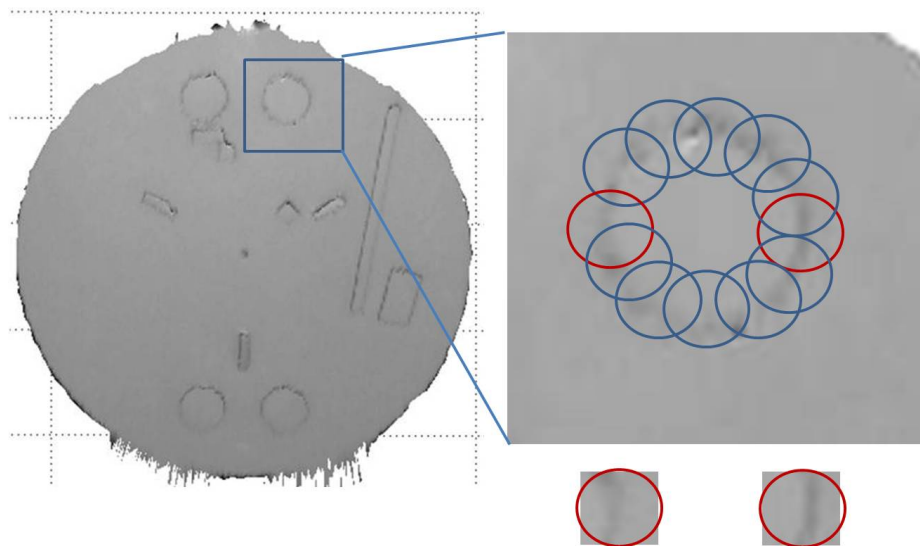


Fig. 51: Point based features extracted from a rotationally symmetric structure. High potential for ambiguity is demonstrated by comparing two candidate sections of the arc representing the locus of points representing curvature maxima and maximum texture gradients. It is virtually impossible to tell where the extracted local features were taken from.

tions in observed texture, and it can potentially be very difficult to guarantee that any feature descriptor pair should have a match in a subsequent observation. Even so, some features persist and can be uniquely described, and at least for a finite set of transformations and lighting conditions, can be accurately matched. Sometimes, in the case of the rotational symmetry example explained above, a very similar appearing extrema on a structure may be matched, and be “reasonably” close in proximity to the to its future observations. Features of this type can be difficult to localize with high precision, but, depending on the structure being measured, they may be all that is available. For this reason, it is important to be able to characterize the localization accuracy and probability of accurate correspondence on the fly as part of the motion estimation process. This is precisely the task performed by utilizing the Optimal Linear Attitude and Translation Estimator as the final filtering process for statistical analysis of residual errors. Since multiple views of features on the same rigid body must be mapped with only 6 free parameters (namely the sensor’s relative linear and angular displacement coordinates), redundancy can be exploited to accept or reject feature match hypotheses.

C. OLTAE: The Optimal Linear Attitude and Translation Estimator

The Optimal Linear Attitude and Translation Estimator was developed from the Optimal Linear Attitude Estimator [10] and is a rigorous extension of the techniques presented in by Mortari, Rojas, and Junkins [10] to include the estimation of translation. Given a set of multi-field, multi-scale $3D$ to $3D$ correspondences, the remaining task is to estimate the transformation which maps the set of features at time t to their correspondences at time $t + \Delta t$. Determining the transformation requires solving for the parameters involved in the six degree of freedom relative motion solution between

the LIDAR and the measured scene. The problem may be posed:

$$\vec{x}_i^* = R\vec{x}_i + \vec{t} \quad (5.21)$$

The common thread between most approaches to solve for the rotation matrix R and translation vector \vec{t} involves a parameterization which requires a linearized approximation of the rotation matrix for small angles. For example R may be parameterized by Euler angles by:

$$R(\phi, \theta, \psi) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\psi) & \sin(\psi) \\ 0 & -\sin(\psi) & \cos(\psi) \end{bmatrix} \begin{bmatrix} \cos(\theta) & 0 & -\sin(\theta) \\ 0 & 1 & 0 \\ \sin(\theta) & 0 & \cos(\theta) \end{bmatrix} \begin{bmatrix} \cos(\phi) & \sin(\phi) & 0 \\ -\sin(\phi) & \cos(\phi) & 0 \\ 0 & 0 & 1 \end{bmatrix} \quad (5.22)$$

Since the introduction of orientation coordinates to parameterize R leads to Eqs. 5.22 being nonlinear in the rotation coordinates (e.g. ϕ, θ, ψ), the estimation of $[\vec{t}, \phi, \theta, \psi]$ is inherently nonlinear. A more effective, and strictly linear approach involves the Cayley Transform. Recall that a rotation matrix may be parameterized by:

$$R = [I + \tilde{q}]^{-1}[I - \tilde{q}] \quad (5.23)$$

Here \tilde{q} represents the matrix form of the cross product operator and the vector \vec{q} is the Gibb's vector.

$$\tilde{q} = [q^\times] = \begin{bmatrix} 0 & q_3 & -q_2 \\ -q_3 & 0 & q_1 \\ q_2 & -q_1 & 0 \end{bmatrix} \quad (5.24)$$

The components of the Gibb's vector are given in terms of the principle axis and angle by:

$$\vec{q} = \tan\left(\frac{\phi}{2}\right)\mathbf{e} \quad (5.25)$$

Substituting equation 5.23 into 5.21 and pre-multiplying by $[I + \tilde{q}]$ we obtain:

$$[I + \tilde{q}]\vec{x}_i^* = [I - \tilde{q}]\vec{x}_i + [I + \tilde{q}]\vec{t} \quad (5.26)$$

Computing the products on the right and left hand sides we can write the above expression as a function of the sum and difference of the associated $3D$ coordinates.

$$\vec{x}_i^* + [\tilde{q}]\vec{x}_i^* = \vec{x}_i - [\tilde{q}]\vec{x}_i + [I + \tilde{q}]\vec{t} \quad (5.27)$$

In order to write the above equation linearly in terms of the Gibb's vector components we use the property $[a^\times]b = -[b^\times]a$ giving:

$$\vec{x}_i^* - [\tilde{x}_i^*]\vec{q} = \vec{x}_i + [\tilde{x}_i]\vec{q} + [I + \tilde{q}]\vec{t} \quad (5.28)$$

$$\vec{x}_i^* - \vec{x}_i = [\tilde{x}_i^*]\vec{q} + [\tilde{x}_i]\vec{q} + [I + \tilde{q}]\vec{t} \quad (5.29)$$

$$\vec{x}_i^* - \vec{x}_i = [\widetilde{x_i^* + x_i}]\vec{q} + \vec{t}^* \quad (5.30)$$

where $\vec{t}^* = [I + \tilde{q}]\vec{t}$ are a set of pseudo-translation coordinates which along with the components of the Gibb's vector are linearly contained. Once the values for the Gibb's vector coordinates and the values for \vec{t}^* are determined by solving the rigorously linear least squares process, the true translation coordinates can be solved for by $\vec{t} = [I + \tilde{q}]^{-1}\vec{t}^*$. The standard linear least squares $\tilde{y} = H\tilde{x}$ is seen by writing the set of linear equations of 5.30 in the following form:

$$\vec{x}_{diff}^i = [\widetilde{x_{sum}^i}, I_{3 \times 3}] \begin{bmatrix} \vec{q} \\ \vec{t}^* \end{bmatrix} \quad (5.31)$$

where the H matrix is $3N \times 6$ and the \hat{x} vector is 6×1 . The residuals of this least squares process are only a function of measurement or localization error, and not associated with errors due to linearization or truncation. They are given by $\tilde{y} - H\hat{x}$ or $x_{diff} - \widehat{x_{diff}}$. An important element to our approach for least squares registration involves the use of individual residual values as a post filter for removing poorly localized or incorrectly matched features. A “residual radius” metric is defined which computes the euclidian distance between a feature point and its motion compensated match location.

$$D = \sqrt{[\tilde{y}_i - H\hat{x}_i]^T [\tilde{y}_i - H\hat{x}_i]} \quad (5.32)$$

Here the distance is computed over the three components of a single measurement from the set of N feature points. Computing this value for each feature match assigns a quality metric for each feature. The smaller the value D the better localized the feature. Figure 52 demonstrates this metric for an accurately localized match, and a common mismatch.

In order for the linear least squares estimation of the relative motion observed by matching corresponding $3D$ to $3D$ feature points to be accurate, at least three well localized points must be correctly matched. In many well textured scenes it may be possible to match on the order of hundreds of feature points, resulting in a least squares process that is heavily over-determined. In such a case, the localization error of each individual correspondence is less important, however, the removal of high residual error features does result in a better OLTAE motion estimate when recomputing the registration. Analysis of residual statistics permits false matches to

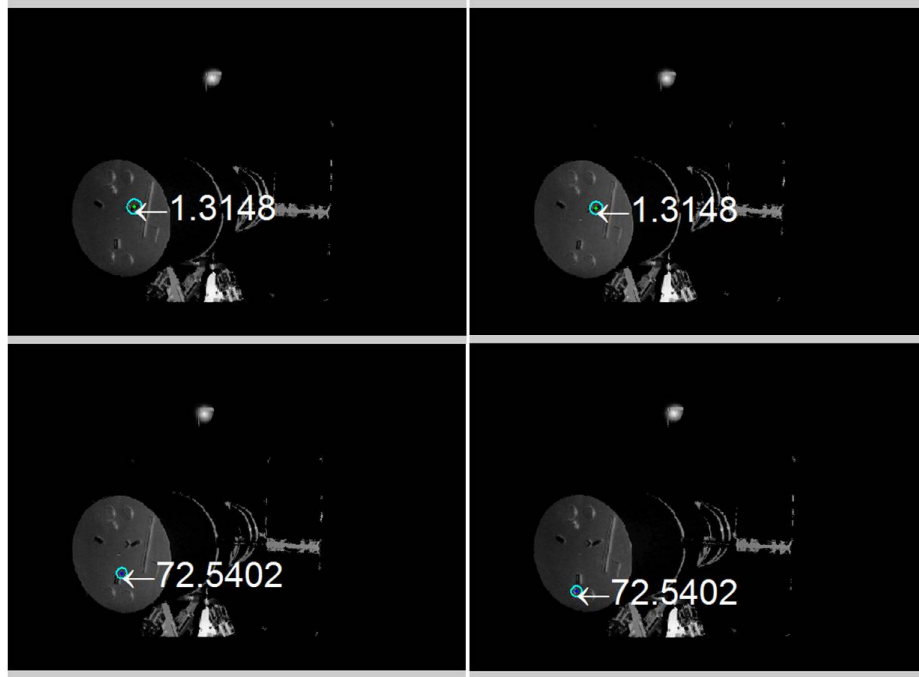


Fig. 52: Feature Fitness criteria based on OLTAE residuals. Top: A good match with a low OLTAE residual score. Bottom: A mismatch with a high residual score, scores above a threshold cause feature matches to be removed.

be easily defeated and deleted, and the statistics of the remaining feature residuals gives local insight into the accuracy of the model. For the case study used in this dissertation, the structures which cause features are in symmetric and repeating patterns. There are also few reliable structures to work with on the object while the rest of the data represents either bland surfaces, inconsistent surface representation due to local variations in lighting, or high frequency, repeating, and noisy data in the case of the solar panels. In a situation with all of these common sources for error and mismatch, it is critical to apply the OLTAE algorithm to a set of correspondences where the majority are well matched and localized. In some instances, as few as 10 to 20 reliable features are able to be extracted and matched, making each individual accuracy even more important. Such a scenario calls for an approach which balances required accuracy of each individual correspondence, with the degree to which the

least squares estimate is over determined.

The approach used in this dissertation defines a minimum number of required correspondences to be used in the OLTAE algorithm and iteratively increases the required localization error from a D value of 1 until the minimum number of points is exceeded. It is conceivable that for some objects, for instance, a bland sphere, too few features are able to be extracted in order to obtain an accurate six degree of freedom solution. On the other hand, other objects may exhibit redundant significant information in their geometry or in texture resulting in accurate solutions based on a high number of well localized correspondences. The degree to which these correspondences are achievable over large pose differences depends on the local sampling, and the consistency of the texture information. Since the appearance of surface features depend on lighting conditions, material properties, relative pose, and sampling effects, potential feature matches can be lost due to a number of reasons. Figures 53 and 54 demonstrate pre and post residual filtering matching results and show the feature sets used to estimate the relative motion, and the resulting aligned surfaces respectively.

In addition to using the individual residuals of the OLTAE least squares relative motion estimate, the covariance values may be computed and used to quantify the uncertainty in the estimated parameters. Figure 55 shows the magnitude of the covariance values for each element of the estimated Gibb's vectors and translation vectors for each of 28 LIDAR scans of the HST model taken while rotating the model in 2 deg increments on a robotic platform.

As a method of determining the magnitude of the error in the motion estimate, a corrective transformation was computed using the Iterative Closest Point algorithm [33], and the magnitude of the associated Gibb's vector components and translation vector values. Figure 56 shows the magnitude of the components of the ICP

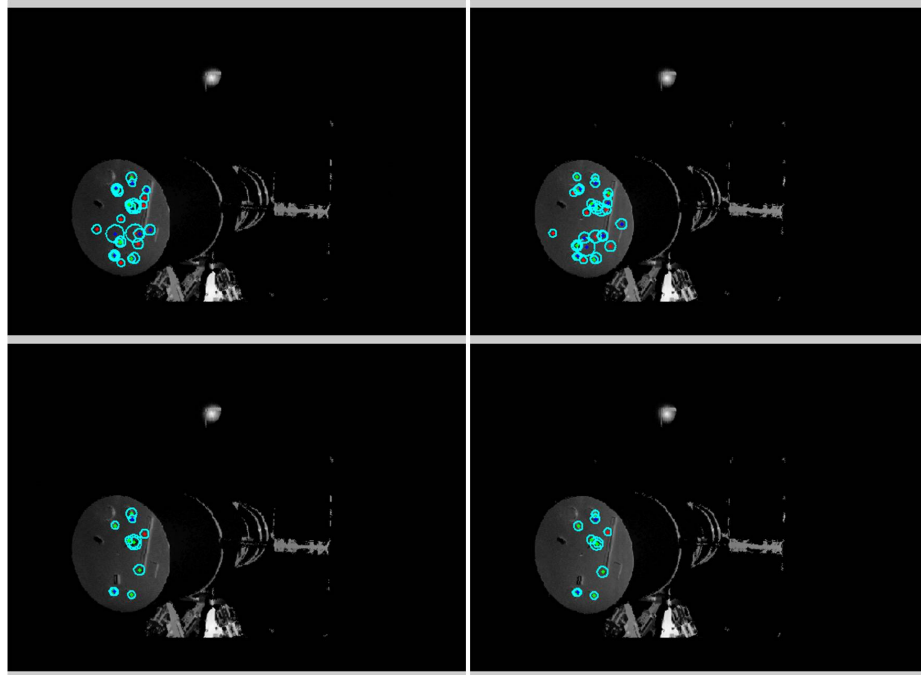


Fig. 53: Top: Matched Features used to estimate the alignment of figure 48a. Bottom: Feature matches reduced by OLTAE residual post filtering used to estimate the alignment of figure 48b.

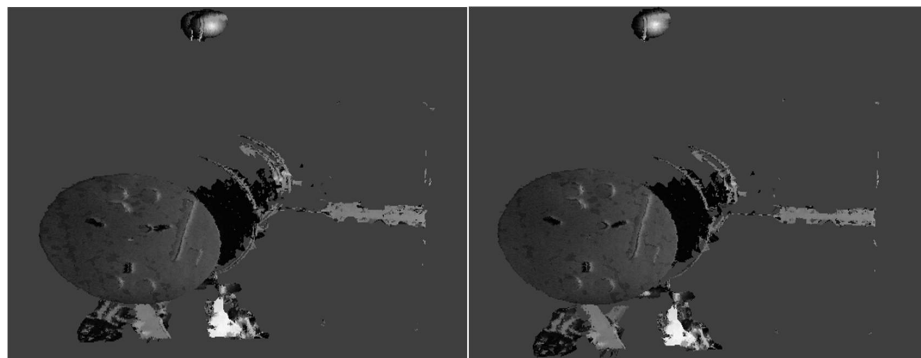


Fig. 54: Left: Resulting OLTAE surface alignment from unassisted feature matching. Right: Resulting OLTAE alignment using residual based post filtering.

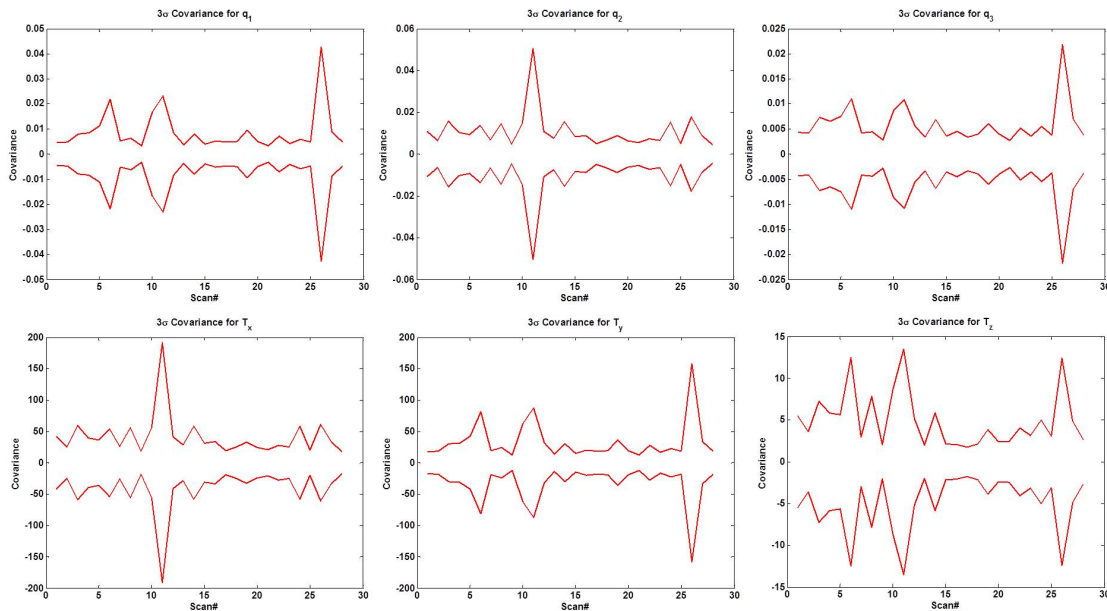


Fig. 55: Covariance for Gibb's vector and translation vector components estimated by the OLTAE algorithm for 28 consecutive relative motion estimates.

correction parameters and the 3σ covariance bounds. The large covariance values for the estimated parameters for scenes 7, 11, and 26 indicate a decrease in the accuracy of the model for those scenes by a factor of 2 to 3. These scenes would be candidates for additional analysis, or could be removed altogether from the resulting composite model.

The next chapter will describe the interactive *MFMSTracker* software developed in Matlab to implement the framework of this dissertation. The end to end process of proximal body observation, multi-resolution feature extraction, description, matching, and robust relative motion estimation is demonstrated along with built-in tracker performance evaluations with respect to sensor noise, artificial lighting variations, and re-sampling effects. Relative motion tracking based on Multi-Field Multi-Scale features and Diffusion Flow Signatures is then evaluated in Chapter 7 for a series of simple relative motions including out of plane rotation, as well as variations

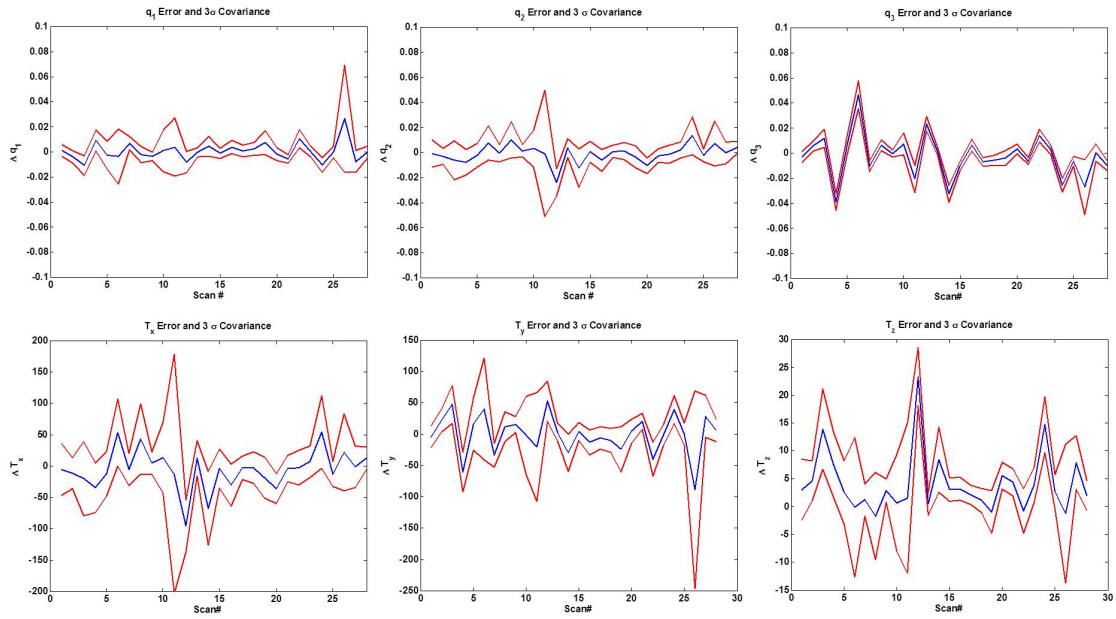


Fig. 56: Iterative Closest Point correction values for the Gibb's vector and translation vector components with 3σ covariance bounds for 28 consecutive relative motion estimates.

in lighting, object space sampling, and sensor noise.

