



Dissertation

---

AUTOMATIC RECONSTRUCTION OF URBAN  
ENVIRONMENTS FROM AERIAL IMAGES

---

**Zebedin Lukas**

Graz, Austria, July 2009

*Thesis supervisors*

Horst Bischof  
TU Graz

Blaise Aguera y Arcas  
Microsoft





## Statuary Declaration

*I declare that I have authored this thesis independently, that I have not used other than the declared sources / resources, and that I have explicitly marked all material which has been quoted either literally or by content from the used sources.*

---

Place

---

Date

---

Signature

## Eidesstattliche Erklärung

*Ich erkläre an Eides statt, dass ich die vorliegende Arbeit selbstständig verfasst, andere als die angegebenen Quellen/Hilfsmittel nicht benutzt, und die den benutzten Quellen wörtlich und inhaltlich entnommene Stellen als solche kenntlich gemacht habe.*

---

Ort

---

Datum

---

Unterschrift



Either mathematics is too big for the human mind or the human mind is more than a machine.

---

*Kurt Godel (1906-1978)*



# Abstract

Recovery of realistic models of urban environments from aerial imagery has been an important research topic in computer vision for more than 20 years. Recently, large scale efforts from large enterprises like Microsoft are underway, aiming at building a virtual equivalent of our planet including realistic models of thousands of cities worldwide. The enormous scale of such a project, encompassing tens of thousands of the largest cities worldwide, makes any manual attempt to solve the arising problems prohibitively expensive. The only way to tackle such an undertaking is by a fully automatic processing pipeline. Such a workflow would start with the raw images collected by a digital aerial camera mounted on an aircraft and would automatically produce a virtual, digital representation of the covered area.

In this thesis algorithms for such a fully automatic processing pipeline are presented. The main contribution of the thesis focuses on the reconstruction of urban environments and the extraction and simplification of individual building models.

The basis for the recovery of virtual models of urban environments is a digital surface model. The common approach of energy minimization is chosen for this task. A set of convex energies and their optimization is studied which are increasingly important in many computer vision tasks. Convex functions have the useful property that a globally optimal solution can be efficiently computed. It turns out that second order regularization is superior to first order methods for deriving a surface model of urban environments. Using synthetic and real-world datasets it is shown that a discontinuity preserving, spatial regularization and a robust data term are very important for recovering high quality surface models from multiple observations.

The digital terrain model is then used to derive large, consistent ortho images from the single input images. This is achieved by reprojecting each image onto the ortho plane via the geometry induced by the surface model. Mathematically this task can be framed as a labeling problem of where to put the borders between the patches. A survey is conducted comparing different algorithms for solving the underlying binary problem

used in the alpha expansion moves. The remaining seamlines are hidden with Poisson blending. On the other hand, the variational method used to derive the digital surface model can be extended to vectorial functions. The resulting functional can be efficiently computed on parallel graphics hardware and additionally exploits the redundancy inherent in the imagery. Ortho images from real-world datasets obtained via the labeling and the continuous color surface approach are presented and compared. The ortho images in conjunction with the digital surface model can be used for interactive exploration via image based rendering techniques, independently from the way they are generated.

Finally a sequence of operations is proposed which allows one to efficiently recover and simplify realistic polygonal models for urban buildings. First a set of geometric hypotheses is collected with robust sampling techniques. The set used in thesis comprises planes and surfaces of revolution, but could be arbitrarily extended with other types. Then a line based over-segmentation of the footprint of the building is computed. Each segment is assigned to one geometry hypothesis, incorporating a smoothness constraint in order to encourage homogeneous regions. The labeled footprints are then converted to virtual, three dimensional models, which can be textured and used in other applications.

**Keywords.** Aerial Images, 2D Range Image Fusion, Building Reconstruction

# Kurzfassung

Die Gewinnung von realistischen Modellen von urbanen Umgebungen aus Luftbilder ist seit mehr als 20 Jahren ein wichtiges Forschungsthema in der digitalen Bildverarbeitung. Kürzlich haben Unternehmen wie Microsoft gross angelegte Anstrengungen in diesem Bereich unternommen, mit dem Ziel ein virtuelles Ebenbild von unserem Planeten zu erschaffen, in dem auch tausende urbane Räume mit realistischen Modellen vertreten sind. Die enorme Grösse von einem solchen Projekt, das zig tausende der grössten Städte der Welt umfasst, macht jeden manuellen Ansatz um die entstehenden Probleme zu lösen unerschwinglich teuer. Die einzige Möglichkeit ein solches Unterfangen zu bewerkstelligen ist die vollautomatische Abarbeitung durch Computerprogramme. Diese Arbeitsschritte würden mit den Rohdaten beginnen, die von einer digitalen Luftbildkamera, die auf einem Flugzeug montiert ist, beginnen und automatisch eine realistische, virtuelle Repräsentation des beflogenen Gebiets erzeugen.

In dieser Doktorarbeit werden Algorithmen für eine vollautomatischen Verarbeitung vorgestellt. Der Hauptbeitrag der Arbeit fokussiert sich auf die Rekonstruktion von urbanen Umgebungen and die Extrahierung und Vereinfachung von individuellen Gebäudenmodellen.

Die Basis für die Gewinnung von virtuellen Modellen von urbanen Umgebungen ist ein digitales Oberflächenmodel. Der bekannte Ansatz der Energieminimierung wird für diese Aufgabe verwendet. Ein Satz von konvexen Energien, die auch in vielen anderen Teilbereichen der Bildverarbeitung an Wichtigkeit gewinnen, und deren Optimierung wird studiert. Konvexe Funktionen haben die nützliche Eigenschaft, dass eine global optimale Lösung effizient berechnet werden kann. Es stellt sich heraus, dass eine Regularisierung zweiter Ordnung den Methoden erster Ordnung überlegen ist um ein Oberflächenmodel von urbanen Räumen abzuleiten. Mit Hilfe von synthetischen und realen Datensätzen wird gezeigt, dass es wichtig ist, Unstetigkeiten zu erhalten, räumliche Regularisierung anzuwenden und eine robuste Kopplung an die gegebenen Datenwerte vorzunehmen, um

qualitativ hochwertige Oberflächenmodelle von mehrfachen Beobachtungen zu erlangen.

Das digitale Oberflächenmodell kann dann benutzt werden um grosse, konsistente Ortho Bilder von den einzelnen Eingabebildern abzuleiten. Das wird dadurch erreicht, dass jedes Eingabebild mittels der Geometrie, die durch das Oberflächenmodell vorgegeben wird, auf die Ortho-Ebene entzerrt wird. Mathematisch wird diese Aufgabe als Kennzeichnungsproblem formuliert, wo die Grenzen zwischen den einzelnen Bildflecken gelegt werden soll. Eine Untersuchung wird durchgeführt um die einzelnen Algorithmen zu untersuchen, die das zugrundeliegende binäre Entscheidungsproblem lösen. An den entstehenden Säumen wird dann mit Hilfe der Poisson-Gleichung übergeblendet. Auf der anderen Seite werden die Variationsmethoden untersucht, die auch schon verwendet wurden um das digitale Oberflächenmodell abzuleiten, um diese so zu erweitern, dass sie auch mit vektoriiellen Funktionen umgehen können. Das resultierende Funktional kann effizient auf paralleler Graphikhardware berechnet werden und nutzt noch dazu die vorhandene Redundanz in den Eingabebildern aus. Es werden Ortho Bilder aus realen Datensätzen gezeigt und verglichen, die mit beiden Ansätzen berechnet wurden. Diese Ortho Bilder können in Verknüpfung mit dem digitalen Oberflächenmodell für eine interaktive Untersuchung mittels bildbasierten Darstellungstechniken benutzt werden, unabhängig davon wie sie ursprünglich erstellt wurden.

Schließlich wird noch eine Sequenz von Operationen vorgeschlagen, die es einem erlauben polygonale Gebäudemodelle effizient zu gewinnen und zu vereinfachen. Zuerst wird ein Satz von geometrischen Hypothesen mittels einer robuste Methode zur Probeentnahme gewonnen. Der Satz, der in dieser Arbeit verwendet wurde, besteht aus Ebenen und Rotationskurven, kann aber beliebig um andere Typen erweitert werden. Dann wird eine linienbasierte Übersegmentierung der Basisfläche des Gebäudes berechnet. Jedem Segment wird unter Berücksichtigung einer Glattheitsauflage eine geometrische Hypothese zugeordnet. Der so etikettierten Gebäudegrundriß wird dann in ein dreidimensionales Modell übergeführt, das mit Bildern verkleidet und in anderen Applikationen verwendet werden kann.



# Acknowledgments

I would like to thank my current and former colleagues from the center for Virtual Reality and Visualization, the Institute of Computer Graphics and Vision at the Technical University of Technology and Microsoft for their support, collaboration, help, comments and critique. Among those, I would especially like to express my gratitude towards Konrad Karner, who introduced me into this research field and provided the necessary freedom to do my research, and Andreas Klaus, who contributed much with his ideas and insights. Thomas Pock deserves special mentioning for his assistance and patience while answering my many questions. I would also like to thank both my supervisors, Horst Bischof and Blaise Aguera y Arcas, for their support and guidance.

I am deeply thankful towards my parents: for the possibility to study, their encouragement and support.



# Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
1.1	Introduction . . . . .	1
1.2	Applications . . . . .	3
1.3	Contribution of this Thesis . . . . .	4
1.4	Geometric versus Semantic versus Image-Based Representation . . . . .	5
1.5	Notation . . . . .	7
1.6	A Highly Automatic Workflow for Aerial Images . . . . .	8
1.6.1	Initial Classification . . . . .	10
1.6.2	Automatic Aerial Triangulation . . . . .	13
1.6.2.1	Extraction of Points of Interest . . . . .	15
1.6.2.2	Calculation of Feature Descriptors . . . . .	15
1.6.2.3	Establishing matching candidates . . . . .	15
1.6.2.4	Outlier elimination . . . . .	15
1.6.2.5	Relative orientation . . . . .	15
1.6.2.6	Final orientation . . . . .	16
1.6.3	Dense Stereo Matching . . . . .	17
1.6.3.1	Local Methods . . . . .	17
1.6.3.2	Global Methods . . . . .	18
1.6.3.3	Semi-Global Methods . . . . .	18
1.6.4	Range Image Fusion and DTM Generation . . . . .	20
1.6.5	Ortho Image Generation . . . . .	21
1.6.6	Land Use Classification . . . . .	22
1.6.7	Building Reconstruction . . . . .	24
1.7	UltraCam as Image Source . . . . .	24
<b>2</b>	<b>Robust Range Image Fusion</b>	<b>29</b>
2.1	Introduction . . . . .	29
2.2	Convex Methods for Image Restoration . . . . .	30
2.3	Generalization to Multiple Observations . . . . .	33
2.3.1	Probabilistic Modelling of Multiple Data Terms . . . . .	33
2.3.2	Extending the Energy Models to Multiple Observations . . . . .	36