CHAPTER VI

MFMS TRACKER: MULTI-RESOLUTION ANALYSIS OF MULTI-FIELD DATA FOR PROXIMITY OPERATIONS

A. MFMSTracker Software

The Multi-Field Multi-Scale Tracker (MFMSTracker) software package is a Graphical User Interface (GUI) application developed in the Matlab environment for $2D$ - $3D$ data fusion problems. It enables the simultaneous mapping and estimation of relative motion between a LIDAR-Camera fusion system and an observed proximal body. MFMSTracker constructs a scale space based on geodesic Gaussian convolution as defined in chapter 4 of this dissertation. Scale space extrema are identified and associated with Multi-Field Multi-Scale features which can be stored for use by a subsequent feature matching, motion estimation, and model registration process. The key components of the MFMSTracker software are:

1. .rtpi LIDAR file Reader and image input functionality
2. Camera to LIDAR Extrinsic Calibration
3. $2D$ and $3D$ Multi-Field Data Display
4. Multi-Field Scale Space Construction by Geodesic Gaussian Convolution
5. Feature Extraction and Description by Multi-Field Diffusion Flow Signatures
6. Feature matching, display, and annotation
7. Control over Required Feature Quality and Number of Features

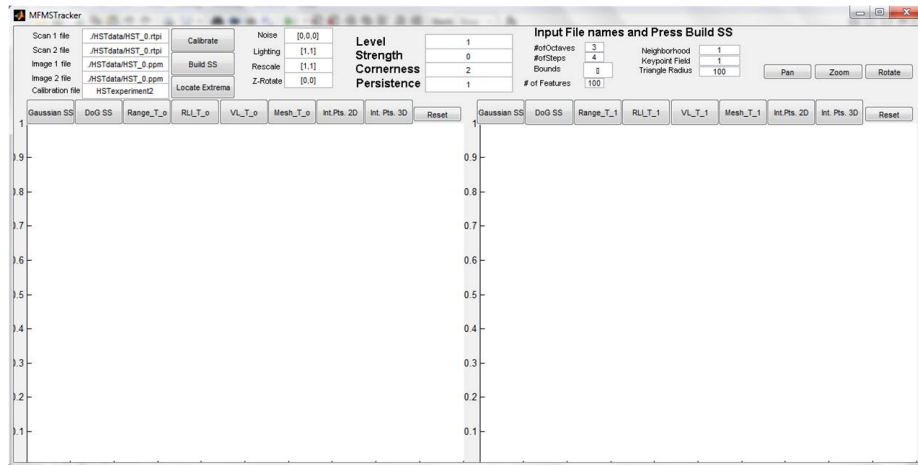


Fig. 57: The MFMSTracker GUI environment.

8. Data/Process Modification with respect to Noise, Lighting, Rotation, Sampling
9. Motion Estimation
10. Model Registration and Merging

Figure 57 displays the starting screen that appears when initializing the MFMSTracker software. The GUI is designed to demonstrate pairwise registration of subsequent observations of a proximal body by a calibrated Camera-LIDAR fusion system. The Camera-LIDAR fusion system used in this dissertation consists of the DeltaSphere3000 scanner and a single 1024×768 imager contained in the BumbleBee XB3 stereo camera setup from Point Grey Research. Assuming that both the LIDAR and the camera are individually intrinsically calibrated, an extrinsic calibration can be obtained by the methods described in chapter 3 by clicking the “calibrate” button.

A secondary GUI interface is presented which is the internal calibration routine to MFMSTracker. On this screen, the .rtpi filenames for the calibration scene and associated camera image file must be input along with the number of calibration objects, for example the 6 calibration spheres used in the example in chapter 3. The

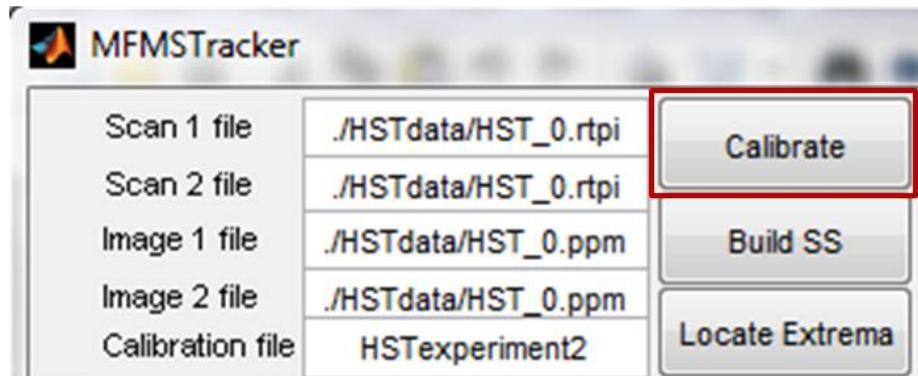


Fig. 58: The MFMSTracker Calibration Routine.

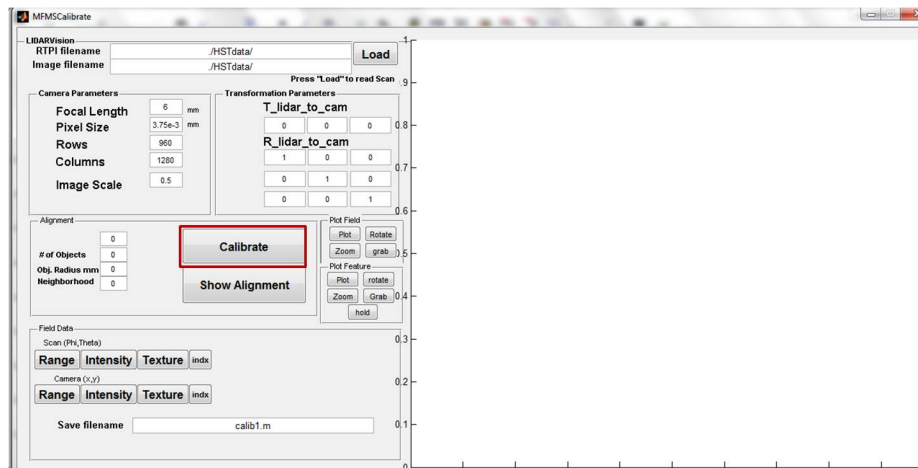


Fig. 59: The MFMSCalibrate calibration process.

user may also specify a radius in millimeters for the calibration object, as well as the expected image radius in pixels. Finally the user may also change the default values used for describing the camera intrinsic parameters necessary for performing the extrinsic calibration.

As an alternative to using calibration objects, a manual calibration may be performed by interactively clicking user associated structures to establish correspondence and localization on the measured object, setting the object radius to zero, and image radius in pixels to a small neighborhood e.g. 3×3 pixels. Once the files are specified,

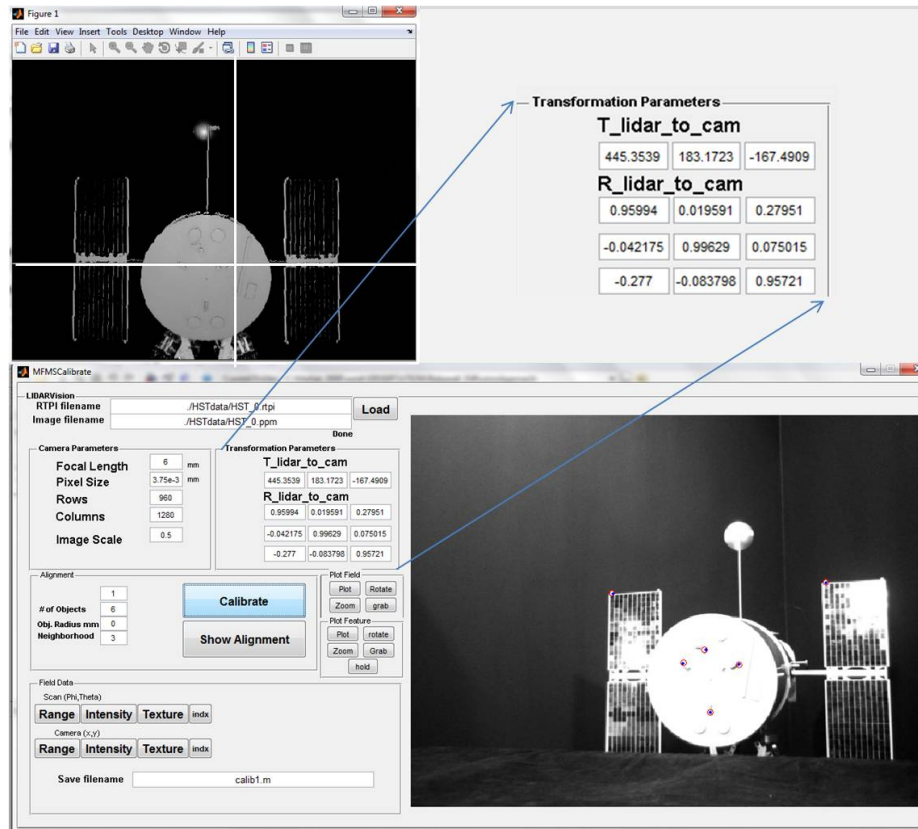


Fig. 60: Resulting calibration performed manually for the HST model.

the “Load” button is pressed to read in the 2D and 3D data. When the indicator bar reads “Done”, the “Calibrate” button may be pressed to begin the interactive, manual calibration process. For this process, the user must specify the number of point to point correspondences to be used in the calibration and enter this number into the “# of Objects” box. Once the “Calibrate” button is pressed, the user is presented a sequence of alternating images representing the LIDAR return light intensity field, and the camera image. The user then clicks the locations for each individual correspondence, when all of the n correspondences are identified, the calibration is computed, and the extrinsic parameters are reported in the “R lidar to cam” and “T lidar to cam” elements. Figure 60 demonstrates the identification of correspondences, along with a resulting calibration.

The calibration parameters can then be saved as an mat-file and entered in the MFMSTracker Calibration File Box for experiments which use a single calibration for several relative motions of an observed object. Once a calibration has been determined, all subsequent scans and images taken in that configuration are aligned by the same file. The next step is to specify the two scan files and associated images to be registered. Once the file names are entered, the “Build SS” button is pressed to generate the scale spaces of the two textured 3D data sets. The software then generates a scale space based on the input values for the number of octaves and number of steps per octave specified. The default values are three octaves at 4 steps per octave. This generates a scale space with 13 levels where the first level is the original data set. The Difference of Gaussian scale space has 12 levels and is constructed as a scalar difference of convolutions for the two texture fields, and a vector difference in 3D positions for the range field. The range is recomputed at each step to report the evolved surface data as a function of range, azimuth, and elevation angles. Once the scale space is constructed, the visualization toolbar may be used for each data set to observe the different levels and fields of the scale space. Figure 61 demonstrates the utility of the visualization toolbar. The top image demonstrates viewing different scenes in different fields, and the bottom image demonstrates viewing the same scenes and fields, but at different levels in the scale space. Figure 62 demonstrates exactly the same views, but for the Difference of Gaussian scale space. The three fields are the Range, Return Light Intensity (RLI), and Visible Light (VL), specified for the configurations at T_o and T_1 . The gray scale used to represent the data in each field can be customized by entering values for the upper and lower bounds inside the brackets in the “Bounds” box. Varying these parameters allows the user to make inferences and to interpret the results of specific processes applied to the underlying numerical data. Figures 61 and 62 show the result of using these parameters to view

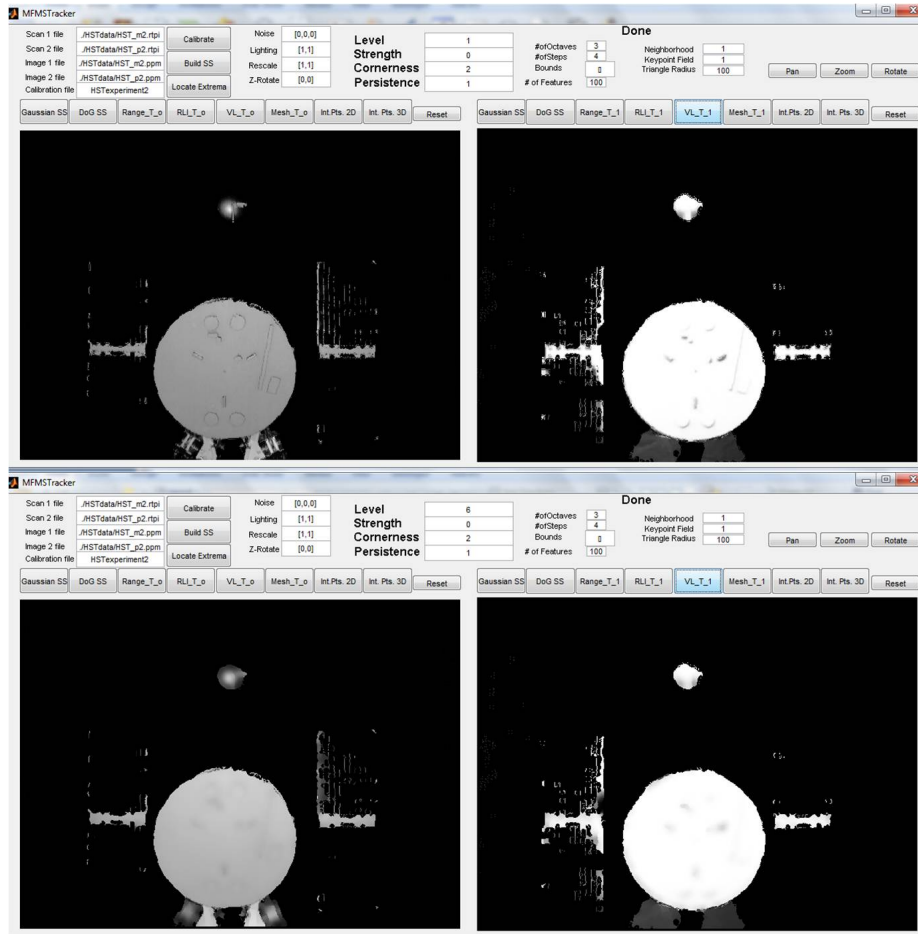


Fig. 61: The Visualization Toolbar is used to change the view to display different field, and scale space level information for the Gaussian Scale Space.

the Gaussian scale space and Difference of Gaussian scale space respectively.

Finally, the 3D visualization capability of MFMSTracker is demonstrated by pressing the $MeshT_0$ or $MeshT_1$ buttons, resulting in a fully textured 3D model of the measured object. To specify the level of the scale space used in the 2D and 3D visualizations a number between 1 and the total number of levels in the selected scale space can be entered the “Level” box. The “Keypoint Field” box is used to specify the field for texturing the 3D model, and the “Triangle Radius” is used to place a constraint on the circum radius of the triangles used in the surface mesh.

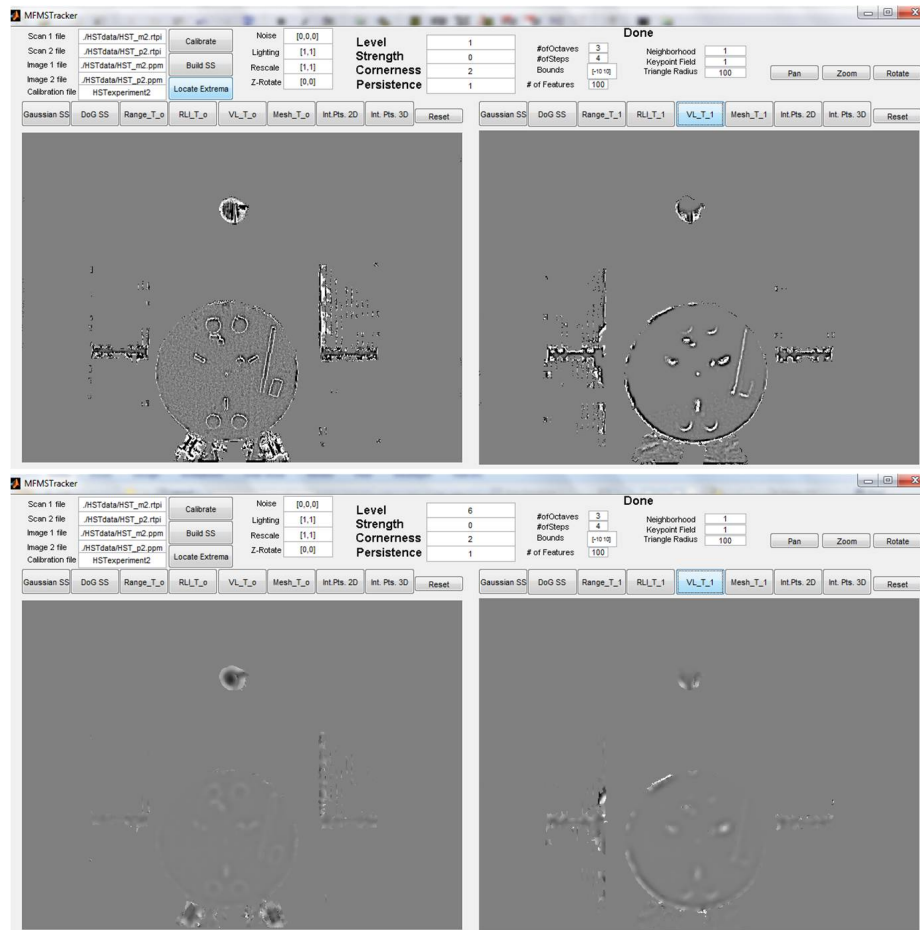


Fig. 62: The Visualization Toolbar is used to display information in the Difference of Gaussian Scale Space.

This constraint prevents any poorly defined triangles which may be in a degenerate configuration.

Following the construction of the scale space, the next step is to locate extrema. This process is called by clicking the “Locate Extrema” button. A non-maxima suppression process, compares each admissible scale space member to its neighbors in a local $n \times n$ region defined at the same scale. If the measurement is either greater than or less than all of its neighbors in either of the 3 fields, it is identified as a candidate feature location. An array containing all candidate feature locations in the scale space is output and used during the feature extraction process. The candidate extrema are then subjected to further evaluation of feature strength, cornerness, and persistence as described in chapter 4 by pressing the “Int. Pts. 2D” button. The top “n” resulting interest points which satisfy the user specified thresholds on strength, cornerness, and persistence are then plotted on top of the current view as shown in figure 63. The user may specify if interest points are to be detected in a single field by entering a 1, 2, or 3 into the “Keypoint Field” box, or any combination of the fields by using a colon notation, e.g. for all fields input 1 : 3 to the “Keypoint Field” box. The “Int. Pts. 3D” button is used to plot the resulting points on the 3D mesh by displaying them as spheres with radius proportional to the 3D scale determined for each feature. The proportionality constant defined for the radius of the marker used to represent the feature and its scale is defined in the “Neighborhood” box.

Finally, the data can be modified in four key ways. First, noise may be added by specifying the standard deviation of zero mean Gaussian noise to be added to each field. Noise affects the feature detection process in multiple ways. It changes the locations of the sampled data with respect to the surface changing curvature values and also changing where the points project to in the camera. This can cause a different texture mapping for the case where the camera image remains the same.