2.4	Efficient Optimization . . . . .	37
2.4.1	Legendre-Fenchel Conjugate . . . . .	37
2.4.2	Primal-Dual Gap . . . . .	39
2.4.3	Tikhonov Energy Model . . . . .	40
2.4.4	$ROF_\varepsilon$ Energy Model . . . . .	42
2.4.5	$TV_\varepsilon-L_\delta^1$ Energy Model . . . . .	43
2.4.6	$TGV^2-L_\delta^1$ Energy Model . . . . .	44
2.5	Experimental Results . . . . .	45
2.5.1	Synthetic Dataset . . . . .	45
2.5.2	Aerial Dataset . . . . .	55
2.5.3	Middlebury Dataset . . . . .	56
2.6	Conclusions . . . . .	63
<b>3</b>	<b>Ortho Image Generation</b>	<b>65</b>
3.1	Introduction . . . . .	65
3.2	Warping of Images . . . . .	66
3.3	Compositing Images . . . . .	68
3.3.1	Composition via Labeling . . . . .	68
3.3.1.1	Graph Cuts . . . . .	70
3.3.1.2	CUDA Cuts . . . . .	71
3.3.1.3	Continuous Cuts . . . . .	72
3.3.2	Helmholtz Blending . . . . .	75
3.3.2.1	Multigrid Implementation . . . . .	77
3.3.3	Composition via a Continuous Color Surface . . . . .	77
3.4	Experimental Results . . . . .	79
3.4.1	Comparison of Labeling Algorithms . . . . .	80
3.4.2	Manhattan . . . . .	83
3.4.3	Graz . . . . .	85
3.5	Conclusions . . . . .	85
<b>4</b>	<b>Building Reconstruction</b>	<b>91</b>
4.1	Introduction . . . . .	91
4.2	Overview of the Method . . . . .	93
4.3	Geometric Primitives . . . . .	94
4.3.1	Planes . . . . .	95
4.3.2	Surfaces of Revolution . . . . .	95
4.4	Segmentation . . . . .	97
4.5	Information Fusion . . . . .	98
4.5.1	Graph Cuts Optimization . . . . .	98
4.5.2	Levels of Detail . . . . .	100
4.5.3	Pixel-Based Graphcut . . . . .	100

---

4.6	Experimental Results . . . . .	101
4.6.1	Graz . . . . .	101
4.6.2	Manhattan . . . . .	101
4.6.3	Quantitative Evaluation . . . . .	103
4.7	Conclusions . . . . .	105
<b>5</b>	<b>Conclusions</b>	<b>109</b>
5.1	Summary and Conclusions . . . . .	109
5.2	Future Work . . . . .	110
<b>A</b>	<b>Acronyms and Symbols</b>	<b>113</b>
	<b>Bibliography</b>	<b>115</b>



# List of Figures

1.1	A flowchart of a highly automatic workflow for processing aerial images. The left row deals with the geometric reconstruction of the scene, whereas on the right hand side semantic information is derived from the images. At the top the input images are fed into the pipeline and towards the bottom more refined products like digital surface model, ortho images, three dimensional building models and land use classification are available. . . . .	9
1.2	On the top a part of a color image is depicted with the corresponding probabilities of their land use assigned by a SVM on the image at the bottom. The intensities of the color channels correlate with the probability for a class: red maps to the class "solid", green to the class "vegetation" and blue to the class "water". . . . .	12
1.3	Two images of the same building with a large viewpoint change. The façades cannot be used for matching as none is visible in both images - rejecting those candidates can prove quite challenging as the texture is very similar. This leaves only the roof and ground for obtaining correct matching. . . . .	14
1.4	An oriented block of five stripes of 31 images each, denoted by small arrows, is shown here. The reconstructed tie points (about 70 000) are displayed as black dots. . . . .	16
1.5	On the left side the key image is shown, which is combined with another input images to produce a pixel-synchronous range image (right hand side). Those range images contain outliers and regions where no reliable depth value could be estimated (black pixels in the range image). . . . .	19
1.6	A digital surface model derived by fusing all available range images into one height field using the method proposed in this thesis. . . . .	20
1.7	This Figure shows ortho images composited from multiple input images using the DSM (a) as a reference surfaces. Even though the DSM is not a true 3D data structure, its geometry can be used to generate a high quality true ortho image as the error caused by roof overhangs and similar truly three dimensional structures is minimal. Ortho image (b) is produced by a labeling and stitching approach, whereas (b) was obtained by estimating the continous color surface. . . . .	23

1.8	The classification results of the RGB-NIR images are combined with the height information to derive a land use classification image. In the top left (a) the height field is shown with the corresponding probabilities (coded in the same fashion as in Figure 1.2) to the right (b). By calculating the difference between DSM and DTM, one obtains the estimated local height (c) - the intensity maps to the absolute value of the height and the color indicates the sign; negative heights only appear in the river because the surface of the water cannot be matched. In (d) the final result is shown. . . . .	26
1.9	These figures depict the data which is used for the reconstruction process: (a) height field, (b) building mask and (c) 3D line segments. Image (d) shows the obtained model by our algorithm. . . . .	27
2.1	3D renderings of a church reconstructed by the proposed $TGV^2-L_\delta^1$ method. The left image shows the untextured surface model, the right image depicts a textured version. . . . .	30
2.2	Probability distributions like those depicted in (a) have a corresponding analytical penalty norm (linear, quadratic and Huber), as shown in (b). . . . .	35
2.3	Signal-to-Noise ratios depending on number of observations. These plots graph the function $10 \log \frac{ g ^2}{ u-g ^2}$ , where $u$ is the minimizer of the energy functional and $g$ is the ground truth signal. . . . .	47
2.4	Input Data: On the left side the key image is shown, which is combined with two other input images to produce a pixel-synchronous range image (right hand side). Those range images contain outliers and regions where no reliable depth value could be estimated (black pixels in the range image). . . . .	50
2.5	A robust dataterm is indispensable for real-world datasets: The Tikhonov and $ROF_\epsilon$ energy models yield very bad results, because they cannot handle outliers. The $TV_\epsilon-L_\delta^1$ and $TGV^2-L_\delta^1$ show much better and similar results. Their difference becomes visible in a 3D rendering and plotting a cross-section of the building. . . . .	51
2.6	A warehouse with a very regular roof structure can be used to examine the behaviour of the energy models on sloped surfaces. Again, the results without a robust dataterm are unsatisfactory. The difference between first and second-order regularization become visible by plotting a cross-section of the roof. . . . .	52
2.7	3D renderings of the surfaces reconstructed with the different energy models make it easier to compare the quality as there is no ground truth. From top to bottom: Tikhonov, $ROF_\epsilon$ , $TV_\epsilon-L_\delta^1$ and $TGV^2-L_\delta^1$ . . . . .	53



- 2.8 A cross-section through the roof of the warehouse comes close to an idealized saw-tooth function. One notices the staircasing artifacts of the  $TV-L^1$  energy model which approximate the slopes. In comparison to that, the second order regularization of the  $TGV^2-L_\delta^1$  model yields a much better approximation to this sawtooth-like structure. The cross-section of the church highlights how the  $TGV^2-L_\delta^1$  energy model recovers the apex and the small turrets. . . . . 54
- 2.9 Manhattan, New York: The upper image shows lower Manhattan which was covered by approximately 250 images, which were acquired with a very high forward and sideward overlap (90% / 80% at 15cm GSD), because the downtown area features very high skyscrapers which make it difficult to get enough observations on the street level there with lower overlaps. The range image fusion process took about 15 minutes on a workstation with four Tesla cards. The two images at the bottom show details from the fused heightfield. . . . . 57
- 2.10 Oakland, California: This dataset consists of just 122 images and the range image fusion process takes 8 minutes. The ground sampling distance is 15cm and the overlaps are about 80% and 60%. Truly three dimensional objects pose a problem because they feature multiple valid height values for one point in the DSM. The left detailed view at the bottom shows that the crane is not fully reconstructed, because it has to contest with the roof surface. Often the size and thus the number of points is crucial whether an object gets included in the surface model. The right view shows that the contour of buildings is very sharp independent from the shape. . . . . 58
- 2.11 Pittsburgh, Pennsylvania: The upper image shows the result obtained for the city of Pittsburgh. The dataset consists of 358 images, the processing time is about 15 minutes. One can see that the hills were nicely recovered. Water is difficult to match, because it constantly moves and has changing reflections, therefore the height samples are not consistent there. This can be seen in the left image at the bottom where the surface is distorted around the bridge. The detailed views show the obtained accuracy for a stadium and a bridge, where even the regular pattern of the cross beams above the street is visible. . . . . 59
- 3.1 This Figures gives a schematic overview of the two possible ways how to derive an ortho image from aerial imagery. Both ways rely on a surface model to warp the images. The traditional approach is then to compute a labeling which minimizes the color differences between warped patches along some seams where a blending is applied. Alternatively a continuous color surface can be estimated which approximates all available samples at the same time. . . . . 66

- 3.2 This Figures shows the result of warping aerial images onto different geometries: in the top row two different aerial images are shown, which are warped on a DSM (middle row) and a DTM (bottom row). The left column depicts the geometry which is used. The accuracy of the DSM results in two pixel-synchronous warped images - some areas are of course not visible (black). The DTM only gives correct correspondences on ground level; all structures above the ground do not match in the warped views. As a benefit, there are no occluded areas as the geometry has no height discontinuities. . 69
- 3.3 On the left hand side the original, unmodified image is shown. 40% of the pixels are then flipped, yielding a distorted image which is used to evaluate the labeling algorithms. . . . . 80
- 3.4 Each row shows the results for a different algorithm: for the top row the 4-connected Graph Cuts algorithm was used, in the middle row the 8-connected Graph Cuts algorithm and in the bottom the Continuous Cuts algorithm. From left to right the smoothness weight is increased. It is noticeable that the Graph Cuts algorithm suffers from metrication errors - increasing the neighbourhood system mitigates this effect, but does not completely get rid of it. . . . . 81
- 3.5 This graph illustrates the advantage of the CPU for solving a binary MRF. The smoothness weight has a very limited impact on the runtime for the CPU as the algorithm focuses on remaining excessive flows which are rapidly pushed to available sinks. The implementation on the GPU on the other hand is good at processing pixels in parallel, but for higher smoothness weights only a fraction of the pixels need to be updated and push the flow further. The continuous cuts algorithm has the problem that it is difficult to estimate when a stable state is reached and no more iterations are necessary. In each iteration the update rules are applied to every pixel even if the resulting change is minimal. For high smoothness weights the diffusion process realized in the total variation takes long to converge. Note that using a different hardware or squeezing some cycles out of one specific implementation does not change the linear behaviour of the algorithms on the GPU. . . . . 82

- 3.6 This Figure shows the intermediate results of the push-relabel algorithm computed on the GPU. The left column shows where remaining capacities to the sink exist, the middle column indicates where excess flow exists which needs to be pushed to the sink. The right column depicts the estimated potential for each pixel. The top row shows the state after the first iteration, in the middle 50 iterations have passed and at the bottom 100. One notices that towards the end some pixels are already labeled and thus do not participate anymore in the optimization (thus reducing the potential of parallelization). It also occurs that the remaining paths from excess flows to sinks form a bottleneck which is not suited for parallel processing. . . . . 84
- 3.7 A typical result for the ortho image generation with the proposed algorithms is shown. In the top left corner the DSM is shown which is used to warp the input images. In the top right corner the color coded result of the labeling obtained from the Graph Cut algorithm is depicted. Each color represents a different input image. One notices how regions close a building are occluded in one image and need to be filled up by another one. In the bottom row, finally, the ortho images are given which are produced by the continuous color surface algorithm (left) and the labeling (right). Not all moving objects can be removed on the left hand side, because the crossing is too crowded. . . . . 86
- 3.8 In this figure the same layout is being used as in Figure 3.7. In the top row the DSM is shown on the left hand side, and the labeling result on the right hand side. In the bottom row ortho images are shown generated by both algorithms. In this scene the continuous color surface effortlessly removes all moving objects, as there is less traffic. . . . . 87
- 3.9 In this figure the same layout is being used as in Figure 3.7. In the top row the DSM is shown on the left hand side, and the labeling result on the right hand side. This scene shows the main square in Graz in front of the town hall. The continuous color surface removes all moving objects and presents a very clean ortho image. The purely stitched ortho still contains many pedestrians. . . . . 88
- 3.10 This scene shows a different drawback of the labeling approach: overhead street lamps are included in the DSM and severely occlude the street. This forces the labeling to change more often as there the algorithm has to compensate those occlusions. The result looks less natural because of the distortions, one car even has a hole in the middle. The regularization term employed by the continuous color surface successfully removes the cars and also hides the occlusions caused by the street lamps. . . . . 89