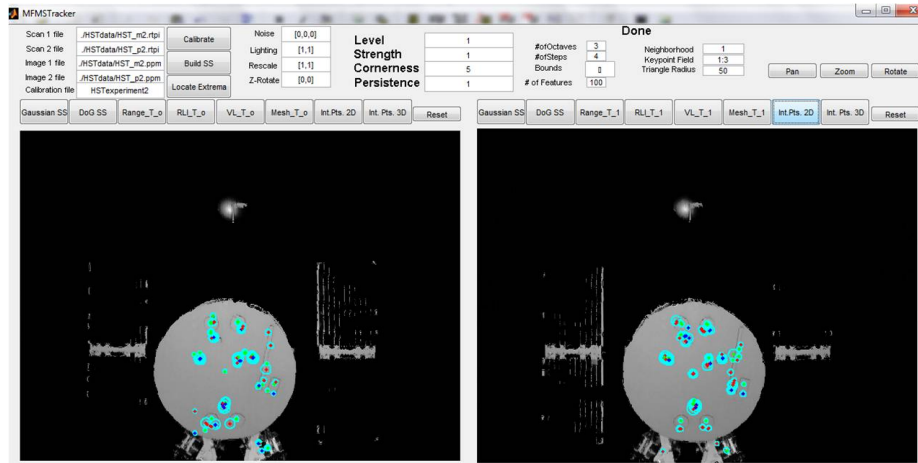


Fig. 63: Detected features displayed over the current view in the MFMSTracker Software

With the addition of noise in the texture fields, the combined result can potentially be significantly different looking data sets for small amounts of additive noise. The second method for modifying the data is to apply a nonlinear transformation to the lighting. The simple approach used in the software is to raise the texture field values to a power, and re-normalize the data set. Powers which are less than 1 reduce the contrast in the texture field, while powers greater than 1 increase the contrast. Figure 64 demonstrates the results of the lighting variation on the same scene.

The remaining modifications that are used in MFMSTracker are associated with artificial methods in re-sampling the data. The first is a simple re-scaling computed by specifying a scaling constant to increase or decrease the size of the original data set. The second method simulates a rotation of the data set about the bore axis of the LIDAR. Each method requires interpolation of the data which is performed using a nearest neighbor method to avoid artifacts associated with zero-valued pixels. Figure 65 shows the resulting data sets associated with each of these transformations.

Once data, modified or not has been used to construct a scale space, it can be viewed in $2D$ and $3D$, and manipulated with the standard pan, zoom, rotate, options

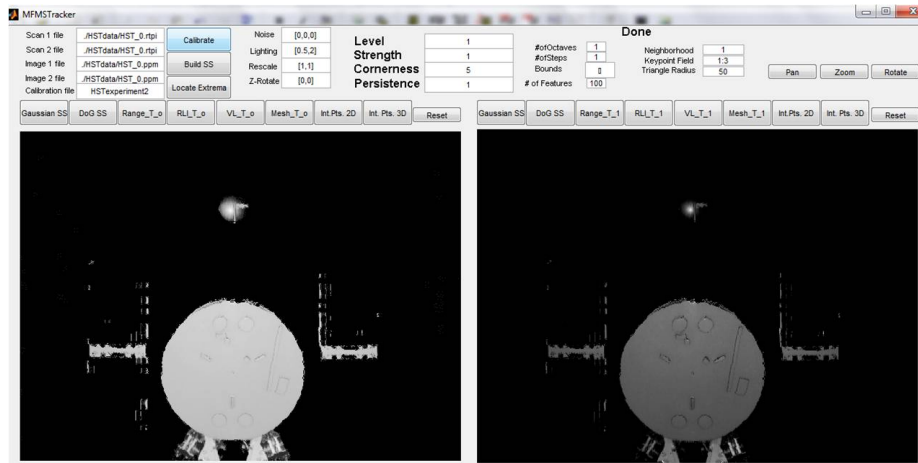


Fig. 64: Nonlinear transformation in Return Light Intensity for powers of 0.5 and 2 respectively.

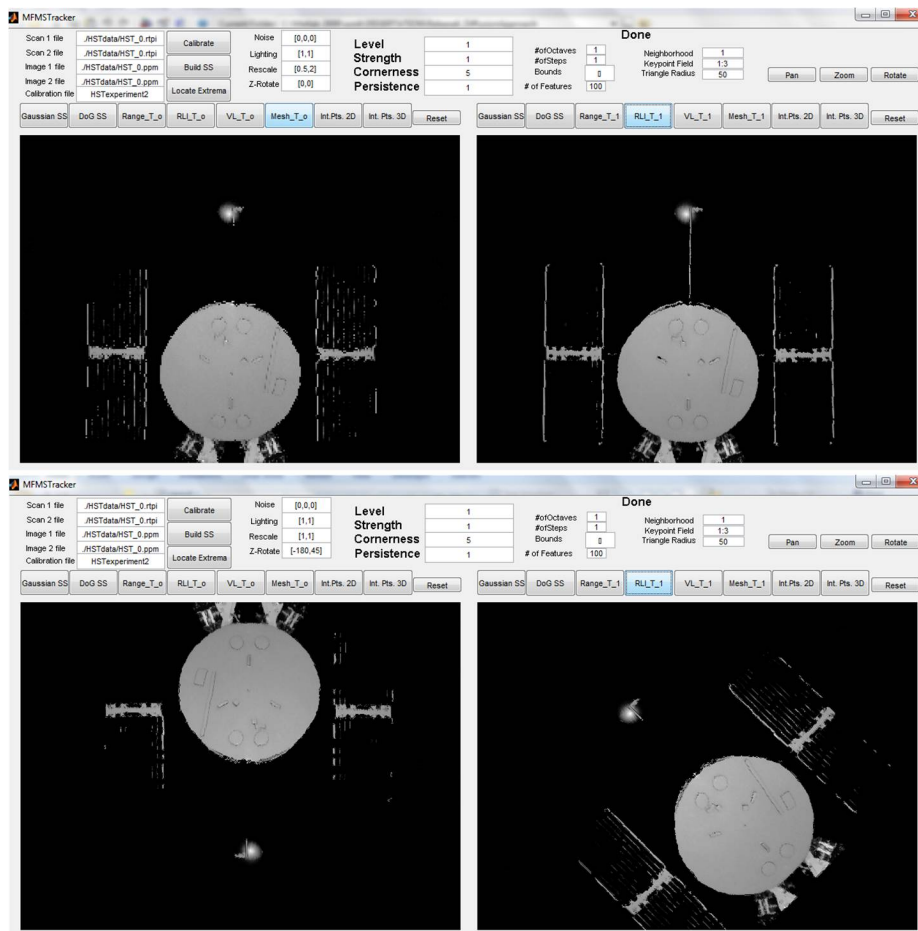


Fig. 65: Transformations causing artificial re-sampling associated with scaling and rotating the data.

in Matlab by pressing the appropriate button. In the event that the display needs to be cleared to see a better representation of the data, the “reset” button can be used. When features have been extracted for each measurement set, the keypoints and descriptors are used to estimate the relative motion by calling the MergeModels.m function, currently a manual process not integrated with MFMSTracker GUI. The next section explains the MergeModels process which reduces feature sets to high accuracy correspondences, estimates the relative motion using the OLTAE algorithm, and maps all subsequent observations of an object to the reference frame of the first observation.

B. Combined Model Registration and Relative Motion Estimation: MergeModels.m

The MergeModels function is used to evaluate scan sequences and to quantify motion estimation and object mapping performance based on extracted feature sets. It consists of three processes, feature matching by comparing Diffusion Flow Signatures, motion estimation using the OLTAE algorithm, and correspondence rejection using OLTAE residual post-filtering. The inputs required are data structures containing the sets of keypoints and descriptors for each scan, the original textured scan data for each scene, and labels used to associate the correct keypoints with the correct scene. The remaining inputs are constant parameters associated with the input scan file. The flow chart in figure 66 illustrates the process used to register subsequent scans using MFMS Features and Diffusion Flow Signatures.

As a final check, the variance of the OLTAE residuals associated with the best estimate for a scene in the set of scans can be used to accept or reject a scene for use in the final model. since motion estimates can suffer greatly due to mismatches, it

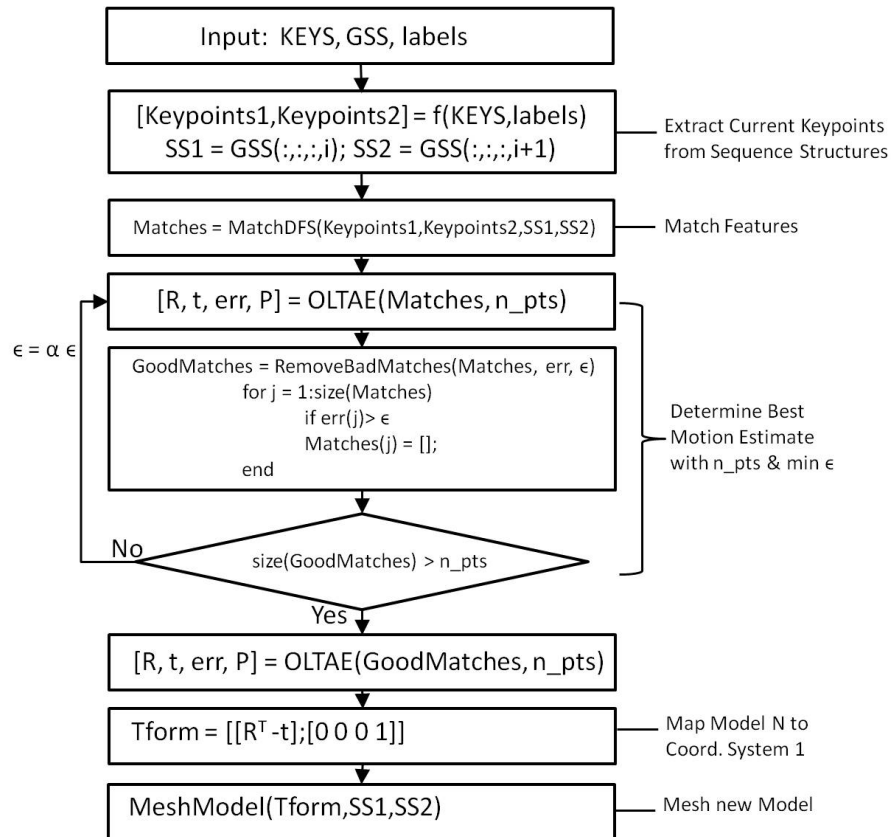


Fig. 66: Process Flow for the MergeModels function. Feature matching is iterated with motion estimation to identify reliable features to track and evaluate motion estimation performance.

is occasionally feasible that situations displaying symmetry, rapid photometric variations, and other sources for non-uniqueness in feature extraction and inconsistencies in surface appearance will cause a failure to achieve a confident and accurate relative motion solution. The use of the covariance to accept or reject scenes allows for control over model quality, as well as for the initiation of a reprocessing procedure which may require different tuning to more effectively address the current physical situation. An example of model rejection being used to achieve the best quality model is shown in the HST case study in chapter 7. The next chapter compares the methods outlined in this dissertation to a common Matlab implementation of the Scale Invariant Feature Transform [2,3] for the case of a planar surface and demonstrates the similarity in feature extraction for the case where the physical surface and image surface are the same shape. Then the general case of an unknown $3D$ object is considered and feature extraction, matching, motion estimation, and object mapping are evaluated for a simple circumnavigation experiment with real lighting effects.

CHAPTER VII

EXPERIMENTAL RESULTS

The Multi-Field Multi-Scale Feature detection framework developed in this dissertation exploits the information provided by associated 3D LIDAR measurements by defining a scale space built by geodesic Gaussian Convolution. Two main experiments were selected to illustrate the extension of scale space techniques to multi-field 3D data and demonstrate their utility in spacecraft proximity operations.

The first experiment considers a simulated constant range surface textured by imagery, and compares a common implementation of the Scale Invariant Feature Transform [44] applied in image space to Multi-Field Multi-Scale feature extraction in object space for two different scenarios. The first case considers a fully populated scene where every pixel was generated by a measurable scene point. Then, the methods are compared for the case where an object only occupies a portion of the sensor field of view, and zero-valued pixels are associated with the absence of a measurement. We stress in this case that the MFMS scale space generation and feature extraction processes are data driven with respect to object space measurements while the SIFT algorithm can not exploit that information.

The second experiment is a simple case study in spacecraft proximity operations. A scale model of the Hubble Space Telescope is observed by a calibrated Camera-LIDAR fusion system. The HST model is then articulated by the Holonomic Omnidirectional Motion Emulation Robot (HOMER) through a series of 2 deg rotations about the HOMER vertical axis. While more complex motions are possible, this motion is equivalent to moving the sensor system through a constant radius arc about the center of rotation in the HOMER coordinate system. The HST model is

illuminated by a point light source co-located with the LIDAR. The measured surfaces range from partially diffuse to highly specular and do not demonstrate the Lambertian behavior which is commonly required for many passive computer vision algorithms. MFMS feature detection and description by Diffusion Flow Signatures is applied to register 27 of 28 observations and to estimate the relative motion accurate to within 12 millimeters RMS. The results herein demonstrate the rigorous extension of scale space techniques to data driven estimates of relative motion in proximity operations, and the accurate mapping of proximal bodies with zero a priori information.

A. Comparison with 2D Multi-Scale Feature Extraction

Two dimensional Scale Space methods do not exploit available 3D information. They localize points in an image and attempt to be robust to two dimensional transformations and transformations of the image texture. Motion estimation methods based on two dimensional feature tracking attempt to estimate the six degree of freedom relative navigation solution from a large number of 2D to 2D correspondences. The approach in this dissertation concerns motion estimation based on 3D to 3D correspondences associated with interest points extracted from a textured 3D surface. For this reason, a multi-resolution feature extraction process was developed which identifies interest points as locations in textured 3D data which are local maxima or minima in diffusion flow velocity. This velocity is a function of the geometry and the texture. The diffusion flow of a surface is implemented by computing successive convolutions of the data with the Geodesic Gaussian. The difference in Geodesic Gaussian convolutions was shown to be a good approximation to convolution with the Laplace-Beltrami operator. The LB operator is the extension of the 2D Laplacian to manifolds. As was shown in chapter 4, the approximation to the LB operator

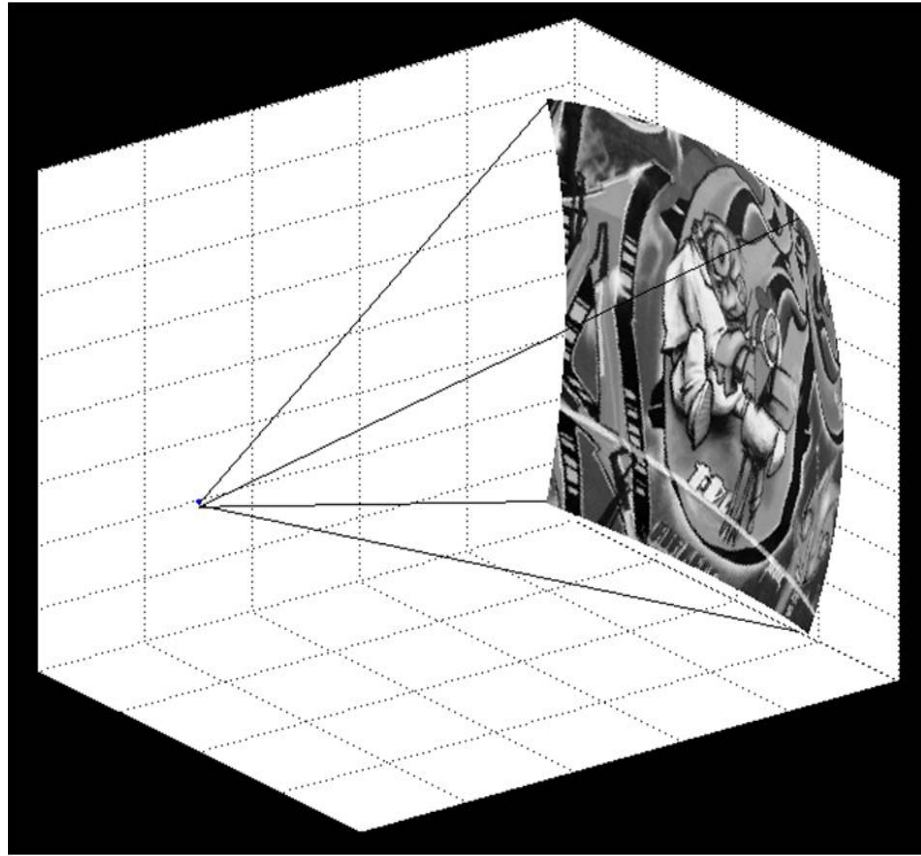


Fig. 67: An image texture is associated with a constant range surface sampled in equal angle segments.

takes into account the inter-sample distance in object space. In order to accurately compare the scale space feature extraction of image processing, to the methods of this dissertation, a common implementation of the Scale Invariant Feature Transform was compared against our results, by associating an image texture with a constant range surface. Figure 67 shows the constant range surface textured by the Graffiti image texture provided by the Oxford Visual Geometry Group [45], a common test image for scale space methods.

By sampling the constant range surface in equal angle increments, the inter-sample distance in object space is constant and in equal steps exactly as the pixels of an image are. In each comparison of the scale space techniques for object space and

image space, the largest 500 extracted features are chosen as a stable subset of features to be used to evaluate the repeatability of each method. Unique to this evaluation process is the use of OLTAE residuals to determine the number of well localized matches used to estimate the $3D$ transformation from $3D$ to $3D$ correspondences. To this end, three transformations to the original data are considered for comparison, the addition of noise to the texture, a nonlinear transformation of the gray scale values achieved by raising the texture values to a power and normalizing the data to the interval $[0, 1]$, and a rigid body rotation of the object space about the bore sight, which is equivalent to an image plane rotation. A perfect registration will achieve an identity rotation matrix and zero translation vector, indicating that all of the matched features had zero residual error and were identified to be in exactly the same locations in image space or object space. When evaluating keypoint repeatability in image processing, localization error is measured in image pixels. In this test, error is measured on the constant range surface with $R = 1000$ millimeters, for which 1 pixel is approximately to 3.5 millimeters. The metrics that are used to evaluate the feature extraction and matching of both techniques are the number of matches out of a possible 500, the mean and variance of the match residuals, the principle angle of the attitude corrective matrix required to return the scene to the desired identity rotation, and 2-Norm of the estimated translation vector.

1. Complete Ground Scene

The graffiti image, a common choice used in the evaluation of scale space feature extraction [24, 25] was selected as the texture for a fully populated constant range surface sampled in equal angle increments. Figure 68, shows the original image next to the image co-registered with the surface, the pixels in the image can be considered constant steps in x and y for an image taken by a camera, or constant steps in ϕ and

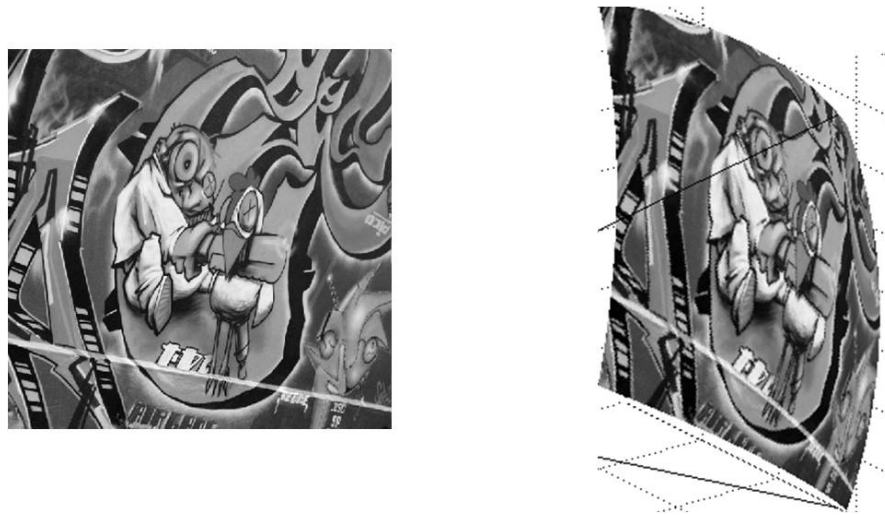


Fig. 68: Original texture used to color a constant range surface sampled in equal angle segments.

θ as shown in the examples of chapter 3. In the case of equal object space or image space steps, both methods reduce to building a Gaussian Scale Space. To illustrate the similarity between both methods with respect to feature extraction, figure 69 shows the 500 largest extracted features for both algorithms using 3 octaves and 4 steps per octave, a cornerness threshold of 5, and a strength threshold of 1.

The similarity in image/object space location and scale of the interest points extracted for both methods demonstrates that the Diffusion flow approach to constructing a scale space of object space measurements is comparable in performance to state of the art image based multi-resolution feature extraction techniques. To further evaluate the descriptor and show that the performance of MFMS interest points and Diffusion Flow signatures is competitive with SIFT, the descriptor matching performance was compared for variations in texture noise, $2D$ rotation, and lighting change. Figure, 70 shows the many variations tested as deviations from the original data shown in figure 68.

For each SIFT feature located in the data, the associated $3D$ point is deter-

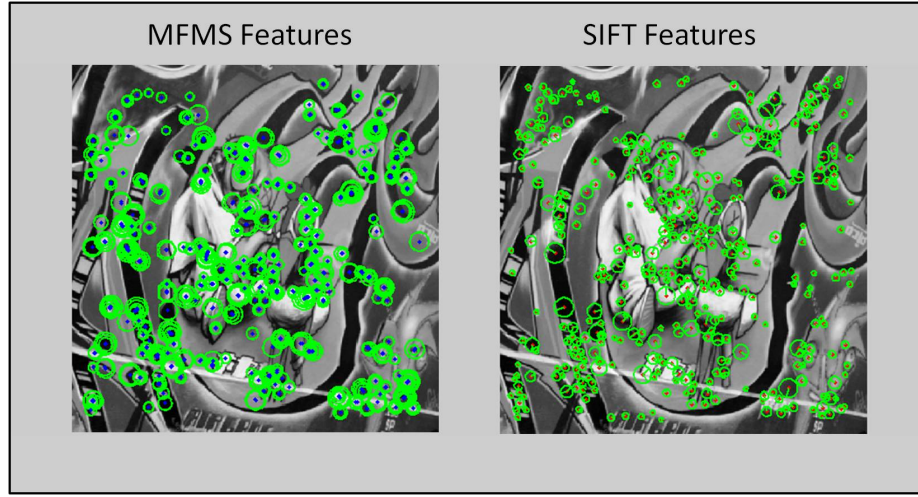


Fig. 69: Scale Space Feature Extraction on the Graffiti image texture field for the Scale Invariant Feature Transform and Diffusion Flow techniques.

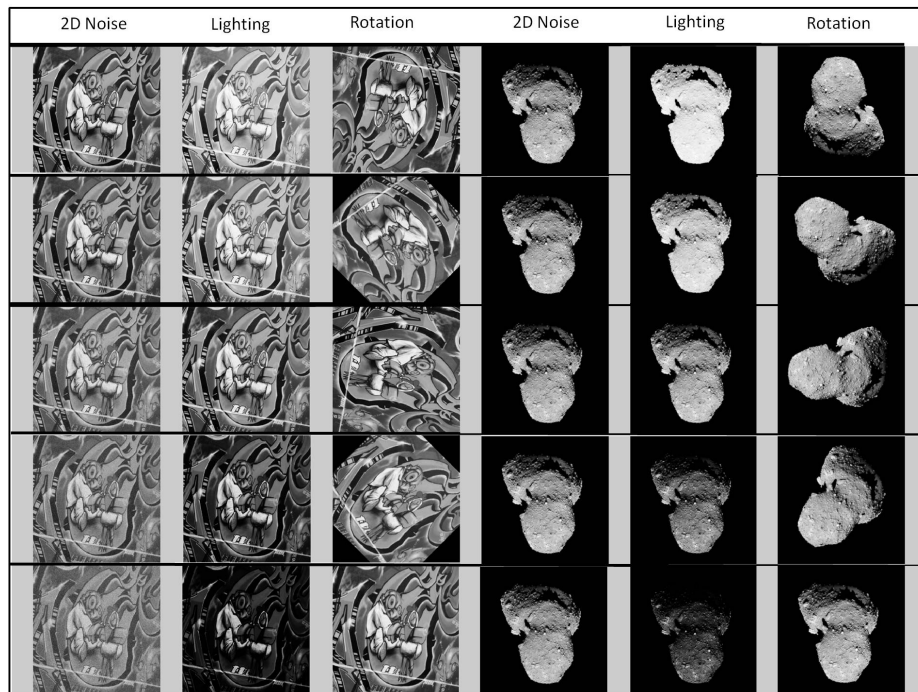


Fig. 70: The original Graffiti and Asteroid textures are corrupted and transformed to evaluate feature descriptor matching performance.

mined by interpolation of the surface while for each feature extracted using MFMS features, the $3D$ point is extracted from the scale space. The matching processes used were the standard best bin first matching approach with default settings provided with the SIFT algorithm and the multi-field matching process described in chapter 5. This evaluation process is intended to illustrate the consistency of feature extraction, matching and motion estimation afforded by each algorithm. If either algorithm extracts exactly the same set of features after a transformation, the subset of the 500 largest features will contain the same points, all of which will match correctly with zero-valued residuals, and a correct motion estimate of identity rotation and zero translation will be obtained. As the data is transformed, matches are lost due to different features being extracted from a slightly different scale space construction in different locations. In all cases, the untransformed case is compared to the transformed case, and a subset of features is retained as sufficiently similar to be called local feature matches. The OLTAE motion estimation algorithm is then applied to the resulting $3D$ correspondences. The errors are then analyzed using the above metrics. In many cases, errors are due to large residuals caused by mismatched features. The best indicator of feature extraction and matching performance is the mean and variance of the match residuals. Showing consistent values in these parameters over various transformations demonstrates robust performance in motion estimation, which requires accurately matched features to work effectively. The following plots summarize each algorithms performance with respect to image noise.

In figure 71 we see that varying the image noise up to 10% of the maximum texture value causes a steady decrease in the number of matches reported by each algorithm down to 50% in the most extreme case. The Multi-Field Multi-Scale matching policy appears to be a little more strict than the SIFT matching policy.

The translational and rotation accuracies of both algorithms in this case are

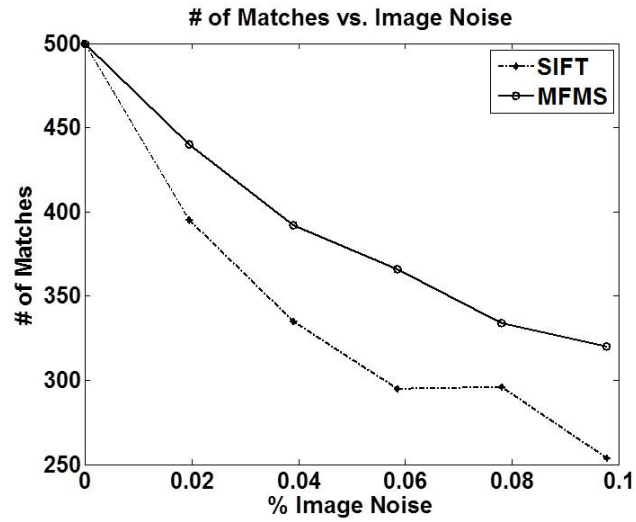


Fig. 71: The image noise is varied from zero to 10 percent causing a consistent decrease in the number of matches.

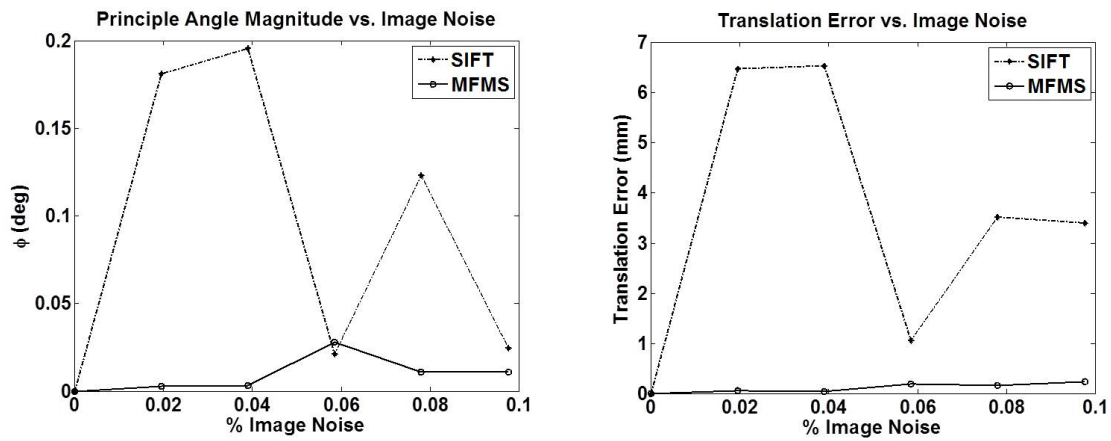


Fig. 72: Rotational and Translational accuracy vs. image noise for the MFMS and SIFT algorithms.

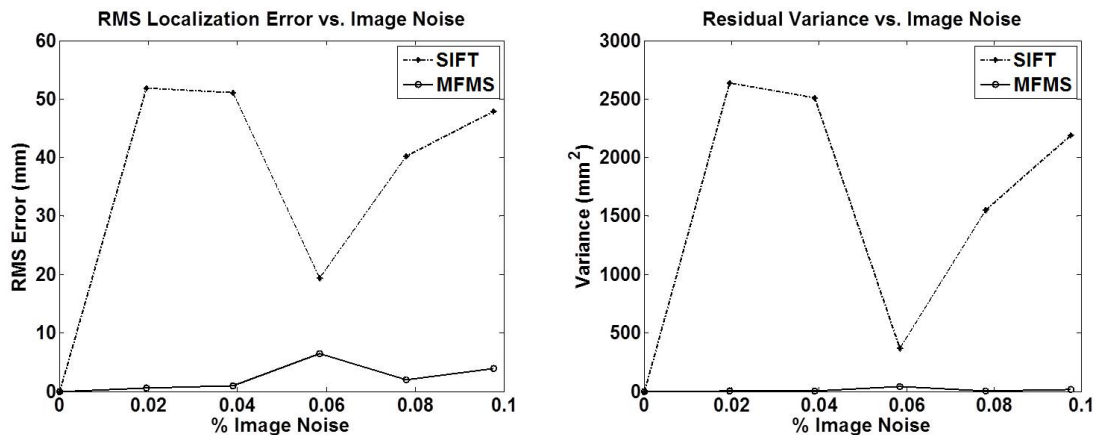


Fig. 73: Residual RMS and Variance vs. image noise for the MFMS and SIFT algorithms.

less than 0.2 degrees and 6.5 millimeters with the Multi-Field Multi-Scale features appearing to be more accurate. The Scale Invariant Feature Transform attempts to determine a sub-pixel location for each feature which for the purposes of this algorithm comparison is reduced to pixel accurate locations. For features which are right on the border between pixels, variations of up to 3.5 millimeters are possible for correct matches.

The root mean square error and variance values for the MFMS approach are lower than for the Scale Invariant Feature Transform throughout the tested variations in image noise which indicates a solution which is more robust to noise. Algorithm performance is completely scene dependent, and while MFMS features appear more accurate in this case, the performance of both algorithms are very similar with respect to rotation and lighting for the Graffiti scene. For the tested lighting variations, the texture values are raised to a power, the number of matches versus this power are shown in figure 74.

The lighting exponent is varied from $\frac{1}{4}$ to 2 which decreases the magnitude of scene gradients for values less than 1 and increases the magnitude of scene gradients

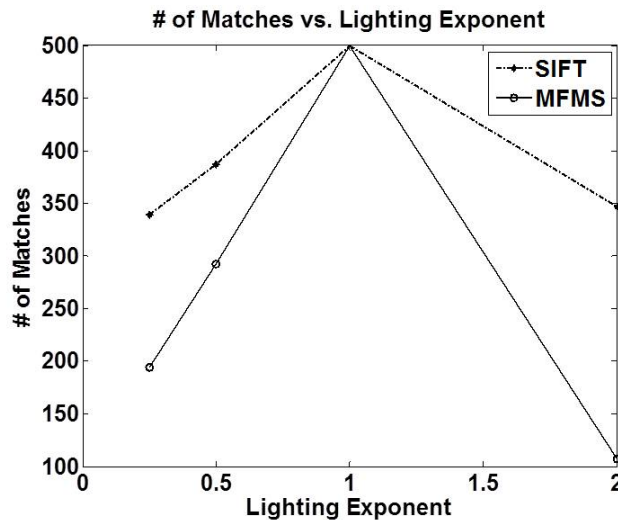


Fig. 74: Number of matches reported for the MFMS and SIFT algorithms vs. lighting variations.

for values greater than 1. By changing the relative magnitudes of scene gradients, the extracted corners can change in terms of feature strength and location. While the SIFT algorithm maintains about 70% of its features as matches throughout the lighting variations, MFMS features rapidly reduces the number of matches while still achieving very similar accuracies in translation and rotation.

Both algorithms achieve accuracies better than 0.1 degrees and 5 millimeters. For exponents less than one which artificially make the scene appear more diffuse the MFMS algorithm is slightly better, but the SIFT algorithm is slightly better for exponents greater than 1. Both algorithms have similar residual statistics, and are on the same order of magnitude of the SIFT algorithm performance for noise.

The previous two variations are functions of the field values contained in the image and not the spatial sampling. By testing the algorithms with respect to image rotation, we observe the variations in performance associated with a simple re-sampling process. For both algorithms, the convolutions are computed with operators implemented over square sub-images. As the image rotates, different image or surface

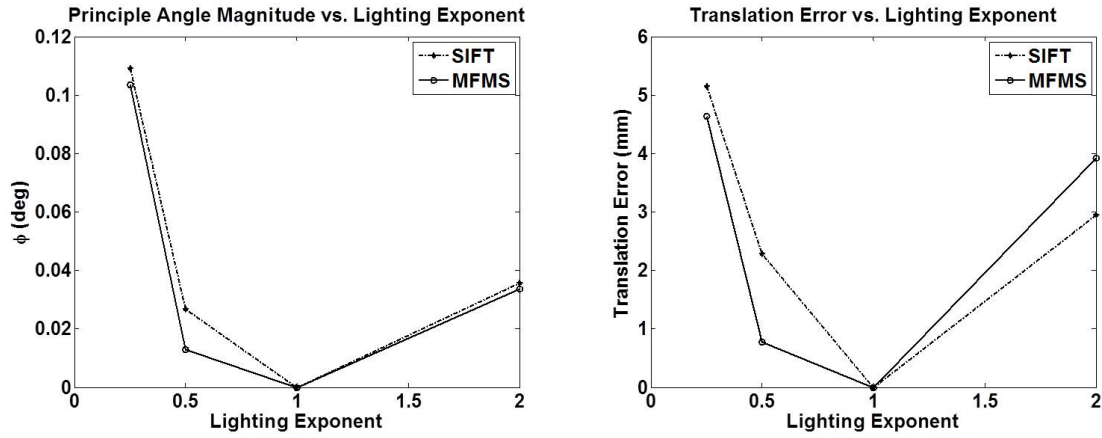


Fig. 75: Rotational and translational accuracies for the MFMS and SIFT algorithms vs. lighting variations.

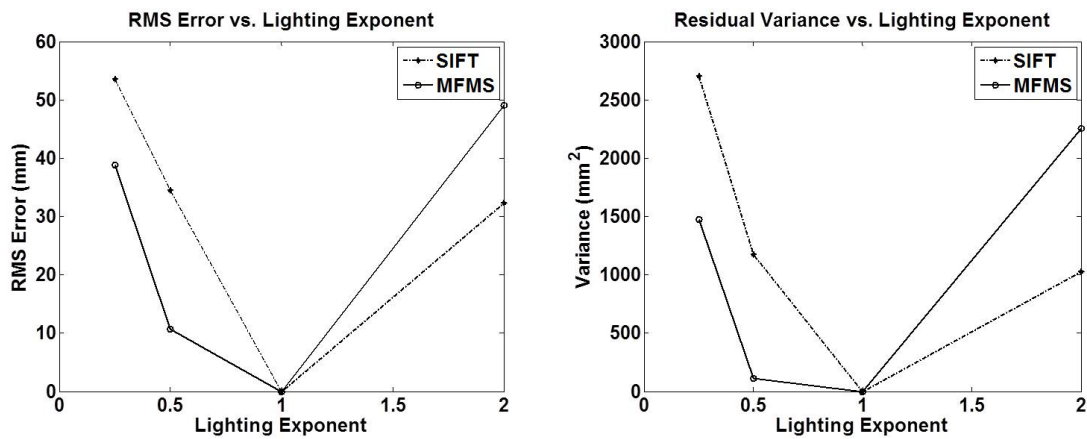


Fig. 76: Residual RMS and Variance values for the MFMS and SIFT algorithms vs. lighting variations.

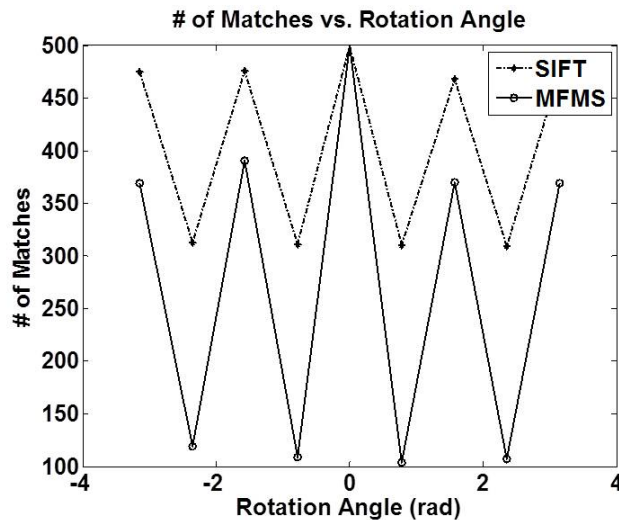


Fig. 77: Number of matches reported for the MFMS and SIFT algorithms vs. rotation.

elements are included in successive convolutions, resulting in slightly different scale spaces. While both algorithms are robust to this, we do see performance degradations caused by this fact.

The stricter matching policy used in the MFMS approach results in as many as 75% of the features not matching throughout the rotations. For this test the zero degree rotation case is compared to rotations from $-\pi$ to π and for both algorithms a significant decrease in the number of matched features occurs for odd multiples of $\frac{\pi}{4}$. The SIFT approach results in more consistent and accurate initial motion estimates but like the MFMS approach, sees performance degradations for rotation angles which are odd multiples of $\frac{\pi}{4}$.

Even though the MFMS approach is slightly less accurate than the SIFT method for rotations of this scene, the matches used to obtain the motion estimate achieve similar performance in terms of the accuracy of the residuals. This will be an important point in more complex scenes which include data dropout.

For the fully populated Graffiti image scene, the MFMS algorithm achieved sim-

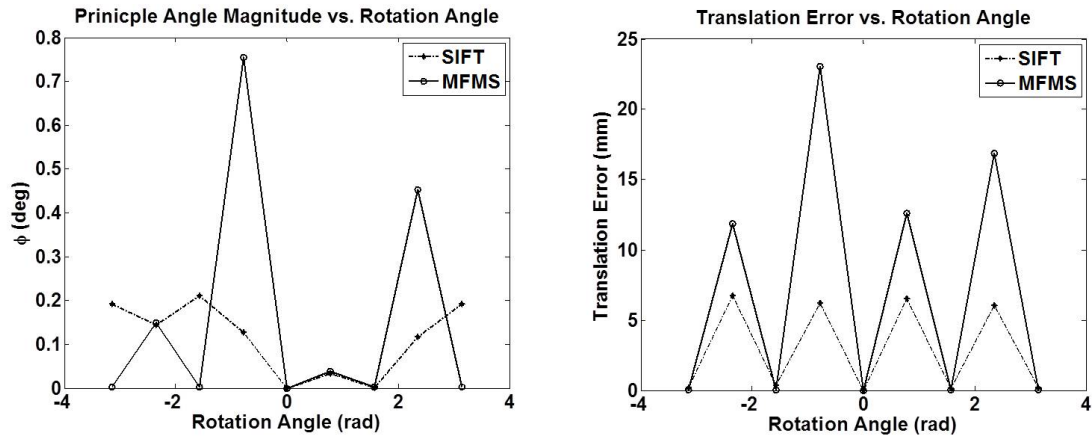


Fig. 78: Rotational and translational accuracies for the MFMS and SIFT algorithms vs. rotation.