4.1	These figures depict the data which is used for the reconstruction process: (a) height field, (b) building mask and (c) 3D line segments. Image (d) shows the obtained model by the proposed method. . . . .	93
4.2	Illustration of the single steps of the proposed method: height data and building mask are used to obtain a set of geometric primitives; In parallel the 3D lines are used to generate a segmentation of the building. Finally, a labeled segmentation is produced. . . . .	94
4.3	Illustrations how starting with the dense height data the 3D curve is derived which generates the dome if it rotates around a vertical axis. (a) Raw height field with the detected axis, (b) all inliers are projected into the half-plane formed by axis and a radial vector, (c) the moving average algorithm produces a smooth curve. . . . .	96
4.4	Segmentation into polygons: (a) The matched 3D lines are projected into the $2\frac{1}{2}$ D height field, (b) outliers are eliminated by a weighted orientation histogram which helps to detect principal directions of the building. (c) Along those directions lines are grouped, merged and extended to span the whole building. . . . .	98
4.5	This Figure compares the results of the pixel-based (a) reconstruction with the segmentation-based (b) one. The façades of the pixel-based approach are not straight thus degrading the visual appearance, which can be observed very well by comparing (c) and (d). . . . .	102
4.6	The stages of the reconstruction are illustrated by means of the building of the Graz University of Technology: (a) Segmented height field, (b) labeled polygons after the Graph Cuts optimization, (c) screenshot of the reconstructed model ( $\lambda = 5$ ) . . . . .	103
4.7	Levels of Detail: The same building was reconstructed with different values for $\lambda$ . The number of geometric primitives used to approximate the shape of the roof is decreasing with higher values for $\lambda$ . In the upper row a screenshot of the reconstruction is depicted, below are illustrations of the matching labeling obtained by the Graph Cuts optimization. . . . .	104
4.8	Quality assessment with a manually generated ground truth: In (a) and (b) the height fields for the manually and automatically reconstructed building are shown, in (c) the height differences are shown. The largest difference in the placement of edges is about two pixels, which is about 30cm. . . . .	105
4.9	The cumulative probability distribution of the height difference for manual and automatic reconstruction. The graph shows the error distribution for 1973 buildings from a data set of Manhattan, New York. The left image shows the graphs for height differences up to 100 meters; the right graph zooms on differences up to five meters. . . . .	106

- 
- 4.10 Two detailed views of typical results for different types of buildings from the Manhattan data set: (a) rectangular buildings, (b) rectangular building with nicely integrated dome. . . . . 107
- 4.11 Two detailed views of typical results for different types of buildings from the Manhattan data set: (a) skyscrapers in downtown and (b) skyscraper with a spire. . . . . 108



# List of Tables

1.1	Current and legacy aerial cameras produced by Vexcel Imaging. Images from all models were used in this work. . . . .	24
2.1	Illustration of the synthetic dataset used to evaluate and compare the energy models. Image (a) depicts the ground truth, the other images (b) - (f) are distorted samples by adding Gaussian noise and 10%, 20%, 30%, 40% and 50% outliers, respectively. . . . .	45
2.2	Experiment with 10% outliers and 5 observations. For each algorithm the best result with respect to the $L^2$ -distance to the ground truth is depicted. One notices that a robust data term is crucial for a good result. The $TGV^2-L_\delta^1$ delivers better results than the $TV_\epsilon-L_\delta^1$ algorithm. The quality of the results is visible in the 3D renderings. . . . .	46
2.3	The Middlebury stereo vision suite consists of four reference datasets which are used by researchers around the globe to evaluate and compare their dense image matching algorithms. This table sorts the algorithms according to their percentage of bad pixels. On the official site a different sorting is used based on the average rank, which is unsuitable for this evaluation, as multiple additional results are evaluated. For each energy model the optimal parameter settings are used. In addition to the energy models studied in this thesis the median and average values of the top 10 disparities maps are also evaluated. The evaluated entries are highlighted in the table. . . . .	61
2.4	This table shows the results obtained by applying the $TGV^2-L_\delta^1$ algorithm to the results of the top 10 algorithms in the Middlebury ranking with an error threshold of 0.5 pixel. The first row shows one image of the stereo pairs of the four data sets. The second row shows the estimate of the disparity map by the $TGV^2-L_\delta^1$ energy model. In the third row an error image is depicted which shows where a wrong disparity value was estimated. See Table 2.3 for numerical details of these results. . . . .	62

- 4.1 The impact of the smoothness parameter  $\lambda$  on the reconstructed model. The number of unique labels used after the Graph Cuts optimization iterations decreases as well as the number of triangles in the polygonal model.  $\Delta$  Volume denotes the estimated difference in volume between the surface obtained by dense image matching and the reconstructed model (data term). The last column refers to the accumulated length of all borders in the final labeling (smoothness term). . . . . 103



# Chapter 1

## Introduction

### Contents

---

<b>1.1 Introduction</b>	<b>1</b>
<b>1.2 Applications</b>	<b>3</b>
<b>1.3 Contribution of this Thesis</b>	<b>4</b>
<b>1.4 Geometric versus Semantic versus Image-Based Representation</b>	<b>5</b>
<b>1.5 Notation</b>	<b>7</b>
<b>1.6 A Highly Automatic Workflow for Aerial Images</b>	<b>8</b>
<b>1.7 UltraCam as Image Source</b>	<b>24</b>

---

### 1.1 Introduction

Geographical Information Systems (GIS) already have a long tradition as digitized databases. But even before the advent of computer systems such collections of geographical data were available in the form of cadastral maps. Historically, surveyors were very careful in the selection of map features, as every instance has to be collected manually - independent of whether a tachymeter is used or the information is derived from aerial imagery (as routinely done today as aerial cameras are readily available and provide the necessary resolution and geometric precision). Either way, much manual labor is involved and the prices for such data are correspondingly high; Redundancy and over-sampling are understandably avoided, resulting, for example, in barely more than absolutely necessary along- or across-track overlap for flight missions during acquisition of aerial imagery targeted at manual evaluation.

Since about the middle of the twentieth century LiDAR (Light Detection and Ranging) systems became popular in some branches of land surveying, typically to generate digital surface models (DSM). Such a DSM is an elevation map which reflects the shape (height) of the surface including buildings, trees and other objects above the ground. It is often stored as a regular grid of height values. Mounted on an airplane a LiDAR device is able to scan the terrain below like a push broom sensor and generate a dense point cloud sampling the ground surface. With proper post-processing algorithms for determining the relative orientation and outlier filtering, one can then derive a dense DSM, which is of high interest for urban planning (building and sewer construction and other applications). Using this technique a ground sampling distance (GSD) - the horizontal distance between two points on the ground - below one meter is affordable. This level of detail would be prohibitively costly with manual evaluation; land surveying offices of cities routinely possess a (manually derived and maintained) digital terrain model (DTM) with a GSD over 20 meters. A digital terrain model differs from a pure DSM in that all objects above the ground, like trees, buildings and bridges are removed and a model of the bald earth is obtained. The severe drawback of LiDAR measurements is that only elevation data is gathered, as no images are collected. Even though the achievable resolution from a LiDAR scanner is comparable to that of image based sensing, there is little to no redundancy in the data which could be exploited by automatic workflows. Still, in the photogrammetric literature several arguments are mentioned in favour of surveying with LiDAR scanning. Recently however, a study [38] revisited this topic in the light of modern photogrammetric methods used today in fully automatic workflows. Three most important aspects are addressed: LiDAR produces instant 3D measurements, without the need for algorithmic treatment of the signal, it can yield more than one hit (first and last hit or full waveform reflections for example) and is reported to be more cost efficient than photogrammetric flight missions. The study concludes that photogrammetric methods are more cost efficient and often yields data of the same or better quality.

In the last two decades digital cameras started to dominate the market for aerial photogrammetry. In this time frame those cameras reached a mature technical state and hence provided the necessary geometric and radiometric stability and resolution to compete with the legacy analogue cameras. In addition, however, they feature the advantage that the acquired digital images are readily suitable for automatic processing by computers compared to analogue images which require a time consuming analogue-to-digital conversion. Advances in various areas of computer vision nowadays allow an alternative to

the traditional manual processing by exploiting the inherent redundancy in digital aerial images. This way many essential products regularly obtained from aerial flight missions, like digital surface models, ortho images and building models, are cost-efficiently derived by automated processing pipelines and can thus be offered at competitive prices.

## 1.2 Applications

Aerial surveying is a valuable tool across many disciplines. The diminishing costs of the acquisition and processing of aerial data allowed by the introduction of digital cameras and a high degree of automation in the processing workflows, aids to further spread the usage of the various products.

The applications of traditional ortho images and maps generated from aerial images span a huge and diverse set, including but not limited to archaeology, mining, wild life censuring, reconnaissance and land surveying. Commercially ortho images are also often used for navigation purposes or to provide better visualization of routes and locations.

Also more sophisticated products, however, have a high demand. Digital surface models are routinely used for simulations of water flows, mass movements (like avalanches, landslides and earthquakes) and other physical models where an accurate knowledge of the terrain surface is important. By calculating and minimizing the noise pollution the quality of life in the urban environment can be increased, for example. On the other hand such simulations help to prepare for emergency situations like flooding, riots and hurricanes in their forecast, management and recovery. Another common usage would be the creation of relief maps. Most importantly however, a precise digital surface model allows to rectify the registered images and stitch them together to obtain a true ortho image. The geometry induced by the DSM also allows to create realistic renderings of the city for visualization purposes and supports advanced algorithms in an automatic workflow in order to extract a DTM or building models for example.

A classical appliance for three dimensional city models is urban planning. Placing of antennas for mobile phone coverage for example requires knowledge of the environment in order to calculate the range and coverage. It is also possible to determine beforehand whether the antenna would have a negative aesthetic impact upon the neighbourhood. Optimizing the arrangement and orientation of solar panels is another topic in urban planning that is helped by realistic city models. Explicit description of building models (polygonal meshes for example) are also easier to transmit and visualize compared to pure elevation data like a DSM.

Apart from applications which require a professional GIS, there exists a broad set of more casual applications where geographical information can be incorporated, however, the accuracy requirements are often less stringent. This includes situations with an aesthetic background like visualizing or marketing an urban environment. Navigation systems are also becoming more ubiquitous and would profit from a three dimensional city model, which could displace the more traditional birds eye display style of those devices. In a next step those navigation systems could be augmented with semantic information and used for virtual tourism or on the spot as tourist guides. On a larger scale this leads to initiatives like Virtual Earth or Google Earth, which aim at reconstructing all major cities worldwide. Meta-information about sites, buildings and locations is linked in from well known sources like Wikipedia, but can also be supplied by users.

A more complete coverage of appliances, usages and target groups is given in [1].

### 1.3 Contribution of this Thesis

In this thesis new algorithms for a fully automatic processing pipeline are presented which surpass existing ones with respect to the quality of the attained results while being fully automated.

The basis for the recovery of virtual models of urban environments is a digital surface model. Such a digital surface model is a compact  $2\frac{1}{2}$ D representation of the shape and structure of the terrain, including objects like buildings and trees. It is often stored as a collection of height values sampled from a regular grid and can therefore be easily stored, transferred and displayed as an image. The common approach of energy minimization is chosen for this task. In particular a set of convex optimization methods is studied which are increasingly important in many computer vision tasks. Using synthetic and real-world datasets it is shown that a discontinuity preserving spatial regularization and a robust data term are very important for recovering high quality surface models from multiple observations.

These digital surface models are then used to derive large, seamless true ortho images from the individual input images. This is achieved by reprojecting each image onto the ortho plane via the geometry induced by the surface model. Two methods are described on how to compose those warped patches into one large, contiguous ortho image. First, the well known approach of labeling and stitching different images together is analyzed. Three different approaches for solving the labeling problem are discussed, with the goal to find the fastest method in order to minimize the processing time for projects covering

large areas. The other method of composing ortho images is to directly exploit the high quality DSM and redundancy inherent in the imagery and estimate a continuous color surface. This method has the additional benefit that it suppresses moving objects.