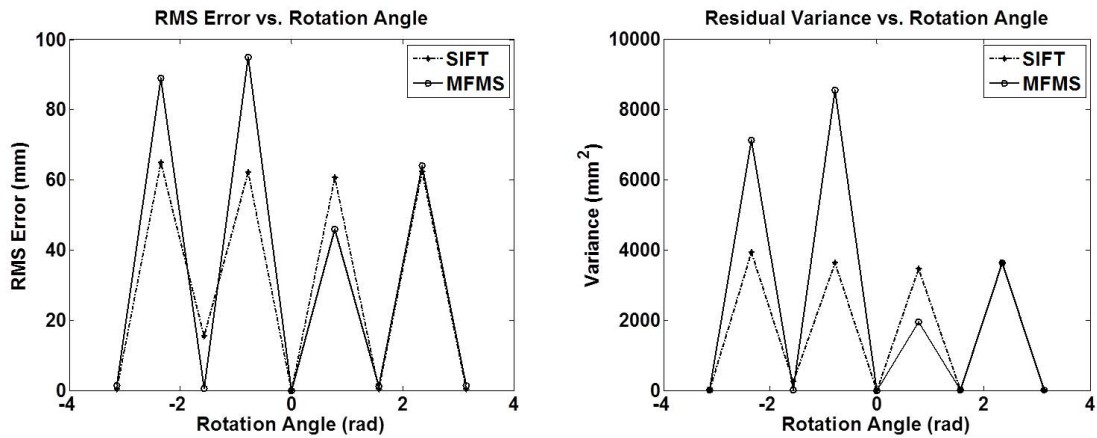


Fig. 79: Residual RMS and Variance values for the MFMS and SIFT algorithms vs. rotation.

ilar performance in feature extraction, matching, and initial motion estimate accuracy. More importantly the residual statistics describing the matching performance are similar and consistent. The next scene represents a simple extension of the scale space methods to data which is not fully populated. Since the SIFT algorithm was designed for image data, it does not distinguish between pixel which are associated with a measured object of interest, and pixels that represent the background. This fact will be important when selecting features to track relative motion between spacecraft in proximity operations.

2. Space Object Image

This experiment extends the analysis performed in the previous sub-section for a fully populated surface to a partial surface associated with an object which only takes up a portion of the field of view of the LIDAR. The asteroid image is again used to texture a constant range surface, however, only non-zero texture values are used, if a texture value is a zero, the associated range measurement is removed. This results in an object with boundaries and holes which presents a difficulty for image based methods which can not distinguish between holes in surfaces and shadows for example. In this case, the Gaussian convolutions of each method are identical only in areas where the entire operator is being applied to a fully-populated region of data . In locations where this is not the case the scale space construction used in this dissertation adapts to account for the missing measurements. The resulting feature sets differ in this case due to the necessity of localizing each feature with a measured location on the surface. The SIFT technique often results in feature points which are not localized on the surface but in the black space exterior to the object. These feature locations are most likely associated with depth discontinuities and are therefore unstable. Just like the Graffiti image scene, the Asteroid scene was tested

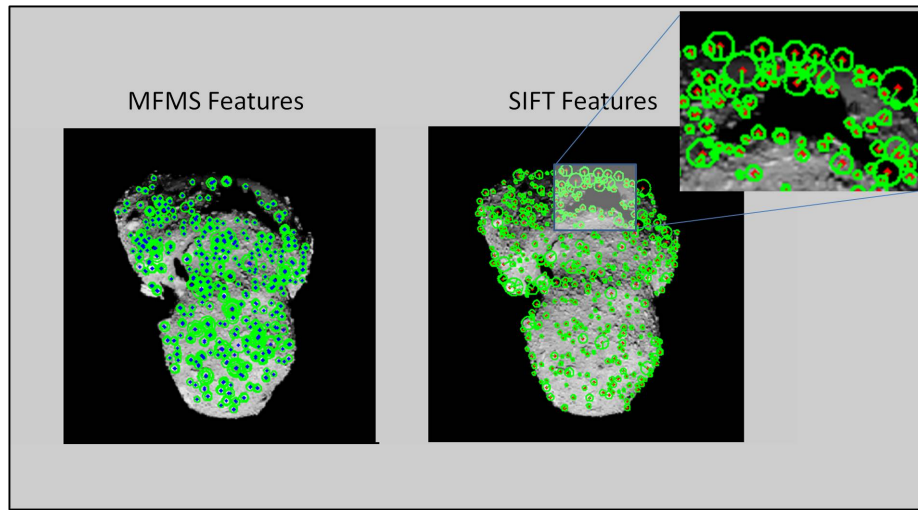


Fig. 80: Scale Space Feature Extraction on the Asteroid image texture field for the Scale Invariant Feature Transform and Diffusion Flow techniques. Notice that for the SIFT technique, many features are extracted in locations which do not reside on the measured/ object.

with respect to variations in texture based noise, lighting, and rotations about the LIDAR bore sight. Figure 80 shows an example feature extraction on the asteroid scene for the Multi-Field Multi-Scale features and the SIFT features.

The following plots and tables report the performance of both methods with respect to the above variations in the data and demonstrate the gain in performance with respect to 3D motion estimation afforded by using a scale space technique based on object space measurements. Similar to the Graffiti image, the number of matches decreases with increasing noise in the texture down to a level of 70% feature matches at 8% noise. Figure 80 shows the features extracted by each method, notice the large number of features extracted on external edges by the SIFT method, these locations can be very dangerous to use when estimating motion by 3D correspondences. These points are in locations of large depth discontinuities, and in the worse case can be localized in locations where no measurements are made at all. Including large numbers of these features in the least squares estimate of the relative motion can cause

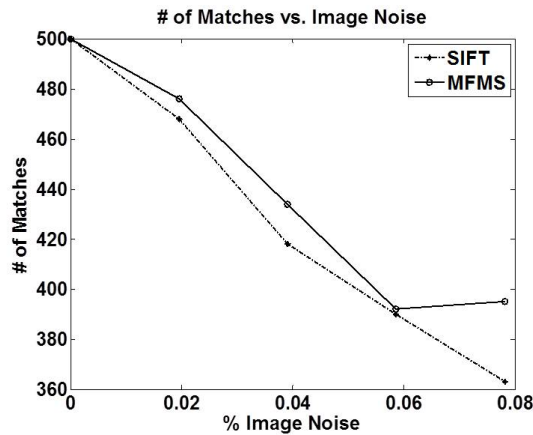


Fig. 81: The image noise is varied from zero to 8 percent causing a consistent decrease in the number of matches.

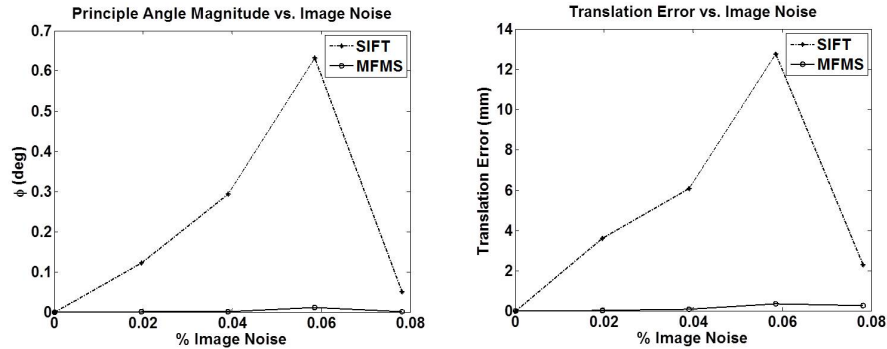


Fig. 82: Rotational and Translational accuracy vs. image noise for the MFMS and SIFT algorithms.

significant errors.

The MFMS approach is much more accurate in rotation and translation than then SIFT approach. While both achieve accuracies less than 1 degree in rotation and 13 millimeters in translations, the MFMS approach is more accurate by an order of magnitude. This is even more apparent when we look at the residual statistics. For the MFMS approach, the root mean square error of the match residuals has a very similar behavior for both the Graffiti scene, and the Asteroid scene. The SIFT method however has a significantly degraded performance for the asteroid scene. The

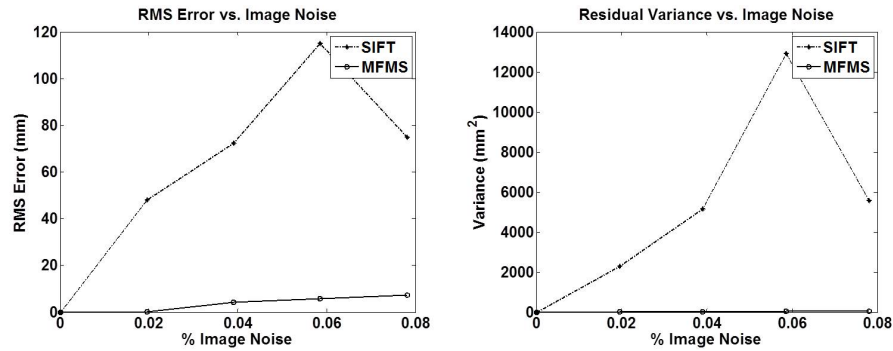


Fig. 83: Residual RMS and Variance vs. image noise for the MFMS and SIFT algorithms.

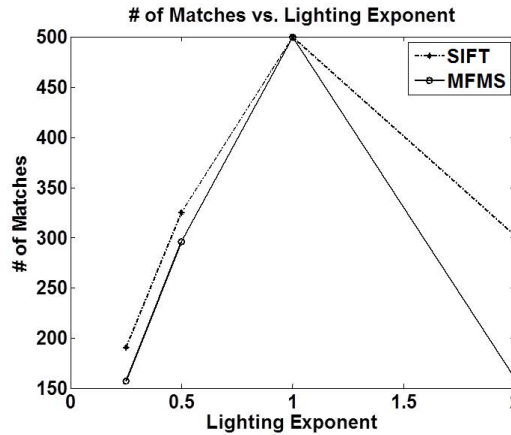


Fig. 84: Number of matches reported for the MFMS and SIFT algorithms vs. lighting variations.

combination of noise with a large number of features located at depth discontinuities makes it difficult to be confident in the estimated motion. The same behavior is observed in the cases of lighting variations and in plane rotations.

For this more complex scenario, the performance of each algorithm as measured by the number of matches retained is more similar with respect to lighting conditions than the Graffiti image. As the lighting exponent is varied for this scene, the boundary between the object's outer edge and the background becomes more pronounced, resulting in more extracted features along depth discontinuities.

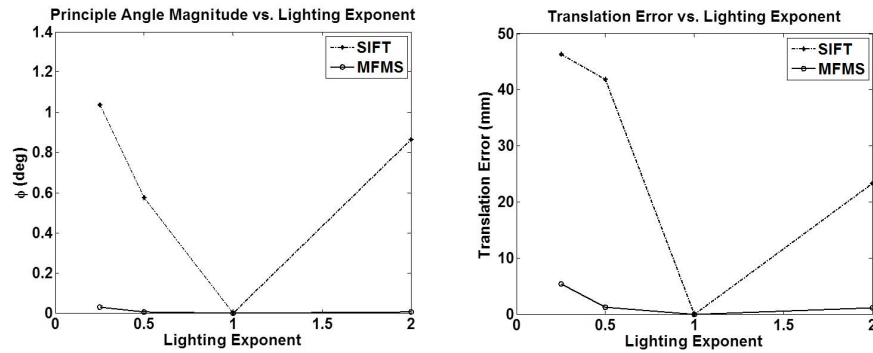


Fig. 85: Rotational and translational accuracies for the MFMS and SIFT algorithms vs. lighting variations.

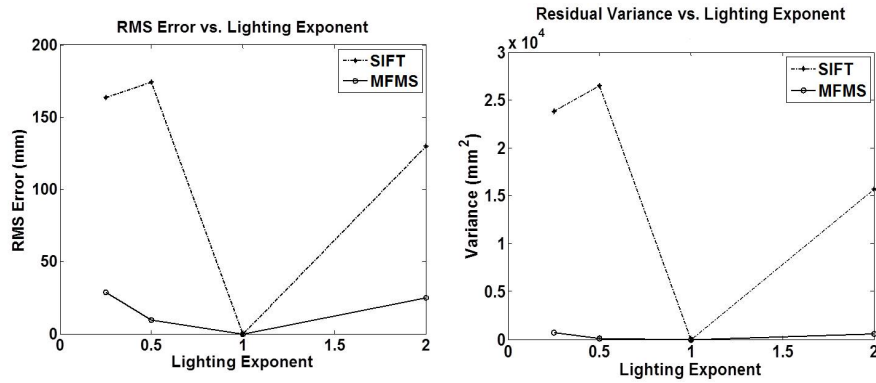


Fig. 86: Residual RMS and Variance values for the MFMS and SIFT algorithms vs. lighting variations.

In this case, the SIFT algorithm achieves results that are accurate to 1 degree and tens of millimeters while the MFMS approach is still achieving motion estimates consistent with performance levels achieved in the first scene. This fact is further illustrated by inspecting the residual statistics. The performance as evaluated by the statistics of the OLTAE residuals shows that the performance of the SIFT algorithm has degraded by a factor of 3 in the RMS error, while the MFMS approach is approximately achieving the same level of performance.

In the case of in-plane rotation, the MFMS algorithm again drops to around 20% matching in the extracted features for rotation angles that are odd multiples of $\frac{\pi}{4}$.

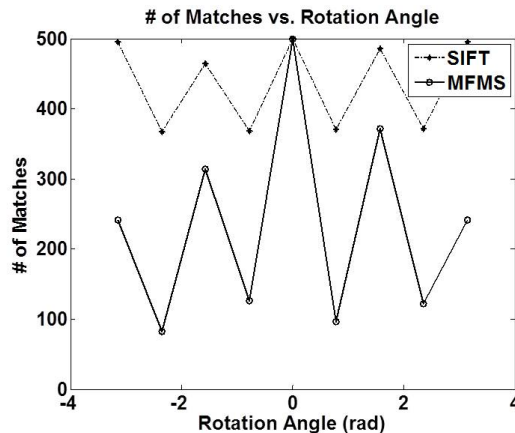


Fig. 87: Number of matches reported for the MFMS and SIFT algorithms vs. rotation.

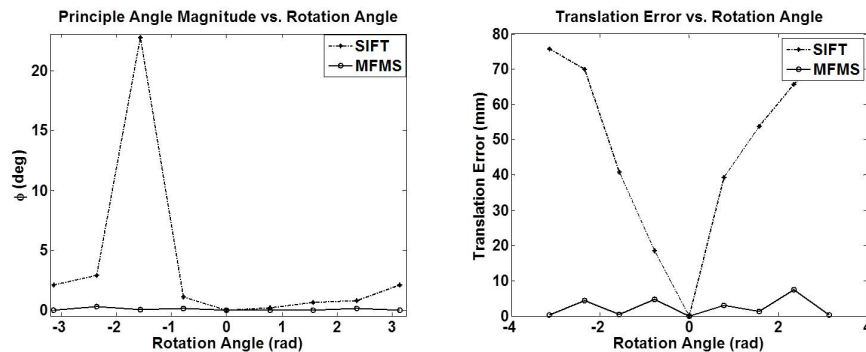


Fig. 88: Rotational and translational accuracies for the MFMS and SIFT algorithms vs. rotation.

A similar behavior is seen in the number of matches for the SIFT algorithm, which consistently reports between 60% and 90% of the features as accurately matched. Unlike the in-plane rotation case for the fully populated Graffiti scene, the SIFT algorithm does not accurately register the scenes.

While the MFMS algorithm achieves a consistent performance and accounts for the locations where data is missing, the SIFT algorithm does not and therefore is subject to a higher number of mismatches with more severe effects on the resulting motion estimate. Again the residual statistics indicate that there should be low confidence in the motion estimates provided by this method.

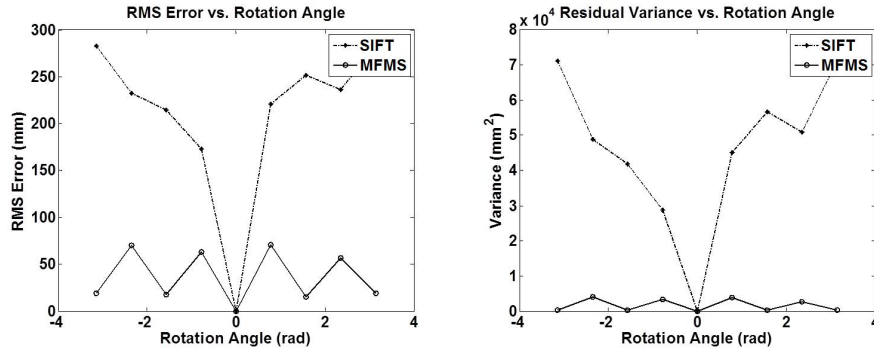


Fig. 89: Residual RMS and Variance values for the MFMS and SIFT algorithms vs. rotation.

The RMS error and variance of the residuals show a distinct difference between MFMS feature extraction and matching, and SIFT feature extraction and matching. MFMS feature extraction requires extrema to lie in regions that are at least continuous to second order, which means they must at a minimum be at the center of a fully populated 5x5 region of measurements. While the SIFT algorithm could be adapted to include this constraint, which would result in a significant increase in performance for this application, it still does not adapt the convolution kernel to account for variations in the object space and therefore does not distinguish which edges are associated with depth discontinuities. This fact is very apparent when observing the extracted features on the HST model.

3. Multi-Field SIFT vs. MFMS on the Hubble Space Telescope Model

An even more complex scenario is used to demonstrate the benefit of extending scale space methods to multi-field data. The HST model used in the circumnavigation experiment of the next section is first scanned and imaged in a single forward facing orientation. The top 100 features from each field in the MFMS framework are used to evaluate the performance of feature extraction and matching with respect to the same variations compared in the previous two scenes. The SIFT algorithm is applied

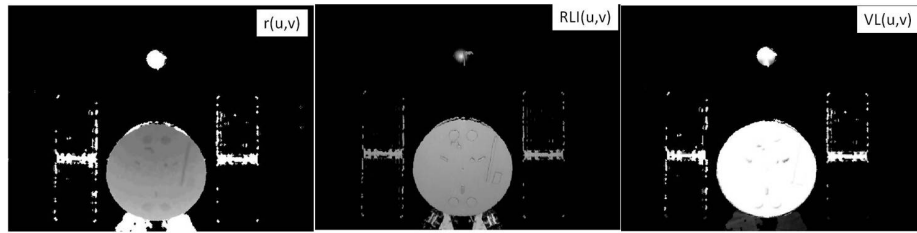


Fig. 90: Range, Return Light Intensity, and Visible Light Fields for the HST model.

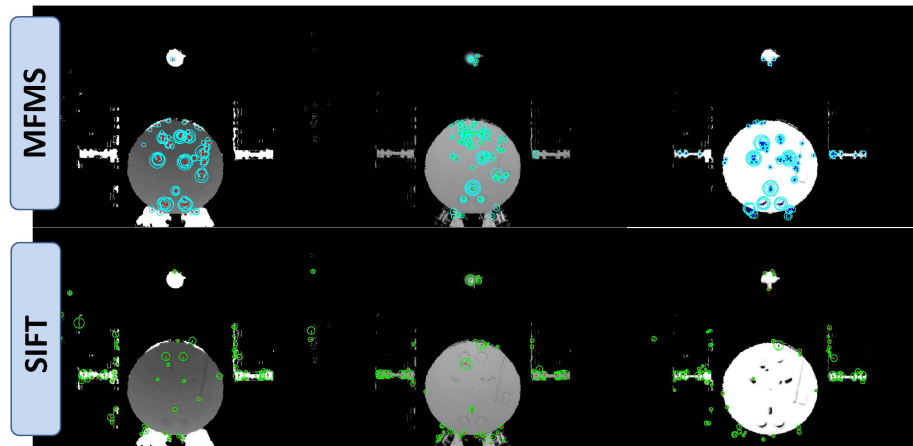


Fig. 91: Extracted features for each of the 3 Fields for the HST model.

individually to each field in the data set, and the top 100 features from each field are used as well to estimate motion. Again, the desired motion estimate is an identity rotation and zero translation. The variations are only $2D$ in nature in that the HST model was not moved, only transformations were applied to the texture values, and the spatial locations of the measurements in the image array. Figure, 90 shows the three fields supplied to each algorithm for feature extraction.

In the cases of texture based noise and lighting, only the texture fields are transformed by adding noise or raising the field value to a power respectively. In the case of rotation, the $3D$ point locations are preserved, only the image space locations are changed. Figure, 91 shows the extracted features in each field for each algorithm.

It is important to note that the MFMS feature extraction results in a larger

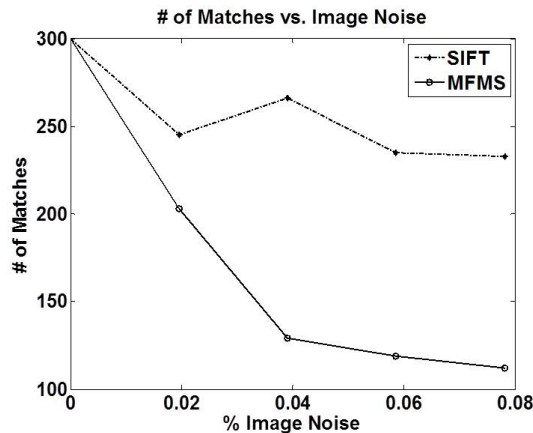


Fig. 92: The noise in each texture field is varied from zero to 8 percent causing a consistent decrease in the number of matches.

number of features that are located on object structures that are not also associated with large depth discontinuities. Many of the extracted SIFT features are in locations that are associated with depth discontinuities as well as locations that are not part of the object structure at all. Also, the HST model is not a constant range surface. It is a complex object with many different structures, repeating patterns, symmetry, and various surface directions. For this reason the penalty associated with different mismatches can be much more severe than in the constant range surface case. These facts explain the overall drop in the accuracies of the initial motion estimates, and motivate the need for an effective post filtering process based on match residual statistics.

The MFMS approach again retains fewer matches than the SIFT algorithm down to approximately 35% of the extracted features for 8% noise in each of the texture fields. The extracted and matched features in this experiment also appear to achieve better accuracy in the initial motion estimate.

Both algorithms appear to be accurate to better than 0.5 degrees and 25 millimeters but the associated points used in the SIFT algorithm were localized in areas near

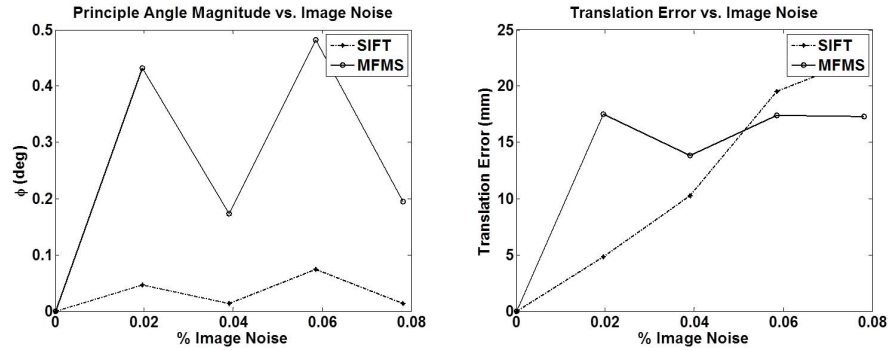


Fig. 93: Rotational and Translational accuracy vs. image noise for the MFMS and SIFT algorithms.

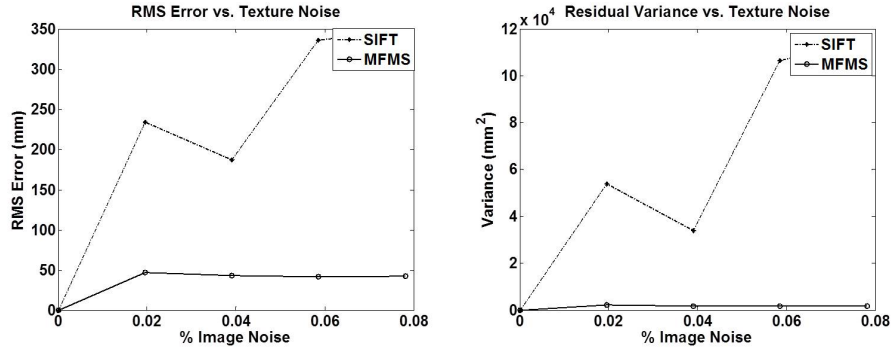


Fig. 94: Residual RMS and Variance vs. image noise for the MFMS and SIFT algorithms.

depth discontinuities and data drop outs which corrupts the solution. The confidence in the solution is summarized by again inspecting the OLTAE residual statistics.

The OLTAE residual RMS error is a good indicator of the quality of the motion estimate. While the SIFT algorithm appears to be more accurate given the magnitude of the principle angle and translation vector, the mean localization error provided by the MFMS algorithms is as much as seven times more accurate than the SIFT algorithm. This indicates a high number of mismatches and poorly localized correspondences. A similar result is observed when evaluating the algorithms for variations in lighting and in plane rotation angle.

The number of retained matches used in the initial motion estimate is very

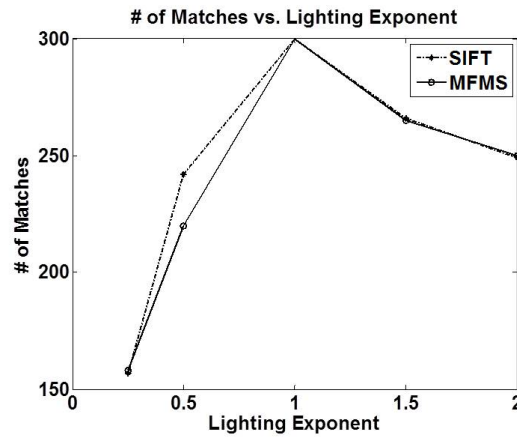


Fig. 95: Number of matches reported for the MFMS and SIFT algorithms vs. lighting variations.

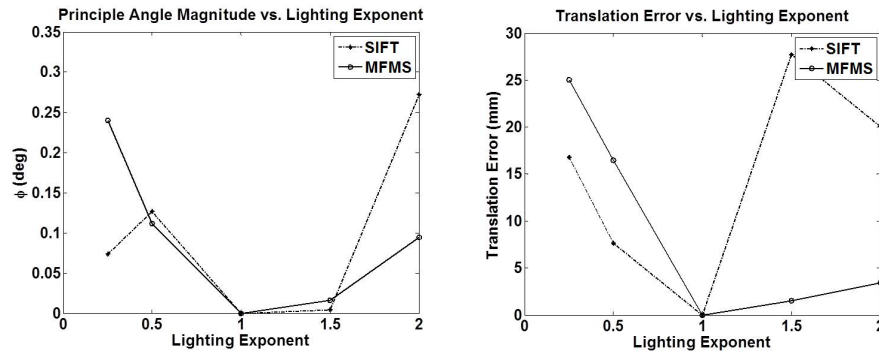


Fig. 96: Rotational and translational accuracies for the MFMS and SIFT algorithms vs. lighting variations.

similar for all lighting exponents in this test. Both algorithms achieve a similar level of accuracy with respect to the tested lighting variations, and the order of magnitude is similar to the case of texture based noise. The MFMS algorithm again achieves a consistent level of confidence to the case where noise was added to the texture fields and in this case is as many as two orders of magnitude better than the SIFT solution.

When testing each algorithm's performance with respect to in-plane rotation, we again see a small percentage of features used to obtain the initial motion estimate for the MFMS algorithm. The SIFT algorithm has significant variation in the number

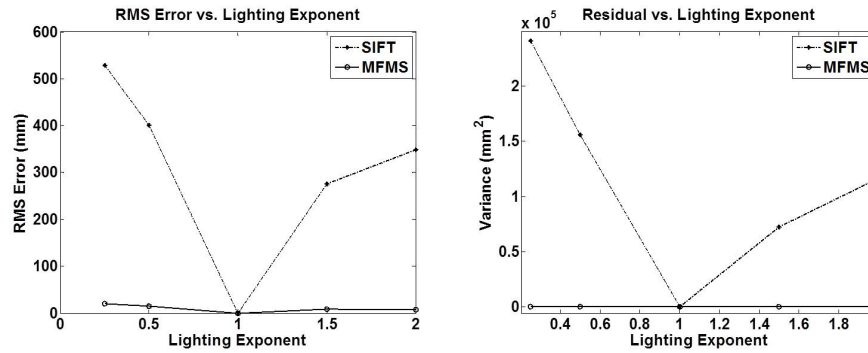


Fig. 97: Residual RMS and Variance values for the MFMS and SIFT algorithms vs. lighting variations.

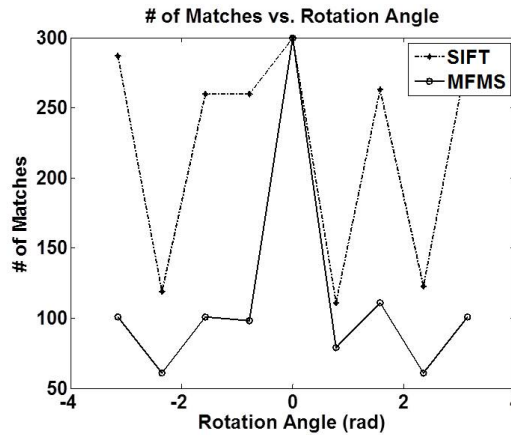


Fig. 98: Number of matches reported for the MFMS and SIFT algorithms vs. rotation.

of features used to obtain its motion estimate.

Both algorithms achieve accuracies better than 3 degrees and 263 millimeters while the MFMS algorithm appears to be slightly more consistent. The consistency of the MFMS approach is further supported by the OLTAE residual statistics which are of the same order of magnitude of the graffiti image and asteroid image residual statistics. Of further interest is the consistent behavior of the residual statistics in all MFMS cases where odd multiples of $\frac{\pi}{4}$ are slightly less accurate and less consistent than other rotation angles.

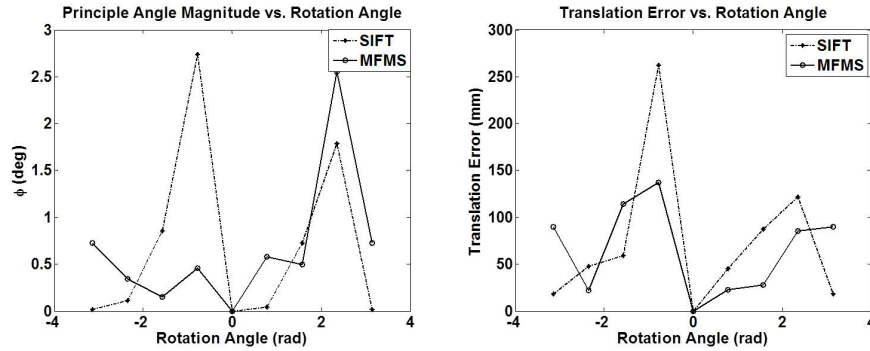


Fig. 99: Rotational and translational accuracies for the MFMS and SIFT algorithms vs. rotation.

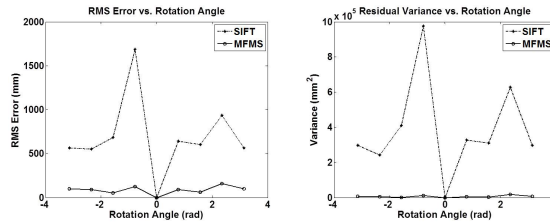


Fig. 100: Residual RMS and Variance values for the MFMS and SIFT algorithms vs. rotation.

Even for these simple variations on a more realistic *3D* object, traditional scale invariant feature detection and matching proves to be not as robust as the Multi-Field Multi-Scale feature detection and matching approach. A more complex scenario associated with proximity operations involves out of plane rotation resulting in local sampling variations associated with *3D* transformations of the relative pose as well as textural variations associated with different materials, lighting, and observation conditions. The next section demonstrates the application of MFMS features and Diffusion Flow Signatures to a simple but realistic proximity operations experiment.

B. HST Model Case Study

The Hubble Space Telescope model is a *1/10th* scale model built to simulate real photometric and spatial variations of *3D* space objects in controlled lighting situations

in the Land Air and Space Laboratory at Texas A& M University. It is used as a challenge object to test vision algorithms for spacecraft proximity operations and vision aided guidance. To date, most structure from motion algorithms based on vision alone have failed to autonomously obtain accurate reliable motion estimates and mappings of the complex structures of the HST model. The partial circumnavigation experiment is used here to demonstrate the practical application of the methods of this dissertation to real data in a scene which physically represents many of the challenges experienced in tracking spacecraft in close range proximity operations.

1. Experimental Setup

The partial circumnavigation experiment consists of 28 observations of the HST model taken with a static, calibrated, camera-LIDAR fusion system. Through each of the observations, the HST model is rotated in increments of 2 deg about the vertical axis of a Stewart platform located on the Holonomic Omni-directional Motion Emulation Robot (HOMER). With respect to the Z axis of the LIDAR, the forward facing panel of the HST processes through incidence angles from -30 deg to 30 deg. Figure 76 shows this simple out of plane relative motion with respect to the LIDAR along with the relative placement of the camera and point light source used to produce the visible lighting effects observed.

The Multi-Field Multi-Scale feature extraction and matching process is used to register the observations, estimate the relative motion, and construct a combined model of the Hubble Space Telescope with zero a-priori knowledge. The results of the MFMS approach with and without a post filtering process based on the residuals of the Optimal Linear Translation and Attitude Estimator algorithm are compared to the results obtained using Scale Invariant Feature based registration. Figures 102 to 105 show the variations in all three fields for all of the scans used in the partial

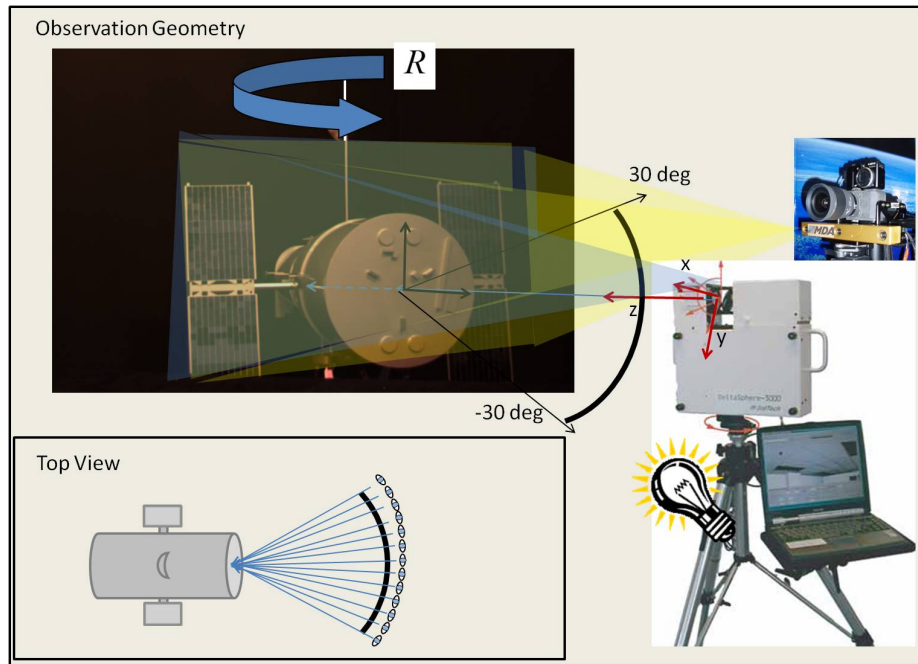


Fig. 101: Experimental setup for the HST partial circumnavigation experiment. The spacecraft is rotated in precise 2deg increments in front of a Camera and LIDAR which make simultaneous observations.

circumnavigation experiment.

Significant variation is observed in the measured structure of the HST model as well as in the Return Light Intensity and Visible Light texture fields. In the case of the Visible Light, significant and sudden changes in lighting conditions are measured as values on the front facing plate go from medium gray scale values to near saturation and back. The measured structure of the HST is asymmetric for large incidence angles, where only the closer solar panel is measured, and symmetric for small incidence angles. Specular reflections are also observed in both textures, especially on the solar panel surfaces. Each algorithm is tested by using feature matches to specify 3D correspondences which are inputs to the OLTAE motion estimation algorithm. Each textured scan is mapped back to the first scan in the series, in this case, the -30 degree pose, and the resulting fused model is inspected for accuracy.

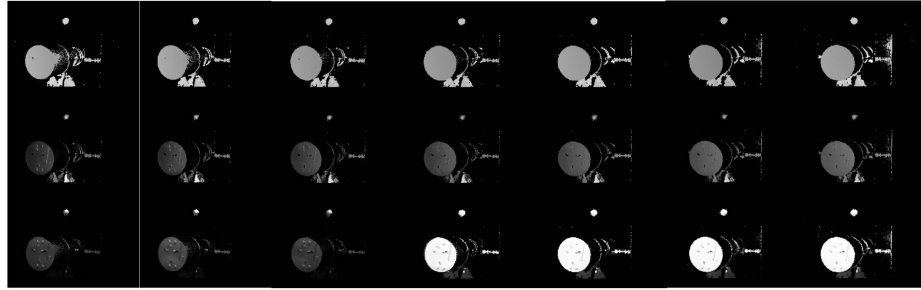


Fig. 102: Top: Range field information for scans 1-7 in the partial circumnavigation experiment. Middle: Return Light Intensity field information. Bottom: Visible Light field information obtained from the camera.

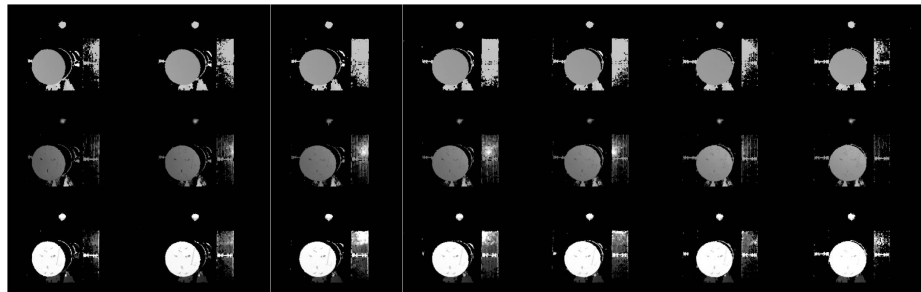


Fig. 103: Top: Range field information for scans 8-14 in the partial circumnavigation experiment. Middle: Return Light Intensity field information. Bottom: Visible Light field information obtained from the camera.

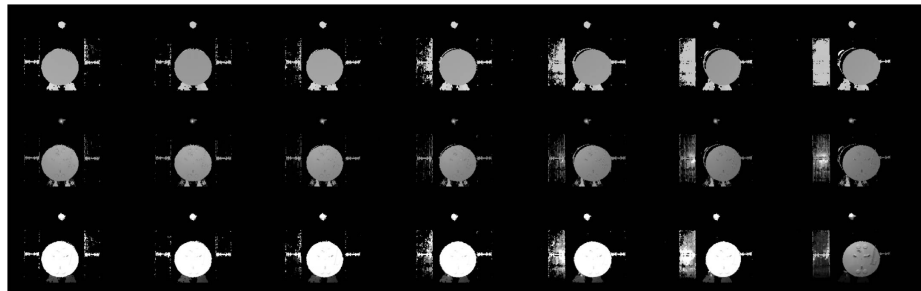


Fig. 104: Top: Range field information for scans 15-21 in the partial circumnavigation experiment. Middle: Return Light Intensity field information. Bottom: Visible Light field information obtained from the camera.

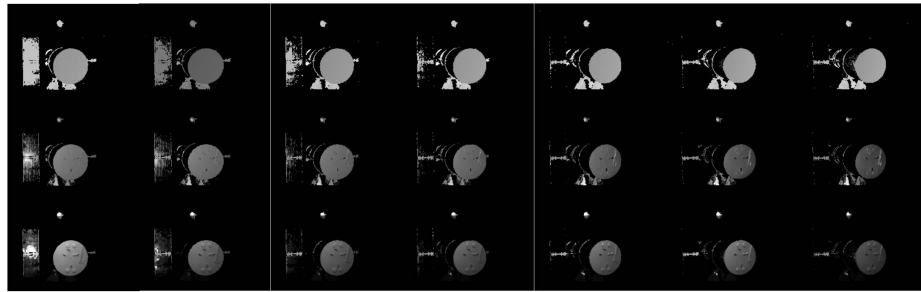


Fig. 105: Top: Range field information for scans 22-28 in the partial circumnavigation experiment. Middle: Return Light Intensity field information. Bottom: Visible Light field information obtained from the camera.