Finally a sequence of operations is proposed which allows one to efficiently recover and simplify realistic polygonal models for urban buildings. First a set of geometric hypotheses is collected with robust sampling techniques. The set used in this thesis comprises planes and surfaces of revolution, but could be arbitrarily extended with other types. Then a line based over-segmentation of the footprint of the building is computed. Each segment is assigned to one geometry hypothesis, incorporating a smoothness constraint in order to encourage homogeneous regions. The labeled footprints are then converted to virtual, three dimensional models, which can be textured and used in other applications.

## 1.4 Geometric versus Semantic versus Image-Based Representation

There exist many possibilities how to encode and store information about a virtual model of urban environments. Every format has different purposes and strengths as they are often tailored for specific applications. A very popular encoding for geometric scenes is the Virtual Reality Modelling Language (VRML), which has a structure closely resembling the interface offered by OpenGL, thus making it easy to display them. This means that every object is specified by vertices, triangles, colours and texture images. This explicit geometric representation has the drawbacks that it requires much space, does not offer an immediately obvious way to derive level of details and most importantly is difficult to enrich with semantic information. However, they are very flexible as such a representation is able to reproduce any object with an arbitrary degree of faithfulness to details at the cost of memory and processing time.

Another very verbose and explicit geometry schemes is often employed to represent the DSM and DTM. These elevation data on a regular grid are stored as a greyscale image with some additional metadata how to convert them back to 3D points. This representation is also very flexible, however, it requires large amounts of space and does not have any semantic information attached to it. Additionally it is restricted to  $2\frac{1}{2}$ D representations of the scenery. This drawback, however, is often not that grave as the viewing conditions in aerial images seldom allows to accurately estimate truly three dimensional features of buildings like roof overhangs.

On the other end of the spectrum of possibilities to encode a scene, lies the purely semantic description employed by humans. The number of floors, type of the roof and windows, color, shading and kind of texture, together with a rough sketch of the footprint are often enough to develop a good idea how a building will look like. Of course many details like cracks in the façade, stains or other individual variations of windows or other parts require a special encoding. This already hints at the drawback of such a way of specifying an object: the language of a semantic description has to support everything that is likely to appear in the scene, otherwise it cannot be expressed. The advantage of such an encoding is the high compression one is able to achieve. The encoding is also not tied to one specific graphical representation. Low power devices like smartphones, handhelds or navigation devices can derive only a rough three dimensional object from such a description, whereas a powerful desktop system could tap into a database of highly detailed templates in order to display a very detailed instance of the scene. An attempt of implementing such an encoding is made by CityGML ([34]), which builds a hierarchical system of objects which recursively describe the geometry of a building. The great benefit of this strategy is that it allows to reuse and parametrize components which were defined earlier. However, even though impressive results and compression ratios are achieved, all models currently have to be constructed by hand.

Turning ones attention to reconstruction algorithms, it is in general easier to target a very explicit geometric representation like voxel spaces or triangle meshes. On the other hand it is more difficult to exploit higher-level prior knowledge about an object that way. Therefore it would be desirable to combine the above mentioned advantages, for example by starting with an explicit geometric representation and moving to a semantically augmented format. However, how such a system would operate is still an open question.

Given the difficulties of geometric modelling, a third possibility of presenting scene information relies on image based modelling and rendering (IBMR). This way, no explicit and consistent three dimensional model has to be reconstructed, but only a proxy geometry for the input images is necessary for the view interpolation and transitions. The drawback of this approach is that the user is restricted to stay close to captured views, otherwise the quality of the synthesized view degrades. In between existing views transitions are generated by blending from one image to another using the available depth information. This approach usually conveys a very realistic feeling of a three dimensional scene, but is only applicable if no explicit virtual model is required (like in many of the scenarios mentioned in Section 1.2). For interactive exploration of a scene it is able to produce a

most realistic representation as it heavily relies on the original input images. Image based rendering has many advantages over traditional textured, truly three dimensional models. For view points coinciding or close to where an input image is available, the rendering gives very realistic and detailed results, as the input image can be used without noticeable distortions. Depending on the quality of the proxy geometry the user experience degrades also smoothly, in the worst case only a single dominant plane can be used [62]. Using only the tie points generated during solving the structure-from-motion problem and sparse 3D lines, it is already possible to find a better proxy geometry which greatly improves the visual appearance of image transitions [61].

The high quality geometry induced by the DSM generated with the algorithm presented in Chapter 2 can be used to allow such an immersive user experience by providing realistic and smooth transitions between images. In Chapter 4 it is then shown how to extract polygonal models of urban buildings, if an explicit geometric model is required.

## 1.5 Notation

In this thesis problems arising in the context of processing aerial imagery are addressed with the well established approach of energy minimization. Those problems are often ill-posed as they do not satisfy one or more criteria of a well-posed problem (existence, uniqueness and stability of the solution or solutions). Regularization stabilizes the solution, helps to introduce assumptions about the generating model and prevents overfitting. It is a well established practice in modern computer vision to phrase problems in terms of energy minimization. Additionally there exists a set of established methods and practices in order to handle such formulations.

In general, the energies investigated in this thesis are of the form

$$E(u) = \alpha \int_{\Omega} \Phi(\nabla u) dx + \int_{\Omega} \Psi(u, f) dx , \quad (1.1)$$

where the function  $u$  is the solution to the problem and  $f$  represents the given observations. This notation is used throughout the thesis. Both function are defined on the domain  $\Omega \in \mathcal{R}^2$  and map to a common vector space. The first term in this equation is the spatial regularization and serves to encourage smooth solutions. The function  $\Phi$  can have different shapes and forms (quadratic or linear for example), depending on the requirements of the problem. The function  $\Psi$  on the other hand ties the solution to the observed data. A parameter  $\alpha$  can be specified by the user and makes the trade-off between data

fidelity and smoothness of the solution.

Equation (1.1) formulates the energy minimization task in a continuous setting. Often, however, it is beneficial to compute a solution in a discrete setting. In this case the integrals are replaced by sums,

$$E(u) = \alpha \sum_{(p,q) \in \mathcal{N}} \Phi(u(p) - u(q)) + \sum_{p \in \mathcal{P}} \Psi(u(p), f(p)) , \quad (1.2)$$

where the set  $\mathcal{P}$  replaces the domain  $\Omega$  and denotes the set of all pixels. The set  $\mathcal{N}$  induces a graph representing all neighbouring pixels, thus appropriately defining the  $\nabla$ -operator via finite differences.

If a different convention or additional notations are used in this thesis, those exceptions are explicitly noted in the text.

## 1.6 A Highly Automatic Workflow for Aerial Images

Traditionally, workflows for aerial images already derive certain products from the given imagery fully- or semi-automatically. It is already a well established practice [77] to generate DSM, DTM, a land use classification and ortho images. More sophisticated products like three dimensional building models are only now becoming more common as automatic workflows replace much manual labour which would make those products prohibitively expensive or even infeasible.

In this section an overview of a highly automatic, state-of-the-art workflow is given, which takes the raw aerial images and generates above stated products from them. The algorithms presented in this thesis integrate into this framework. A schematic view of such a workflow is depicted in Figure 1.1. Some steps of the workflow may be done in parallel, others depend upon completion of previous stages, as can be seen in the diagram. This workflow serves as a framework and provides the necessary context within which the contributions of this thesis have to be seen. The focus within this thesis lies on the second half of the pipeline dealing with the reconstruction of urban environments and the extraction and simplification of individual building models.

This overview only focuses on the part relevant to the development of the algorithms described in this work. The properties of the image acquisition system, like radiometric and geometric calibration, and the flight management system are not discussed here. The initial classification works on each individual color image and can be calculated right at the start, as it depends only on some manually collected training samples. It produces a



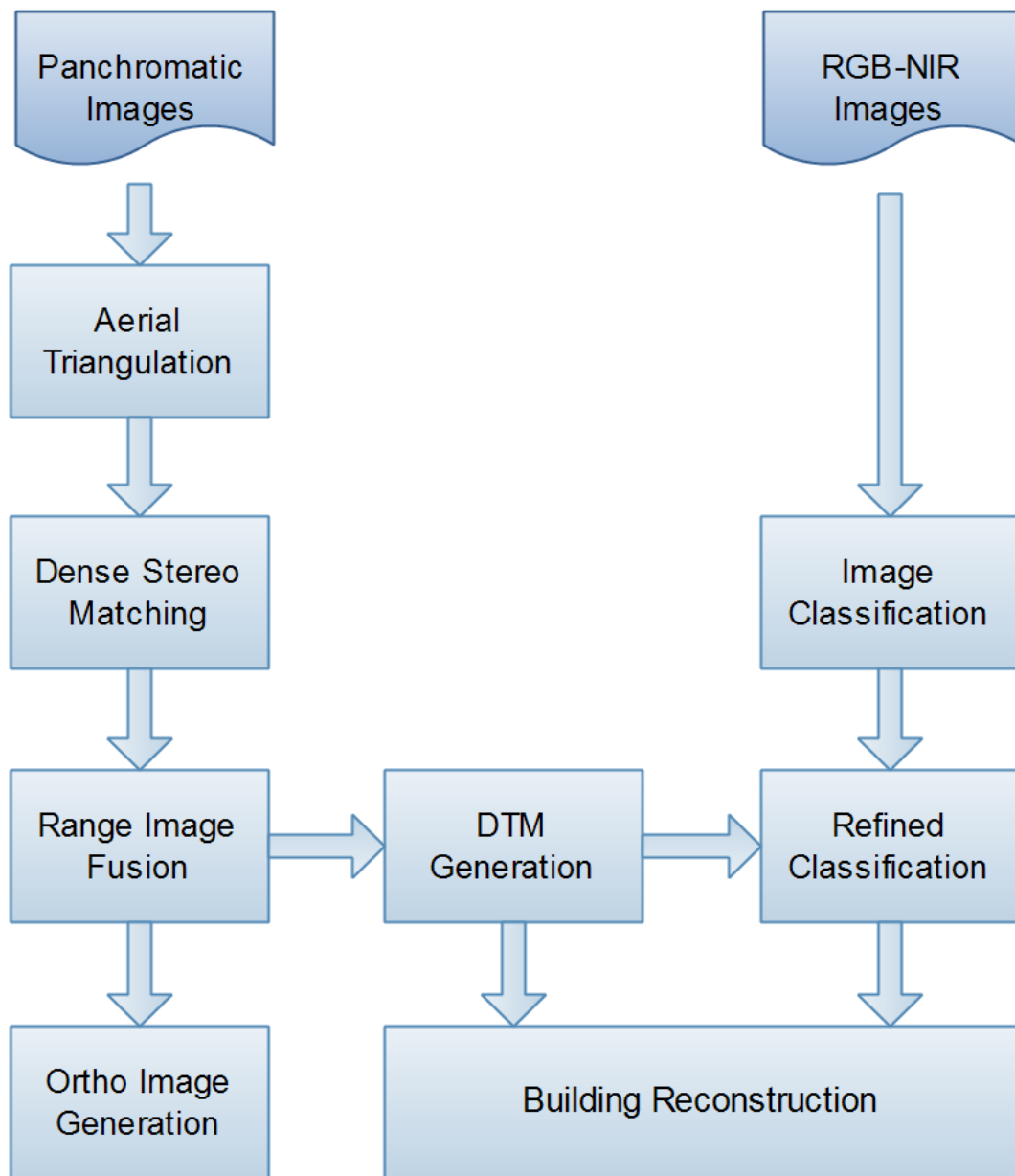


Figure 1.1: A flowchart of a highly automatic workflow for processing aerial images. The left row deals with the geometric reconstruction of the scene, whereas on the right hand side semantic information is derived from the images. At the top the input images are fed into the pipeline and towards the bottom more refined products like digital surface model, ortho images, three dimensional building models and land use classification are available.

per-pixel probability distribution of the land use for each image. Those probabilities are later merged to obtain a robust land cover classification. The foundation of the geometrical reconstruction, on the other hand, consists in the Aerial Triangulation, which determines the relative and also absolute position (using either GPS measurements or ground control points) and orientation of each image. Using that information, image tuples are used to generate dense depth information using an area based image matching algorithm. The individual range images are then fused together, to obtain a DSM, which is later refined into a DTM by removing buildings, trees and other objects. The geometry induced by the DSM is also necessary to compute true ortho images and the land use classification. Traditional ortho images are computed by warping the color images on the DTM and stitching them together. Finally the land use classification is transformed into a building mask, which in turn is necessary for the generation of three dimensional building models.

Similar workflows have recently been proposed ([18, 30]), which demonstrate that it is possible to automatically derive a digital terrain model (DTM), digital surface model, land use classification and ortho image from aerial images.

The only step requiring manual interaction is the training of the classifiers for the color and NIR images. If a geo-referencing of the data set is required, additional manual input is required in form of ground control points.

In the following the prototype workflow used as the framework in this thesis is described. More information on each part can be found in [77].

### 1.6.1 Initial Classification

The goal of the initial classification is to produce a probability distribution for each pixel in the input images describing its land use classification. This information is not only useful for later stages in the processing pipeline (for recognizing building blocks for a 3D model reconstruction for example), but is itself a product which is often sought by land surveying offices and other customers.

In this early stage of the pipeline no height information is available - also dense stereo matching algorithms often yield outliers or bad matches, therefore this height information is not reliable. On the other hand it is not necessary to already obtain a final classification result and assign a definite class to each pixel. The redundancy of the image data can be better exploited at a later stage of the pipeline if the full probability distribution for the available classes is stored instead of one class. Thus, it is desirable to compute probabilities for each class, as this classification will be followed up later, once height information is

available (giving access to high quality correspondences) in order to refine and spatially regularize the assignment of classes.

Given the fact that the set of land use classes is often known in advance or mandated by the user, a supervised multi-class classification problem is faced. This means that an operator has to select samples for each land use class from the input images which are typical for a specific class and which are then used as training data for the classifier. Following the classical approach of pattern recognition, discriminating features are selected and used to train a classifier. This classifier then generalizes the observed feature values and is used to obtain a probability distribution for all pixels in the input images.

The selected training samples are crucial for the classifier to learn the class boundaries and therefore have to be picked carefully. Those groups of pixels should be a good representation of the land cover and surface phenomenon for a specific class. Using this information, the classifier is trained to generalize this mapping of feature values to class label. The operator has to decide on the training sites, even though a semi-automatic approach is plausible, where similar, already labelled or classified samples are used to generate the set of training samples.

The identification of those training sites has to be done after the images have been processed with the radiometric camera calibration parameters and when the weather and lightening conditions or the land properties change significantly. Those procedures are important if two different camera systems are used for example or images for one project area are taken on multiple days. However, usually the operator has to identify training areas in a single image per flight only, because often the weather and lightening condition do not change dramatically during one flight. As a rule of thumb the training samples for each class should sufficiently capture the variation of the data.

In the prototype workflow discussed in this thesis, the initial classification uses Support Vector Machines (SVM) as they are a very common classifier for use with multi-spectral, aerial data. Huang et al. [28] demonstrated the applicability of SVM to derive land use from operational sensor systems and to evaluate systematically their performances in comparison to other popular classifiers. The number and nature of the target classes can be adapted to the target area (rural or urban for example), even though a basic set of classes is often sufficient and does not need to be modified. This basic set used for initial classification in urban and suburban areas consists of the classes "solid", "shadow", "water" and "vegetation" for the purposes of this workflow. Figure 1.2 depicts the classification result of a test area in an input image. One notices that the lack of spatial regularization or

grouping leaves outliers and fuzzy borders between the classes. This task is left to a later stage which is then also able to exploit the redundancy of the imagery once high quality correspondences are available.

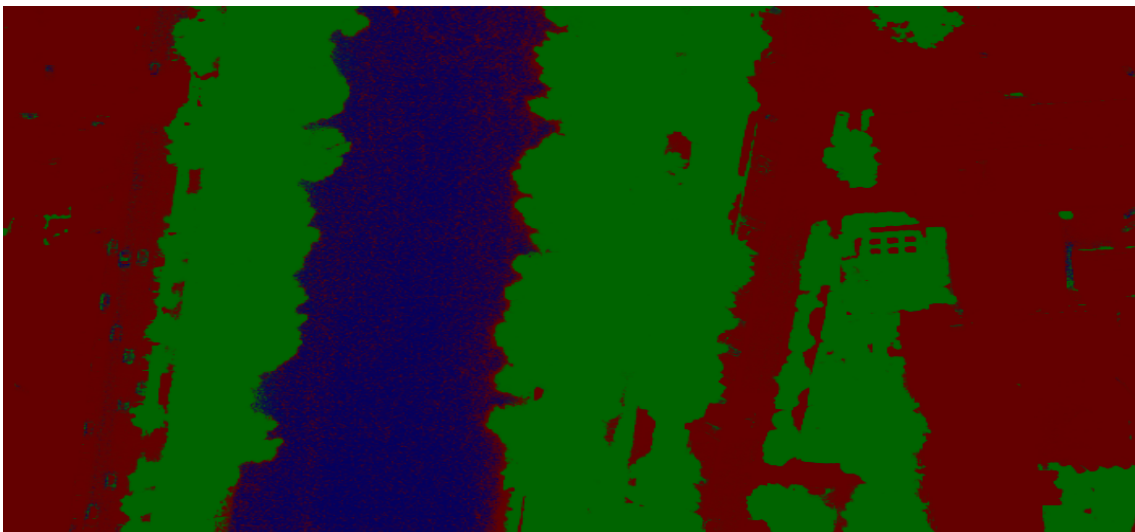


Figure 1.2: On the top a part of a color image is depicted with the corresponding probabilities of their land use assigned by a SVM on the image at the bottom. The intensities of the color channels correlate with the probability for a class: red maps to the class "solid", green to the class "vegetation" and blue to the class "water".

## 1.6.2 Automatic Aerial Triangulation

The goal of aerial triangulation (AT) is the establishment of correspondences, denoted as tie points, between adjacent images [59]. Digital airborne cameras are able to deliver highly redundant images which result in small baselines. Ideally in data sets specifically acquired for automatic processing the stripes of images have at least 80% forward overlap and at least 60% side overlap - very dense downtown areas with high skyscrapers like Manhattan require an even higher overlap (up to 90% forward and 80% sideward overlap) in order to guarantee that even streets between skyscrapers can be reconstructed for example, as they need to be visible in at least two images. The requirements of the AT with regard to the image along- and across-track overlap are less stringent, as only enough candidate matches have to be established to robustly compute the relative orientation between adjacent images. As few as five correct correspondences ([47]) are enough for that with a calibrated camera, even though in practice many more are computed in order to allow for a robust selection scheme like RANSAC.

The forward overlap, however, is restricted by the trigger rate of the camera and further depends on the aircraft altitude and the flying speed. In the case of a low altitude the baseline relative to the depicted scene and thus the viewpoint change becomes larger as illustrated in Figure 1.3. For image pairs with a low side overlap the viewpoint change may be even larger. For long flight lines the changes of shadows caused by the time difference in the image acquisition or a change in the weather conditions for example can already cause problems for the computation of correspondences. Therefore a wide baseline matching problem has to be faced in order to establish correct correspondences [42].

Techniques that do not take slanted surfaces into account, such as window based correlation methods, may fail in this case, due to projective and radiometric distortions. The following assumptions are made to allow a fully automatic processing: The projective distortion of small surface patches can be approximated by affine transformations. The motivation is that a small planar surface patch locally undergoes an approximate affine transformation if the viewpoint changes. In general smooth surfaces can be modelled by piecewise planar patches. This approximation is valid for urban areas as well as for rural regions.

As mentioned above a certain overlap between the images is required. The requirements with regard to the overlap for dense image matching are much higher than for aerial triangulation - for a successful aerial triangulation in urban environment the along-track overlap should at least exceed 60% and the across-track overlap 20%. Thus, every point



(a)

(b)

Figure 1.3: Two images of the same building with a large viewpoint change. The façades cannot be used for matching as none is visible in both images - rejecting those candidates can prove quite challenging as the texture is very similar. This leaves only the roof and ground for obtaining correct matching.

is visible in at least three images in one flight strip and the strips themselves are linked among each other via tie points at their shared border.

The in-plane rotation angle between adjacent images within a single image strip is another important factor and should be kept as low as possible. If the altitude is nearly constant, the scaling factor does not vary significantly and does not need to be accounted for in the descriptor of the points of interest, thus facilitating the matching procedure.

The aerial triangulation can be decomposed into six separate steps, described in the next paragraphs.

### 1.6.2.1 Extraction of Points of Interest

Several thousand Points of Interest (POIs) are extracted in each image by applying a Harris operator (Harris POIs) and by utilizing line intersection as proposed in [6]. The POIs are sorted according to their corner strength and a weighted non-maximum suppression is applied to guarantee a good distribution over the image. A sub-pixel refinement of the Harris POIs is then applied by a quadratic fit to the pixel intensity values.

### 1.6.2.2 Calculation of Feature Descriptors

The feature descriptor for POIs from line intersection are calculated within the sector enclosed by the two lines and are weighted reciprocal to their distance from the line intersection point. All descriptors are gradient orientation histograms which are similar to those proposed by [40]. The main difference is that the descriptor of Lowe uses more discretization steps, since it has to contain enough information to find a unique correspondence.

### 1.6.2.3 Establishing matching candidates

The feature vectors are matched to find a 1 to n mapping between POIs of all adjacent images. Each of the n best candidates is evaluated by applying an adaptive area based correlation. This process optimizes the normalized cross correlation by adapting an affine transformation for each candidate correspondence. In order to fulfill the non-ambiguous criteria, only matches with a highly distinctive score are retained.

### 1.6.2.4 Outlier elimination

The robustness of the matching process is enhanced by backmatching and by topological filtering [68]. Another restriction is enforced by the epipolar geometry. Therefore the RANSAC method is applied using the well known five point algorithm ([47]).

### 1.6.2.5 Relative orientation

As a result from the last step, inlier correspondences as well as the essential matrix are obtained. The relative orientation of the current image pair can then be obtained by decomposing the essential matrix.



### 1.6.2.6 Final orientation

The previous steps are accomplished for all images (Extraction of POIs and computation of descriptors) and then for all adjacent image pairs (Computing matching candidates, outlier elimination and relative orientation). In order to have all images registered in a common coordinate system, a canonical orientation is assigned to the first camera. The pose for all subsequent cameras is found by the robust estimation of a scale factor for the baseline vector (joining the camera pairs) using point correspondences which are visible in at least three views. Those chains of corresponding POIs are created within a linking stage.