The metrics used to indicate the quality of each individual motion estimate are the RMS error and variance of the OLTAE residuals. The following sub-sections illustrate the performance of each algorithm.

2. Registration Using Scale Invariant Feature Transform

The Scale Invariant Feature Transform is used to extract features from the textured scan data and the associated 3D correspondences are used as input to the OLTAE motion estimation algorithm. The Root Mean Square error of the residuals is reported in figure 106.

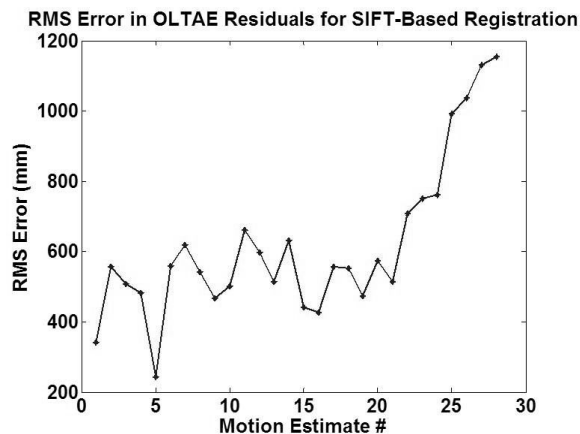


Fig. 106: RMS Error for OLTAE residuals for SIFT matches used to obtain motion estimates.

The mean RMS error over all motion estimates is 618.0792 millimeters which indicates poor registration performance. The locations of the extracted features are similar to the case of the static HST test, resulting in very inconsistent feature extraction and matching.

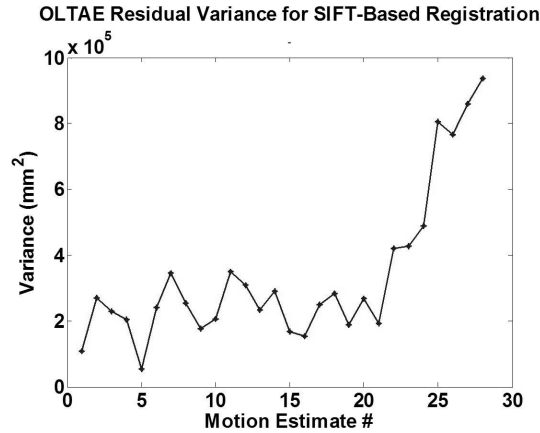


Fig. 107: Variance of OLTAE residual values of the SIFT matches used to estimate the relative motion.

The variance of the OLTAE residuals is also very high for all motion estimates which is a good indicator of poor registration performance. This fact is very clear when the resulting model consisting of all 28 scans is inspected.

It is difficult to determine if any of the individual models are accurately registered, and the estimated relative trajectory which should be a circular arc appears random. Directly applying the Scale Invariant Feature Transform to each field and determining the closest measured 3D point for each feature does not result in consistent, accurately localized correspondences, and as a result, the motion estimation algorithm fails.

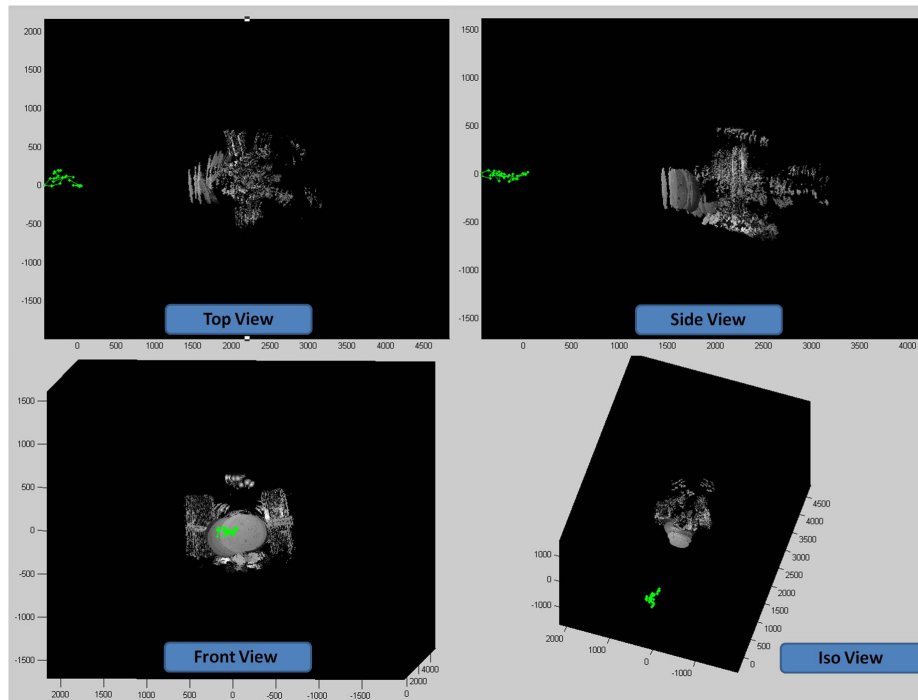


Fig. 108: The resulting registration of 28 textured scans using matched SIFT features.

3. Registration Using Multi-Field Multi-Scale Features

The Multi-Field Multi-Scale feature detection and matching framework is used to determine 3D correspondences which are well defined on the surface of the HST model. The OLTAE algorithm is again used to estimate the relative motion and register each of the 28 individual textured scans.

The mean RMS value over all motion estimates is 52.1808 which is an order of magnitude better than the result obtained using SIFT correspondences and consistent with the algorithms performance for all other experiments demonstrated in this dissertation. Motion Estimate 21 has a maximum RMS error value of 140.25 which indicates that for that particular scene the solution is degraded due to poorly localized features or large residual mismatches. If this scene is simply rejected from the scan sequence, and the motion estimates and registered model are recomputed, we

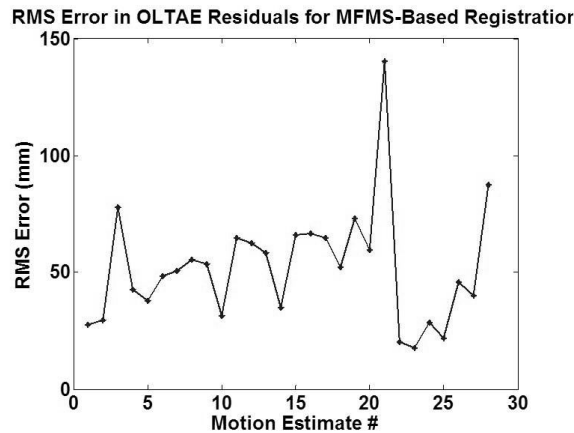


Fig. 109: RMS Error for OLTAE residuals for MFMS matches used to obtain motion estimates.

obtain improved performance. The result of this process is demonstrated at the end of the next section which results in the final model.

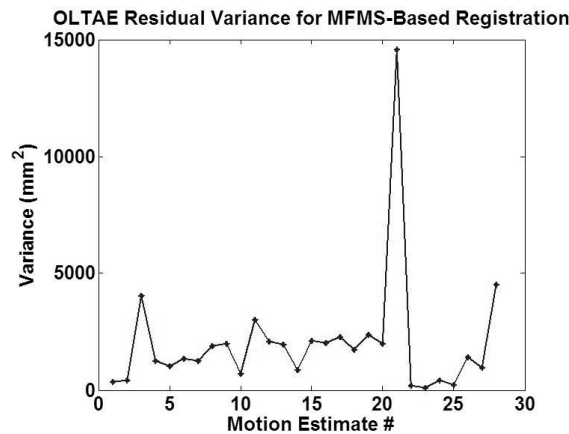


Fig. 110: Variance of OLTAE residual values of the MFMS matches used to estimate the relative motion.

The model obtained by registering MFMS features and using only the initial OLTAE motion estimates is shown in figure 111. Notice that the relative motion trajectory is an arc as expected, but some scans are registered improperly due to feature mismatches. This process can be improved by employing the OLTAE residual post filtering described in chapter 6.

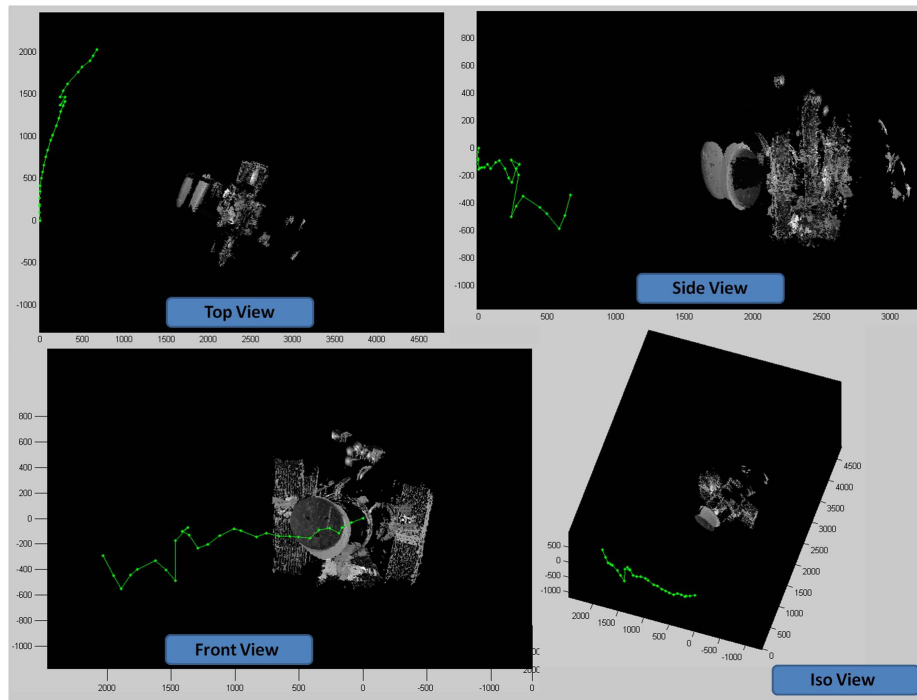


Fig. 111: The resulting registration of 28 textured scans using matched MFMS features.

4. Registration using MFMS Features and OLTAE Residual Filtering

Again, the Multi-Field Multi-Scale feature detection framework is used to determine 3D the correspondences that are provided to the OLTAE motion estimator. Additionally, the residual post filtering process of chapter 6 is used to refine each individual motion estimate and prevent gross errors caused by feature mismatching.

The resulting RMS error is again improved by an order of magnitude with a mean value of 6.0532 millimeters over all motion estimates. Candidate scans for rejection can be easily identified by inspecting the reported variance in the OLTAE residuals of each motion estimate.

For this experiment the model is expected to improve if the 21st model and motion estimate is rejected. The resulting improvement is summarized by figures 115 to 117.

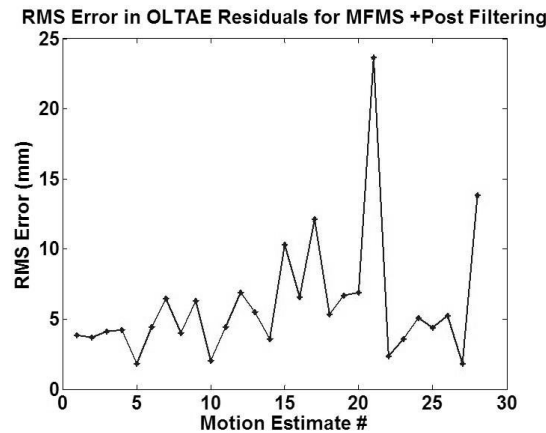


Fig. 112: RMS Error for OLTAE residuals for the post filtered MFMS matches used to obtain motion estimates.

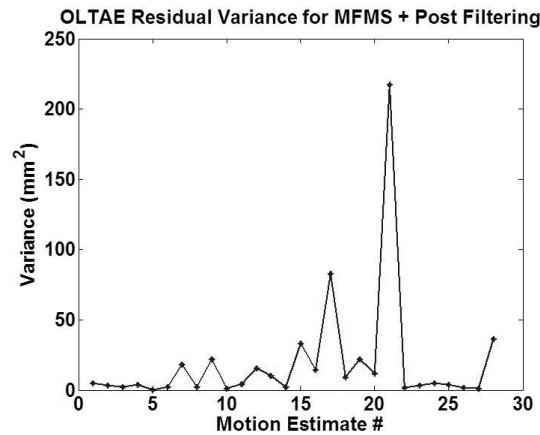


Fig. 113: Variance of OLTAE residual values of the post filtered matches used to estimate the relative motion.

The resulting combined model is significantly improved over feature matching alone and the estimated relative motion trajectory is a much more consistent arc.

The RMS errors for all motion estimates used to construct the final model are less than 12 millimeters. The reported accuracy of the Deltasphere 3000 lidar is 7 millimeters. It is encouraging to see that we are achieving registration results at accuracies comparable to the measurement accuracy of the scanning device.

The final reconstructed model using 27 of the 28 scans is shown in figure 117.

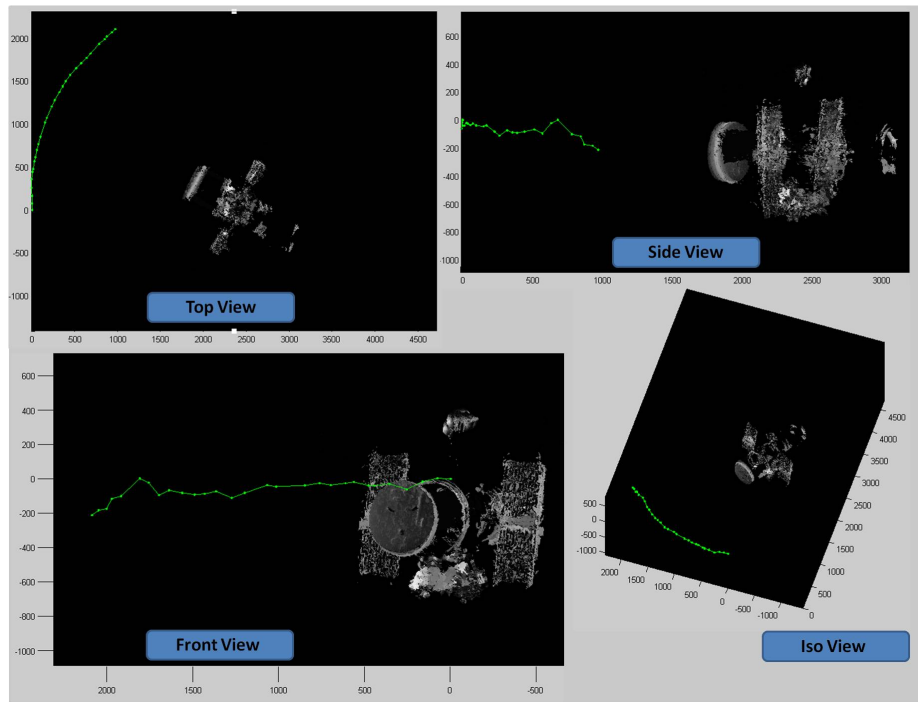


Fig. 114: The resulting registration of 28 textured scans using post filtered MFMS matches.

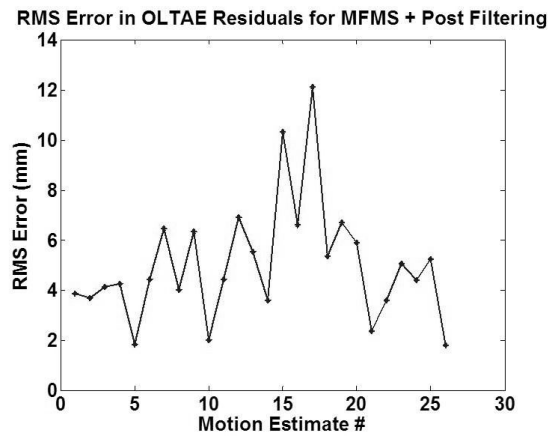


Fig. 115: RMS Error for OLTAE residuals for the post filtered MFMS matches used to obtain motion estimates.

The cylindrical structure of the body can easily be seen as well as the backward facing plate which was not directly observed. This relative motion estimation and object mapping performance is a key element in supporting autonomous spacecraft

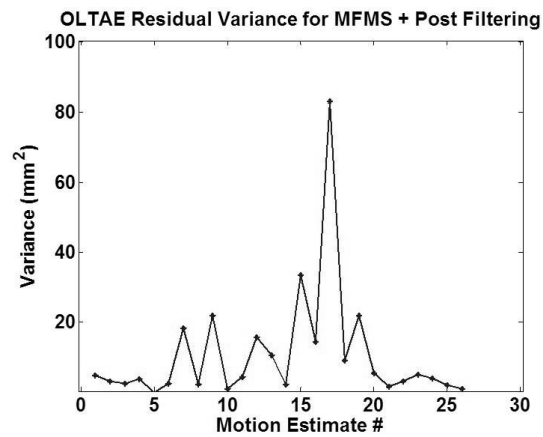


Fig. 116: Variance of OLTAE residual values of the post filtered matches used to estimate the relative motion.

proximity operations.

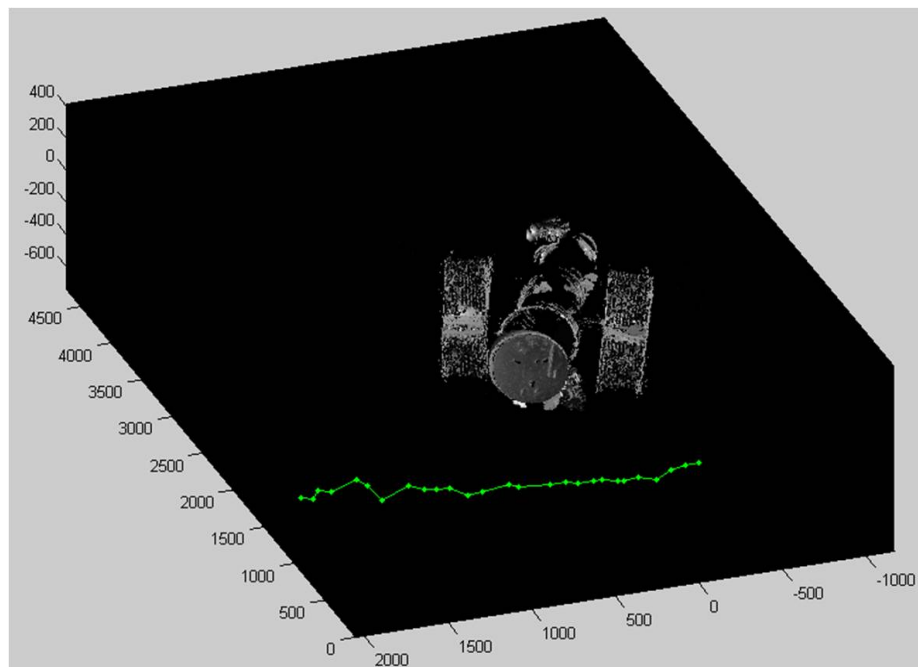


Fig. 117: The resulting registration of 27 textured scans using post filtered MFMS matches.

CHAPTER VIII

CONCLUSIONS AND FUTURE WORK

The methodology developed in this dissertation extends current techniques in scale-space image analysis to multi-field data and enables generalized *3D* feature extraction and tracking in spacecraft proximity operations. The resident Multi-Field Multi-Scale features extracted are local extrema in difference images computed between consecutive convolutions with a geodesic Gaussian defined in the object space. These features represent local maxima and minima in data curvature making them capable of describing geometrically interesting but texturally bland objects as well as texturally rich but geometrically simple objects. This capability is a key necessity of a feature tracker which is to be used to enable autonomous spacecraft proximity operations. The developed descriptor known as the Multi-Field Diffusion Flow Signature, was inspired by the Spin-Image technique [6, 7] used for the global and local description of regularized *3D* models, effectively produced accurately localized *3D* correspondences that were robust to *2D* and *3D* transformations of the data. Combined feature match rejection driven by the residuals produced by the Optimal Linear Translation and Attitude Estimator, MFMS feature extraction and Multi-Field Diffusion Flow Signatures were used to accurately track the relative motion of a realistic space object.