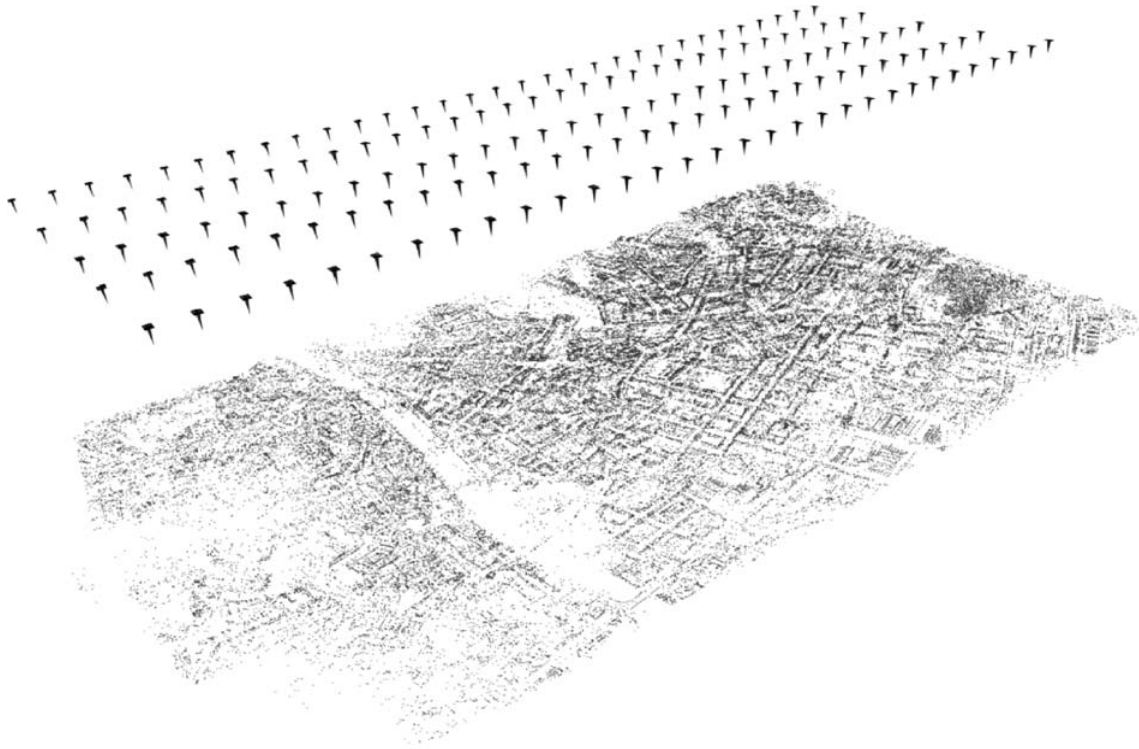


Figure 1.4: An oriented block of five stripes of 31 images each, denoted by small arrows, is shown here. The reconstructed tie points (about 70 000) are displayed as black dots.

Figure 1.4 shows an oriented block of images. The 5 x 31 aerial images are oriented with respect to each other in a local coordinate system. This scene contains 70000 3D points which are shown as black dots. Each 3D point has typically five corresponding tie points - the mean reprojection error is approximately a fifth of a pixel. No additional data



has been used and the whole block of images was oriented without any human interaction.

A bundle block adjustment later refines the relative orientation of the whole block and integrates other data like GPS or ground control information, thus transforming the local coordinates into a real world coordinate system where the z-axis gives the vertical direction.

### 1.6.3 Dense Stereo Matching

Once the aerial triangulation is finished, an area based matching is performed to produce a dense depth estimate for the input images. During the last few years many new dense image matching algorithms were introduced. A good comparison of stereo matching algorithms is given by [58]. Those methods can be roughly categorized into three classes, depending on the set of pixels where the optimization of the disparity values is performed.

Many dense stereo matching algorithms assume that the reconstructed scene can be decomposed into a set of fronto-parallel planes. Most methods therefore start by computing a disparity space volume by sweeping ([20]) a fronto-parallel plane from the front to the back of the scene. The images are warped onto that plane and a matching score is computed for each pixel. The matching score normally operates on a small neighbourhood around each pixel. Popular choices include sum of squared differences, sum of absolute differences and normalized cross correlation. This disparity space volume then contains correlation values for all possible fronto-parallel planes. Depending on the number of sweeping planes, the size of the disparity space volume may become a problem. Therefore the overlapping area of each adjacent image pair is tiled into slightly overlapping regions, in order to allow to keep the data structure in memory at all times to allow fast access.

The task of a later optimization phase consists in finding the best disparity value for each pixel. Advanced optimization techniques consider not only the score distribution at each pixel, but also occlusions and matting at depth discontinuities ([9]).

#### 1.6.3.1 Local Methods

Local methods are among the fastest as they simply choose the locally best disparity value. This makes it often necessary to use larger correlation windows in order to obtain the correct disparity value. Larger aggregation windows, however, imply, that disparity values at depth discontinuities are blurred. On planar surfaces false matches are also still possible, especially if the texture features a repetitive pattern.

The advantage of this method is that the disparity space volume does not need to be kept explicitly in memory. Therefore huge images may be matched at once without tiling.

### 1.6.3.2 Global Methods

Global optimization techniques typically define an objective function which connects all pixels via a neighbourhood relationship ([58], [36]). Assigning a disparity value to one pixels therefore influences the costs for assigning a disparity values to all its neighbours. Often the problem is stated in terms of an energy function as given in Equation (1.2). The function  $\Psi$  then computes the matching score for a certain pixel and disparity value, whereas the function  $\Phi$  penalizes different disparity values of adjacent pixels. Finding the minimum of the above stated energy function is NP-complete for non-submodular smoothness penalty functions ([10]), therefore only approximate solutions can be found in reasonable time. Among the most successful methods are Graph Cuts and Belief Propagation. Because of the complexity and size of the optimization, in general those methods are much slower than methods considering and optimizing a larger neighbourhood system. By using submodular smoothness penalty functions the global optimal configuration can be efficiently computed [52], however, those methods are still slower than semi-global optimization algorithms and even yield worse results according to the Middlebury evaluation suite.

### 1.6.3.3 Semi-Global Methods

The size of images from modern aerial cameras and the amount of images needed to cover even a modestly sized city mandate that the dense image matching algorithm should be fairly fast. The prototype pipeline discussed in this thesis therefore uses a disparity space volume constructed from the 3x3 normalized cross correlation matching score and a cascaded scanline optimization. Both operations are very fast and produce good results.

Like global methods, the scanline optimization incorporates the neighbours for assigning a disparity value to a pixel. However, in order to avoid the performance penalty incurred by global methods, the optimization is only applied along a line, thus reducing the dimensionality of the problem and achieving a higher performance. Applying the scanline optimization only in horizontal direction for example, however, creates unpleasant streaking artefacts (even though there exist possibilities to mitigate this effect, see [74]). Another possibility is to apply the scanline optimization multiple times in different directions ([26], [27]).

The energy optimized by the cascaded scanline optimization along a single line is defined as:

$$E(u) = \sum_{(p,q) \in \mathcal{N}} \alpha_1 T(|u(p) - u(q)| = 1) + \sum_{(p,q) \in \mathcal{N}} \alpha_2 T(|u(p) - u(q)| > 1) + \sum_{p \in \mathcal{P}} S(p, u(p)) , \quad (1.3)$$

where  $\mathcal{P}$  denotes the set of all pixels and  $\mathcal{N}$  the set of adjacent pixels along the current scanline. The function  $S$  computes the pixel-wise data term costs of assigning disparity  $u(p)$  to pixel  $p$  (truncated normalized cross correlation for example). The two other sums make use of a the binary function  $T$  which is one if the argument is true, and 0 otherwise. This way equal disparity values between adjacent pixels are not penalized at all, a small penalty  $\alpha_1$  for small jumps in the disparity value between adjacent pixels is added and a larger penalty  $\alpha_2$  for larger jumps. Along a scanline the minimizer of this function can be efficiently obtained using dynamic programming for example.

Finally a backmatching check ensures that many outliers and bad matches are removed. This check requires that the disparity value for one pixel in the left image and the disparity value for the corresponding pixel in the right image do not differ by more than one.

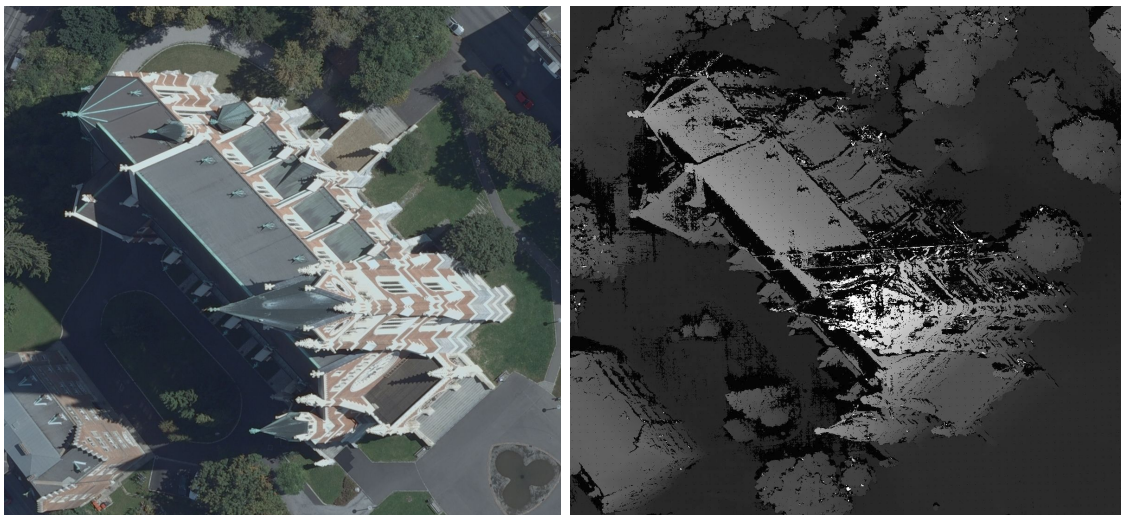


Figure 1.5: On the left side the key image is shown, which is combined with another input images to produce a pixel-synchronous range image (right hand side). Those range images contain outliers and regions where no reliable depth value could be estimated (black pixels in the range image).

Figure 1.5 shows the result of the dense stereo match using the cascaded scanline optimization for a small part of an aerial image.

#### 1.6.4 Range Image Fusion and DTM Generation

An important step in the workflow is the fusion of the individual range images into a consistent DSM. This DSM forms the basis of many other products like ortho images or three dimensional building models.



Figure 1.6: A digital surface model derived by fusing all available range images into one height field using the method proposed in this thesis.

Often the DSM is directly derived from the imagery by a multi-image matching considering all images at once to determine the height of a given location on the ground [78]. A comparison of multi-view with two-frame stereo matching is out of the scope of this work. Two-frame stereo matching with a subsequent range image fusion step has the advantage, that the cost calculation and minimization algorithms are theoretically better understood by the extensive evaluation of the Middlebury evaluation suite.

The individual range images are all converted into a point cloud and projected onto a horizontal reference plane (one could also use or combine them with points obtained from a LIDAR scanner). Independent of the acquisition technique, the task consists then in finding a surface which best approximates those samples.

This problem can be modelled as an energy minimization task given in Equation (1.1). In this case  $u$  is the approximating surface which is sought and  $f$  represents the collected samples for each point on the ground. The functions  $\Phi$  and  $\Psi$  can be different metrics penalizing deviations from zero.

In this thesis the solution is found by using a variational framework. A robust  $L^1$  data term lends resilience to outliers in the set of samples and a regularization term helps to obtain smooth results. An example of a DSM obtained with this approach is depicted in Figure 1.6. A detailed analysis of this work is presented in Chapter 2.

With the availability of the DSM it is then possible to derive a DTM. This is inherently a difficult problem, because the goal is to remove certain objects like trees and buildings from the surface, but leave natural structures like hills untouched. A problematic area for automatic DTM extraction is an extensive, forested hill for example as it is difficult to infer the correct height inside of the forest if the ground height is only available in remote areas at the border of the forest.

An automatic procedure to achieve this is presented in [70]. Their algorithm is also formulated in a variational framework similar to the preceding range image fusion. Optionally they allow a human operator to improve the results by selectively editing areas. The main idea is to first heavily regularize the DSM in order to obtain candidate regions which are most probably part of the DTM. Then all other areas are masked out and a smooth surface is interpolated. The critical step of identifying regions where the DTM height is known is interactive and allows to add and subtract certain regions where the automatic approach failed.

### 1.6.5 Ortho Image Generation

An ortho image is derived from perspective imagery by warping the input images on a reference surface, thus compensating for the distortions caused by the terrain below. Classic methods rely heavily on a simple stitching procedure considering only the DTM, as this surface is often very smooth and thus does not distort the input images too much, thus avoiding to introduce irritating errors and artefacts. On the other side such an approximate geometry does not allow to compensate for leaning buildings resulting in

clearly visible façades or mismatches at stitched tile borders. Using the result of the dense image match, however, it is possible to accurately resample the input images and calculate a true ortho image, which accounts for the recovered geometry of the surface below, thus removing the artefacts encountered in traditional ortho image generation methods. Similar work was published by [48]. Their approach allows to calculate every pixel in the ortho image separately and exploits the redundancy of all available views. The advantage is that every pixel in the ortho image is computed from the reconstructed 3D point and therefore, depending on the quality of the dense image match, no vertical structures like façades are visible.

In the final ortho image it is desirable to have contiguous parts coming from one and the same image and avoid having a salt-and-pepper like structure where single pixels come from different sources as most of the noticeable errors are introduced at image boundaries. The decision where to use which warped patch is obtained by a global optimization which considers the viewing angle and the color differences at the border of the patches as well as the border length. This problem can again be formulated as an energy minimization task as indicated by Equation (1.1). The remaining color mismatches are hidden by blending the warped images using the Helmholtz equation. This strategy, however, does not exploit the redundancy inherent in the imagery.

Another approach consists in applying the variational regularization technique, developed for the range image fusion, to the task to estimate a continuous color surface through the observations. This turns out to filter out moving objects like cars and avoids the time consuming labeling task.

A detailed examination of this topic is presented in Chapter 3.

### 1.6.6 Land Use Classification

The DSM is not only used to calculate color ortho images, but similarly the classification results of the input images can be warped upon the reference surface. For each pixel in the ortho image, this yields a number of probability distributions for their land use - one set of probabilities per input image. Using the information about the local height (computed from the difference between DSM and DTM), it is possible to derive the probabilities for subclasses of the initial classification. Vegetation in areas with a low height for example is usually grassland, whereas solid objects with a high difference between DSM and DTM are candidates for buildings. This separation can be performed with a static sigmoid function or with another classifier. The advantage of the second method is, that it better adapts



Figure 1.7: This Figure shows ortho images composited from multiple input images using the DSM (a) as a reference surfaces. Even though the DSM is not a true 3D data structure, its geometry can be used to generate a high quality true ortho image as the error caused by roof overhangs and similar truly three dimensional structures is minimal. Ortho image (b) is produced by a labeling and stitching approach, whereas (c) was obtained by estimating the continuous color surface.

to local variations, on the other hand it requires additional manual effort to train this classifier.

In the prototype workflow used in this thesis the static approach was used. The probabilities of the vegetation class is split into two complementary probabilities for grassland and trees using the calculated height from the DSM and DTM by multiplying it with two complementary sigmoid functions. The same is done for the solid class which is split into two classes representing street-level objects and buildings.

At the end a multi-class labelling algorithm is used to find a definite assignment of labels which takes the class probabilities into account and their spatial relationship. The regularization adds robustness to outliers and produces smoother results. A discrete energy minimization task as stated in Equation (1.2) is used to obtain the land use classification. The function  $u$  gives the labelling (land use class) for the pixels in this case. A Potts model is used for the smoothness function  $\Phi$ , the function  $\Psi$  computes the data term by taking the negative logarithm of the available probabilities for each land use class. An approximated solution of this problem can be obtained using binary Graph Cuts with alpha expansion moves for example. In Figure 1.8 the result of that process is depicted.

Typename	Panchromatic Resolution	Color and NIR Resolution
UltraCamD	11500 x 7500	2350 x 1700
UltraCamX	14430 x 9420	4810 x 3140
UltraCamX-Prime	17310 x 11310	5770 x 3770

Table 1.1: Current and legacy aerial cameras produced by Vexcel Imaging. Images from all models were used in this work.

### 1.6.7 Building Reconstruction

The result of the land use classification can easily be converted into a building mask by filtering out the respective class. Each four-connected blob in that mask is treated as an individual building block and is fed into the building reconstruction algorithm.

The goal of the building reconstruction step is to obtain accurate and realistic polygonal models of urban buildings.

The method proposed in this thesis does not need any manual intervention and uses only data derived from the original aerial imagery. It combines dense height data together with feature matching to overcome the problem of precise localization of height discontinuities. The nature of this fusion process separates discovery of geometric primitives from the generation of the building model in the spirit of the recover-and-select paradigm ([39]), thus lending robustness to the method as the global optimal configuration is chosen. The integration of the theory of instantaneous kinematics ([55]) allows to elegantly detect and estimate surfaces of revolution which describe a much broader family of roof shapes. A major feature of the proposed method is the possibility to generate various levels of geometric detail. A detailed presentation of this method is given in Chapter 4.

Figure 1.9 illustrates the available data which is used for the reconstruction algorithm and also shows the result of the proposed algorithm.

## 1.7 UltraCam as Image Source

In this work aerial imagery captured by UltraCam devices is used. Those devices are developed by Vexcel Imaging Corporation, which was acquired by Microsoft in 2006. In general their aerial cameras feature an array of high resolution panchromatic sensors which are stitched together to obtain one large image frame. Additionally there is a low resolution color and near-infrared (NIR) sensor, which allow an accurate classification of the image pixels. All sensors have a radiometric resolution of 16 bit.



---

Their most advanced product is the UltraCamX-Prime, which produces panchromatic images with a resolution of 17310 by 11310 pixels. A complete overview of the available aerial cameras is given in Table 1.1.

Still, all algorithms and procedures described in this thesis do not depend on the type of the camera and are universally applicable to other frame sensors as well, given that they provide the necessary radiometric and geometric stability.

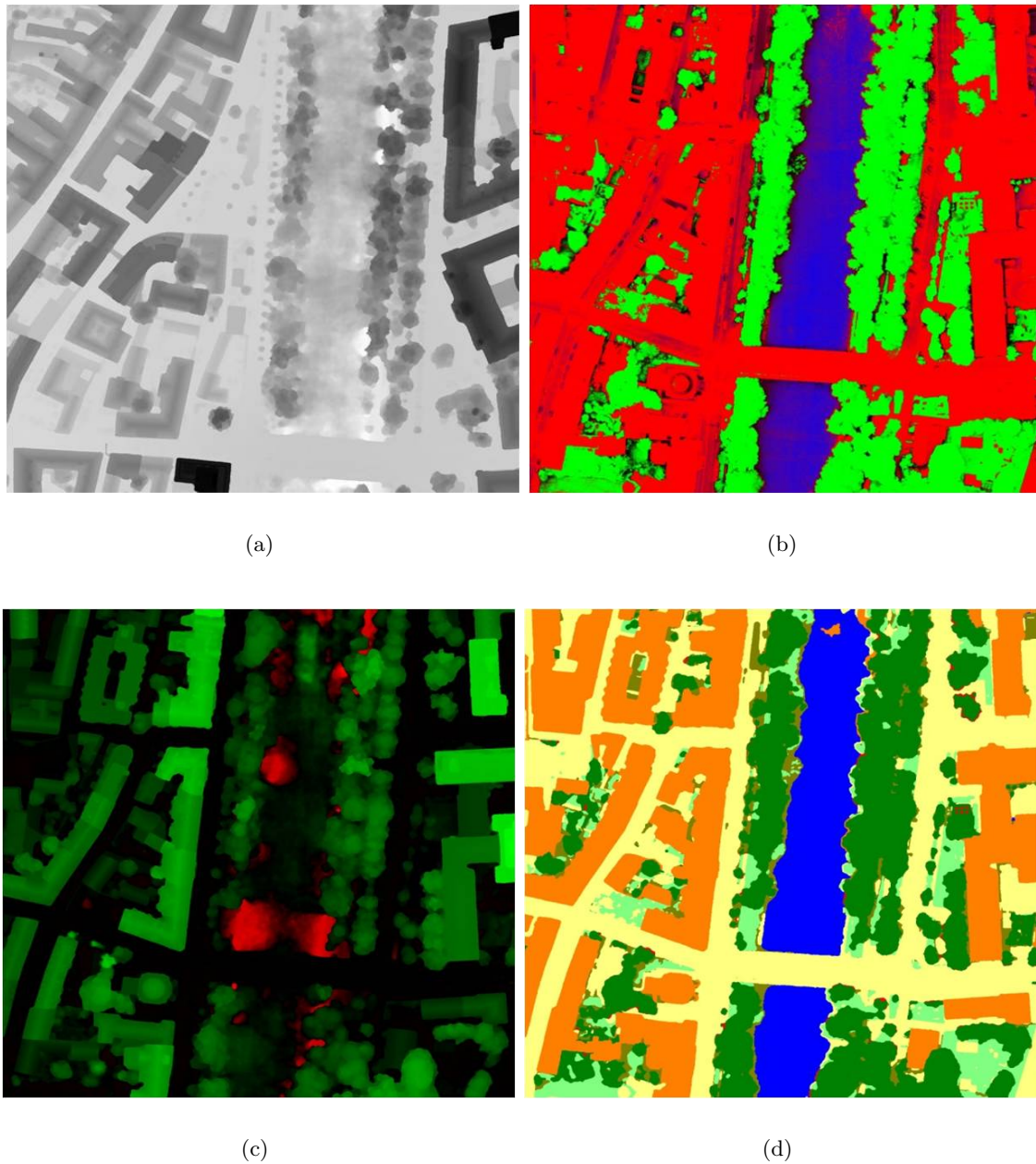


Figure 1.8: The classification results of the RGB-NIR images are combined with the height information to derive a land use classification image. In the top left (a) the height field is shown with the corresponding probabilities (coded in the same fashion as in Figure 1.2) to the right (b). By calculating the difference between DSM and DTM, one obtains the estimated local height (c) - the intensity maps to the absolute value of the height and the color indicates the sign; negative heights only appear in the river because the surface of the water cannot be matched. In (d) the final result is shown.