The final model was constructed using 27 partial scans sweeping through an angle of approximately 60 degrees using motion estimates which were all localized to within 12 millimeters RMS. The approach developed here was also compared to the state of the art Scale Invariant Feature Transform for the case where surface based and image based analysis are equivalent. Multi-Scale Multi-Field feature extraction and

matching achieved comparable performance to SIFT feature extraction and matching and outperformed the SIFT technique for scenes which included significant areas of zero measurement.

A. Conclusions

This dissertation sought to answer four main scientific questions:

By what data driven process can features be extracted from textured range imagery and tracked through general relative motion?

By extending scale space techniques to apply to measured manifolds using a discrete approximation to the Geodesic Gaussian kernel, it is possible to combine surface evolution by mean curvature flow with the analysis of texture by extracting features as extrema in the Difference of Geodesic Gaussian Space. These extrema are maxima and minima in diffusion flow velocity, which for a range image describe locations of highest curvature, and for a texture field describe corner like features much like the standard techniques of scale space feature extraction in imagery. By using Diffusion Flow Signatures to describe the extracted point features, the descriptor is based on geometric and textural information and is populated in a similar way as Spin-Images. Diffusion Flow Signatures do not depend on an accurate estimate of the surface normal using local information which can make the population of the descriptor inconsistent. Combining both of these techniques for feature extraction and description with the matching process which weights the quality of each field's contribution to the descriptor and using the Optimal Linear Translation and Attitude Estimator to obtain relative motion estimates, results in well conditioned sets of 3D correspondences can be used and post-filtered to obtain accurate motion solutions even in the presence of field variations and sampling variations. A byproduct of the

statistical analysis of OLTAE residuals is a covariance estimate of (i) the model and (ii) the translation and rotation coordinates.

How can local geometry and texture be combined to allow point based descriptors to enable rigorously linear six degree of freedom relative motion estimation?

The combination of Multi-Field Multi-Scale feature extraction and Diffusion Flow Signatures has been shown to consistently provide a robust set of object space 3D correspondences which provide a well conditioned least squares problem to the Optimal Linear Translation and Attitude Estimator. Using OLTAE to determine the relative motion results in rigorously linear estimates of the parameters describing the relative rotation and translation of the sensor system and observed object. Furthermore, the residual statistics provided by the OLTAE algorithm are an effective tool which can be used to monitor the confidence in the obtained motion solution as well as reject scans in the worst case where motion can not be estimated to specified accuracy requirements. An important aspect of this approach proved to be the robustness of the multi-field approach which was able to provide useful features to track even in the presence of discontinuous surfaces, different materials, and extreme variations in lighting conditions. For most state of the art vision techniques, lighting is still a problem and surfaces are expected in the algorithm assumptions to have Lambertian behavior. This is not a requirement of the methods presented here.

Does the addition of direct range measurement allow for significant increases in precision or robustness?

Yes, direct range measurements, if incorporated into the construction of the scale space allow for the information regarding the quality of the sampling along the surface to drive the feature extraction process. This allows a for the identification of features which reside on meaningful structure on the measured object and can be

effective in preventing the extraction of features which reside near significant depth discontinuities. By knowing the range to extracted multi-scale features it is easy to define an object space scale which must be closely matched in order to correctly associate features. This is in contrast to Scale Invariant Features which are subject to matching features at any two dimensional scale. Finally, having the range to each extracted feature on the surface allows for simple and effective motion estimation to be based on $3D$ to $3D$ correspondences which reduces the number of required feature matches. This is important for spacecraft proximity operations where objects can have very few unique features.

Can the combined scale space analysis of 3D object space manifolds and associated texture along with the application of Multi-Field Diffusion Flow Signatures result in robust motion estimation for real data in proximity operations?

Yes, the methods developed in this dissertation have been shown to provide consistent performance on simulated and real data. The partial circumnavigation experiment demonstrated that multi-field object space feature extraction and matching is possible even in the presence of significant variations in lighting conditions. The use of combined geometric and textural scale space analysis proves to be a key tool in enabling autonomous spacecraft proximity operations. With zero *a priori* knowledge of the object to be measured, Multi-Field Multi-Scale feature extraction and description by Multi-Field Diffusion Flow Signatures were successfully applied to register 27 independent scans of a $1/10^{th}$ scale Hubble Space Telescope model with a maximum root mean square error of 12 millimeters. The mean RMS error was approximately 6 millimeters which is comparable to the accuracy of the LIDAR sensor itself. A close-up of the resulting final model is shown in figure 118.

The stated goal of this dissertation was to develop a feature detection and track-

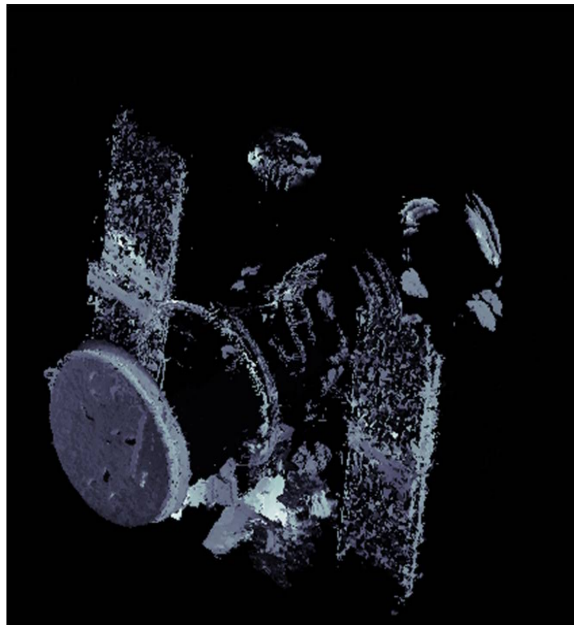


Fig. 118: The final model resulting from autonomous feature tracking using MFMS features and OLTAE residual post filtering.

ing approach which (i) localizes features based on uniqueness in multiple data sets, (ii) encodes the true feature scale(s), (iii) is robust to photometric variations, (iv) can find trackable locations in the absence of texture, and (v) consistently samples local geometry in the construction of its descriptor. Multi-Field Multi-Scale feature extraction does indeed localize features based on uniqueness in multiple data sets. In fact, it can easily be extended to include more than the three fields used in the examples of this dissertation. By generalizing the extrema detection process to identify the local minimum and maximum values of diffusion flow velocities the same approach was shown to find unique features which can have geometric or textural meaning, or both. The range was used to associate a measured object space scale to associate with each keypoint in order to significantly reduce the number of necessary feature comparisons performed to determine 3D point correspondences. In the partial circumnavigation experiment MFMS feature extraction and matching using

Diffusion Flow Signatures proved to be robust to significant photometric variations including in the visible light field near saturation conditions. The extracted features were localized frequently on meaningful object structure even when the textural information decreased as the image of the surface became quite bland, the range field still provided meaningful feature points. Finally the construction of the Diffusion Flow Signatures was not dependent on the local estimation of geometric properties, and therefore provided a consistent construction of a local descriptor which proved to be robust to variations in observation conditions.

This dissertation has made the following important contributions:

- A meshless approach to combined Scale Space analysis of LIDAR and camera measurements.
- A combined multi-field, multi-scale (MFMS) feature detection process which defines its descriptor in the object space and is robust to noise and lighting variations.
- Statistical characterization of the resulting feature's OLTAE residuals to select the most consistent and accurately localized set of multi-field multi-scale interest points.
- The connection between surface evolution by mean curvature flow and scale space analysis of images embedded on $2D$ manifolds is demonstrated and applied to raw $3D$ data.
- A software pipeline which allows interactive scale space analysis of range data and camera imagery as well as enables the visualization of the extracted features and constructed models.

- Experimental results on real data have demonstrated Multi-Field Multi-Scale features capability to enable simple examples of spacecraft proximity operations.

Also unique to this dissertation was the method developed to evaluate two dimensional scale-space feature extraction techniques in the object space using associated $3D$ correspondences and the Optimal Linear Translation and Attitude Estimator. Using this approach to evaluate feature extraction and matching techniques with the end goal of supplying well conditions sets of point correspondences to a motion estimator is an effective way to test developed detection methods, matching algorithms, feature localization, and robustness to several variations in the data. Scale space analysis of textured range data is complex and there are several design choices necessary in the design of an algorithm to be used to enable feature extraction and tracking for spacecraft proximity operations. The software package developed in this dissertation was intended allow a user to not only perform feature extraction and matching in a laboratory setting using a scanner and camera, but also to vary parameters and thresholds as well as interact with the constructed scale spaces and extracted features. Along with many other facets of my continuing work related to the extension of scale-space techniques to multi-sensor systems, this software will continue to be developed and improved.

B. Remaining Questions and Future Work

The extension of scale space techniques to multi-field $3D$ data is still a new and developing area of research. While much utility and potential for significant impact to be made in the area of simultaneous localization and mapping in the space environment has been demonstrated in this dissertation, there is still much more work to be done to develop efficient, autonomous relative navigation between a spacecraft

and an unknown object. While developing the methods used in this dissertation, many more research questions have been asked than answered. Can unknown objects be decomposed into feature based models? Are features alone enough to build an accurate reconstruction of observed objects? Is there an effective means to compress 3D model information that preserves quality features while simplifying texturally bland or geometrically simple regions? Can a more effective use of scale space techniques be designed which uses the known image space boundaries of an object to dictate the number of useful scales in a scale space? Given a set of detected features and their known characteristics can subsequent scale spaces be built which will have higher probability of sampling the object space at the same spatial frequencies given a predicted relative motion? Can more complex scanning geometries be effectively accommodated by this framework? To what degree can this process be accelerated by *a priori* knowledge? What is the path to parallelizing this approach given that the kernel is adaptive and data dependent?

Future efforts are underway and are focused on increasing the efficiency of the current framework, developing a C based implementation as well as extending the applicability of this approach to other sensors including infra-red cameras, more complex scanning devices, and recent 3-D cameras such as the Xbox Kinect sensor and Remote Ultra Low Light Imaging (RULLI) which will be used for rapid prototyping and demonstration of current and future developments on small spacecraft models. To achieve effective object tracking and mapping with unknown proximal bodies through truly unrestricted relative motions, all of the above questions need to be addressed. A solution which further extends the techniques presented in this dissertation to an effective operational solution running on a space based platform will be a key contribution towards the end goal of achieving successful autonomous spacecraft proximity operations. Adding these issues will be the focus of my continuing work.

REFERENCES

- [1] Weismuller, T. and Leinz, M., “GN&C Technology Demonstrated By The Orbital Express Autonomous Rendezvous And Capture Sensor System,” *Proceedings of the 29th Annual AAS Guidance And Control Conference*, Breckenridge, CO, 2006, pp. 6–16.
- [2] Lowe, D. G., “Distinctive Image Features from Scale-Invariant Keypoints,” *International Journal of Computer Vision*, Vol. 60, No. 2, 2004, pp. 91–110.
- [3] Lowe, D. G., “Object Recognition from Local Scale-Invariant Features,” *ICCV*, 1999, pp. 1150–1157.
- [4] Lo, T. and Siebert, J., “Local Feature Extraction and Matching on Range Images: 2.5D SIFT,” *Computer Vision and Image Understanding*, Vol. 113, No. 12, 2009, pp. 1235–1250.
- [5] Lo, T. and Siebert, J., “SIFT Keypoint Descriptors for Range Image Analysis,” *Annals of the BMVA*, Vol. 2008, No. 3, 2008, pp. 1–18.
- [6] Johnson, A., *Spin-Images: A Representation for 3-D Surface Matching*, Ph.D. thesis, Robotics Institute, Carnegie Mellon University, Pittsburgh, PA, August 1997.
- [7] Johnson, A. and Hebert, M., “Using spin images for efficient object recognition in cluttered 3D scenes,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 21, No. 5, May 1999, pp. 433 – 449.

- [8] Brusco, N., Andreetto, M., Giorgi, A., and Cortelazzo, G. M., "3D Registration by Textured Spin-Images," *5th International Conference on 3D Digital Imageing and Modeling*, IEEE, 2005, pp. 262–269.
- [9] Dinh, H. Q. and Kropac, S., "Multi-Resolution Spin-Images," *IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, Vol. 1, IEEE Computer Society, Los Alamitos, CA, USA, 2006, pp. 863–870.
- [10] Mortari, D., Rojas, J. M., and Junkins, J. L., "Attitude and Position Estimation from Vector Observations," *Proceedings of the American Astronautical Society (AAS) Space Flight Mechanics Meeting*, Citeseer, Maui, HI, 2004, pp. 04–140.
- [11] Ho, H. T. and Gibbins, D., "Multi-scale Feature Extraction for 3D Models Using Local Surface Curvature," *Digital Image Computing Techniques and Applications*, IEEE, Canberra, Australia, 2008, pp. 16–23.
- [12] Woffinden, D. and Geller, D., "Navigating the Road to Autonomous Orbital Rendezvous," *Journal of Spacecraft and Rockets*, Vol. 44, No. 4, 2007.
- [13] Do Carmo, M. P., "Differential Geometry of Curves and Surfaces," *Nature*, Vol. 169, No. 4301, 1976, pp. 560–560.
- [14] Besl, P. J. and Jain, R. C., "Invariant surface characteristics for 3D object recognition in range images," *Comput. Vision Graph. Image Process.*, Vol. 33, January 1986, pp. 33–80.
- [15] Besl, P. J. and Jain, R. C., "Three-dimensional object recognition," *ACM Comput. Surv.*, Vol. 17, March 1985, pp. 75–145.

- [16] Ittner, D. J. and Jain, A. K., “3-D surface discrimination from local curvature measures,” *Proceedings of the IEEE Computer Society Conference on Computer Vision and Pattern Recognition*, San Francisco, CA, 1985, pp. pp. 119 – 123.
- [17] Moravec, H. P., “Rover Visual Obstacle Avoidance,” *Proceedings of the 7th International Joint Confernece on Artificial Intelligence*, Vol. 2, 1981, pp. 785–790.
- [18] Harris, C. and Stephens, M., “A Combined Corner and Edge Detector,” *Proceedings of the 4th Alvey Vision Conference*, Vol. 15, pp. 147–151.
- [19] Koenderink, J. J. and van Doorn, A. J., “Surface shape and curvature scales,” *Image Vision Computing*, Vol. 10, No. 8, 1992, pp. 557–564.
- [20] Witkin, A. P., “Scale-Space Filtering,” *Readings in Computer Vision: Issues, Problems, Principles, and Paradigms*, 1987, pp. 329–332.
- [21] Zhang, Z., Deriche, R., Faugeras, O. D., and Luong, Q. T., “A Robust Technique for Matching two Uncalibrated Images Through the Recovery of the Unknown Epipolar Geometry,” *Artificial Intelligence*, Vol. 78, No. 1-2, 1995, pp. 87–119.
- [22] Schmid, C. and Mohr, R., “Local Grayvalue Invariants for Image Retrieval,” *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 19, No. 5, 1997, pp. 530–535.
- [23] Mikolajczyk, K. and Schmid, C., “An Affine Invariant Interest Point Detector,” *ECCV (1)*, 2002, pp. 128–142.
- [24] Mikolajczyk, K. and Schmid, C., “Scale & Affine Invariant Interest Point Detectors,” *International Journal of Computer Vision*, Vol. 60, No. 1, 2004, pp. 63–86.
- [25] Mikolajczyk, K. and Schmid, C., “A performance evaluation of local descriptors,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, Vol. 27, No. 10, 2005, pp. 1615–1630.

- [26] Gonzalez, C., R., and Woods, R. E., *Digital Image Processing (3rd Edition)*, Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 2006.
- [27] ter Haar Romeny, B., *Front-End Vision and Multi-Scale Image Analysis: Multi-scale, Computer Vision Theory and Applications*, Vol. 27, Springer, 2003.
- [28] Moravec, H., “Obstacle Avoidance and Navigation in the Real World by a Seeing Robot Rover,” *tech. report CMU-RI-TR-80-03, Robotics Institute, Carnegie Mellon University & doctoral dissertation, Stanford University*, No. CMU-RI-TR-80-03, September 1980.
- [29] Lindeberg, T., “Feature Detection with Automatic Scale Selection,” *International Journal of Computer Vision*, Vol. 30, No. 2, 1998, pp. 79–116.
- [30] Burt, P. and Adelson, E., “The Laplacian Pyramid as a Compact Image Code,” *IEEE Transactions on Communications*, Vol. 31, No. 4, 1983, pp. 532–540.
- [31] Ke, Y. and Sukthankar, R., “PCA-SIFT: A More Distinctive Representation for Local Image Descriptors,” *Computer Vision and Pattern Recognition, IEEE Computer Society Conference on*, Vol. 2, 2004, pp. 506–513.
- [32] Bay, H., Ess, A., Tuytelaars, T., and Gool, L. V., “Speeded-Up Robust Features (SURF),” *Computer Vision and Image Understanding*, Vol. 110, 2008.
- [33] Besl, P. J. and McKay, N. D., “A Method for Registration of 3-D Shapes,” *IEEE Trans. Pattern Anal. Mach. Intell.*, Vol. 14, No. 2, 1992, pp. 239–256.
- [34] Chen, Y. and Medioni, G. G., “Object modelling by registration of multiple range images,” *Image Vision Comput.*, Vol. 10, No. 3, 1992, pp. 145–155.

- [35] Chua, C.-S. and Jarvis, R., “Point Signatures: A New Representation for 3D Object Recognition,” *International Journal of Computer Vision*, Vol. 25, No. 1, 1997, pp. 63–85.
- [36] Frome, A., Huber, D., Kolluri, R., Bulow, T., and Malik, J., “Recognizing Objects in Range Data Using Regional Point Descriptors,” *Computer Vision - ECCV*, 2004, pp. 224–237.
- [37] Gernsheim, H. and Gernsheim, A., *The History of Photography*, Thames & Hudson, London, 1969.
- [38] DeltaSphere, “The DeltaSphere-3000 3D Scene Digitizer,” <http://www.deltasphere.com/DeltaSphere3000.htm>, 2011.
- [39] Brady, J., Ponce, J., Yuille, A., and Asada, H., “Describing Surfaces,” *Proceedings of the 2nd International Symposium on Robotics Research*, MIT Press, 1985, pp. 5–16.
- [40] Terzopoulos, D., *Multiresolution Computation of Visible-Surface Representations*, Ph.D. thesis, 1984.
- [41] Desbrun, M., Meyer, M., Schröder, P., and Barr, A., “Implicit Fairing of Irregular Meshes using Diffusion and Curvature Flow,” *Proceedings of the 26th annual conference on Computer Graphics and Interactive Techniques*, SIGGRAPH 99’, ACM Press/Addison-Wesley Publishing Co., New York, NY, USA, 1999, pp. 317–324.
- [42] Belkin, M. and Niyogi, P., “Towards a Theoretical Foundation for Laplacian-Based Manifold Methods,” Springer, Philadelphia, Pa, USA, 2005, pp. 486–500.

- [43] Belkin, M., Sun, J., and Wang, Y., “Constructing Laplace Operator from Point Clouds in R^d ,” *Proceedings of the 20th annual ACM-SIAM Symposium on Discrete Algorithms*, SODA 09’, Society for Industrial and Applied Mathematics, Philadelphia, Pa, USA, 2009, pp. 1031–1040.
- [44] Vedaldi, A. and Fulkerson, B., “VLFeat: An Open and Portable Library of Computer Vision Algorithms,” <http://www.vlfeat.org/>, 2008.
- [45] “Oxford Visual Geometry Group: Data,” <http://www.robots.ox.ac.uk/~vgg/data/>.

VITA

Brien Roy Flewelling was born in Norfolk, VA. He received his Bachelor of Science degree in aerospace engineering from Syracuse University, Syracuse, New York in May 2006. He began his graduate studies at Texas A&M University in August 2006 under Dr. John Junkins. In 2007, Brien was named as a Department of Defense Science, Mathematics, and Research for Transformation (SMART) Scholar and he graduated with his Ph.D. in May 2012. His research interests focus on multiresolution analysis, sensor fusion, image processing, and active and passive sensing for proximity operations. He has recently joined the Guidance, Navigation, and Control group at the Air Force Research Laboratories in Kirtland Air Force Base, New Mexico, where he will continue his current research in the newly formed Sensor-based Control of Relative-Attitude Motion (SCReAM) Laboratory.

Brien Flewelling's email address is: Brien.Flewelling@kirtland.af.mil. He can be reached at:

Air Force Research Laboratories - Space Vehicles Directorate
c/o Dr. Brien Flewelling
3550 Aberdeen Ave SE
Kirtland AFB, NM 87117