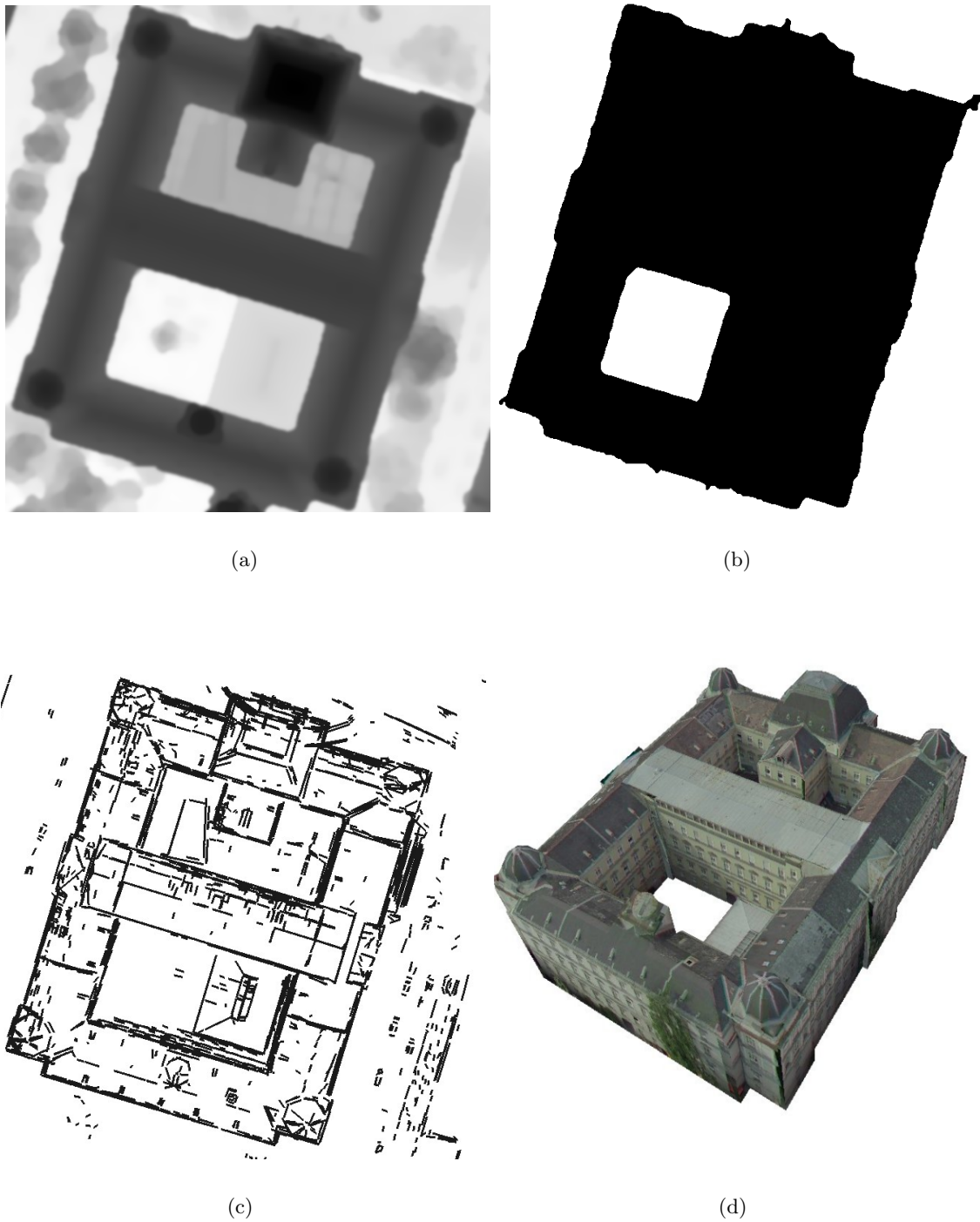


Figure 1.9: These figures depict the data which is used for the reconstruction process: (a) height field, (b) building mask and (c) 3D line segments. Image (d) shows the obtained model by our algorithm.



## Chapter 2

# Robust Range Image Fusion

### Contents

---

<b>2.1</b>	<b>Introduction</b>	<b>29</b>
<b>2.2</b>	<b>Convex Methods for Image Restoration</b>	<b>30</b>
<b>2.3</b>	<b>Generalization to Multiple Observations</b>	<b>33</b>
<b>2.4</b>	<b>Efficient Optimization</b>	<b>37</b>
<b>2.5</b>	<b>Experimental Results</b>	<b>45</b>
<b>2.6</b>	<b>Conclusions</b>	<b>63</b>

---

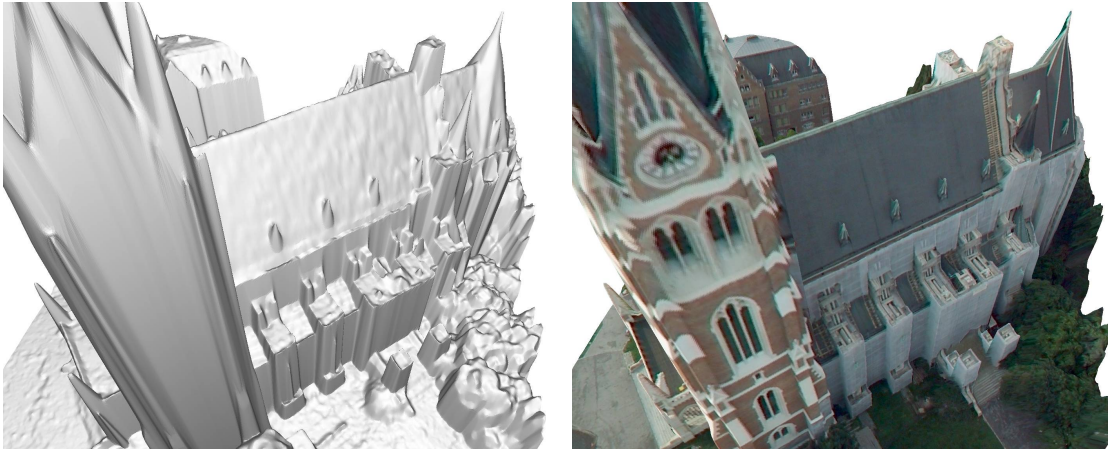
## 2.1 Introduction

Robust integration of range images is an important task for computing high-quality 3D models. The inherent redundancy of digital images often allows to compute several measurements of ranges to one and the same point of a 3D object. Range integration can be either done in  $2\frac{1}{2}D$  (e.g. [21]), or in full 3D (e.g. [76]). For the task of 3D modeling from aerial images (see Figure 2.1), a full 3D range image fusion is often not necessary, nevertheless the redundancy in the imagery induced by the high overlap should be exploited for the reconstruction of a digital surface model (DSM) as such a high quality DSM is the foundation for many subsequent algorithms in aerial imagery processing pipelines.

The approach that is going to be described in this section is based on classical regularization methods for ill-posed inverse problems [69]. The basic idea of these methods is to formulate the solution of a problem as the minimizer of an energy functional, taking into account the smoothness of the solution itself and the distance of the solution to some

given data. Over the years, these methods have been further developed and successfully applied for a number of image restoration tasks [17, 57].

A closely connected approach was presented by Pock et al. in [53]. They utilized the classical Mumford-Shah segmentation model [43] to infer piecewise smooth depth maps from multiple sources. However, being non-convex, their formulation does not guarantee to reach the globally optimal solution. This often leads to unstable results in practice. To overcome this limitation a convex formulation is presented, i.e. an energy functional that allows to compute a global minimizer. It will turn out that methods based on Total Generalized Variation regularization and a robust  $L^1$  data fidelity term lead to the best performance.



(a)

(b)

Figure 2.1: 3D renderings of a church reconstructed by the proposed  $TGV^2-L^1_\delta$  method. The left image shows the untextured surface model, the right image depicts a textured version.

## 2.2 Convex Methods for Image Restoration

The Tikhonov model [69] is one of the earliest - dating back to 1943 - and simplest regularization method used for ill-posed problems. It is defined as the quadratic variational problem

$$\min_u \left\{ E_{Tikhonov} = \frac{1}{2} \int_{\Omega} \alpha |\nabla u|^2 dx + \frac{1}{2} \int_{\Omega} (u - f)^2 dx \right\}, \quad (2.1)$$

where  $\Omega \subset \mathbb{R}^2$  is the image domain,  $f : \Omega \rightarrow \mathbb{R}$  is the observed image function which is assumed to be corrupted by Gaussian noise, and  $u : \Omega \rightarrow \mathbb{R}$  is the sought solution. The free parameter  $\alpha \in \mathbb{R}^+$  is used to control the amount of smoothing in  $u$ . The left term is for regularization and leads to smooth solutions. The right term measures the distance of the solution to the observed data. The parameter  $\alpha$  balances the tradeoff between the data fidelity and the smoothness. Being quadratic in  $u$ , the Tikhonov model poses a simple optimization problem, but it does not perform very well for image restoration tasks. The reason is that the quadratic regularization term leads to an oversmoothing of edges and the quadratic data term does not allow for strong outliers in the observed data.

The history of  $L^1$  estimation procedures goes back to Galileo (1632) and Laplace (1793) and has also been studied by the robust statistics community [29]. However, the first who introduced  $L^1$  estimation methods for image restoration were Rudin, Osher and Fatemi (ROF) in their seminal paper on edge preserving image denoising [57]. In its unconstrained form, the ROF model is defined as the variational model

$$\min_u \left\{ E_{ROF} = \int_{\Omega} \alpha |\nabla u| dx + \frac{1}{2} \int_{\Omega} (u - f)^2 dx \right\}, \quad (2.2)$$

where the left term is the so-called Total Variation of  $u$ . The main advantage of the ROF model is, that it allows for sharp discontinuities in the solution. Computing the minimizer of the ROF model is a challenging problem due to the non-differentiability of the Total Variation term. The lagged diffusivity method of Vogel and Oman [72] and Chambolle's duality based projection method [14] are two out of many algorithms how to solve this problem.

One disadvantage of the Total Variation regularization is the so-called stair-casing effect, a phenomenon which leads to block-like artefacts in the solution  $u$ . For reconstructions this is undesirable, since it leads to piecewise constant approximations of slanted surfaces. A simple possibility to alleviate these artefacts is to replace the  $L^1$  norm in the Total Variation term by the Huber norm:

$$Huber_{\varepsilon}(x) = |x|_{\varepsilon} = \begin{cases} \frac{x^2}{2\varepsilon} & : |x| \leq \varepsilon \\ |x| - \frac{\varepsilon}{2} & : |x| > \varepsilon \end{cases}. \quad (2.3)$$

This norm is quadratic for small values and linear for large values. Hence small discontinuities are smoothed because of the quadratic part of the norm and only for larger jumps the norm behaves like the robust  $L^1$  norm. The parameter  $\varepsilon$  can therefore be used to control the maximally allowed slope of a surface before it is modelled as a discontinuity.

ity. This norm has also been shown to be an excellent choice in the context of medical imaging [32].

The study of the ROF model is therefore continued by focusing on the following variant:

$$\min_u \left\{ E_{ROF_\varepsilon} = \int_{\Omega} \alpha |\nabla u|_\varepsilon dx + \frac{1}{2} \int_{\Omega} (u - f)^2 dx \right\}, \quad (2.4)$$

This formulation is now able to handle discontinuities and avoid stair-casing artefacts to a certain degree. Outliers in the observations, however, still heavily skew the reconstruction because of the quadratic norm in the data term.

The TV- $L^1$  model [3, 17, 45] is obtained from the ROF model by replacing the  $L^2$  norm in the data term with the  $L^1$  norm.

$$\min_u \left\{ E_{TV-L^1} = \int_{\Omega} \alpha |\nabla u| dx + \int_{\Omega} |u - f| dx \right\}. \quad (2.5)$$

Although this change seems to be minor, it offers some desirable improvements. First, it turns out that the TV- $L^1$  model is more effective than the ROF model in removing speckle noise [45]. Second, the TV- $L^1$  model is contrast invariant. This makes it useful for scale-driven feature selection [19] and denoising of shapes [46].

Again the Huber norm in the smoothness term can be used to avoid piecewise constant solutions. Similarly, the  $L^1$  norm in the data term can be replaced by an Huber norm. The reason is that for small deviations in the function  $f$ , the quadratic part of the Huber norm leads to an averaging-like behaviour, whereas for large deviations, the linear part leads to a median-like behaviour:

$$\min_u \left\{ E_{TV_\varepsilon-L_\delta^1} = \int_{\Omega} \alpha |\nabla u|_\varepsilon dx + \int_{\Omega} |u - f|_\delta dx \right\}. \quad (2.6)$$

Restricting the spatial regularization to the first order (only the first derivative is used) without any bias for smaller discontinuities has one serious disadvantage: The minimizer of the ROF and TV- $L^1$  model exhibit a preference for piecewise constant functions. This so called "stair-casing" effect produces noticeable artefacts in sloped regions. Resorting to the Huber norm alleviates this problem to a certain degree for, but only by trading off the robustness of the  $L^1$  with the smoothing property of the  $L^2$  norm. Surfaces with a large slope might not be affected by this change and still exhibit stair-casing.

Recently the theory of Total Variation was extended to higher orders [11]. This allows to formulate an energy model where only changes in the second derivative are penalized for example, yielding a model which prefers piecewise affine instead of piecewise constant



solutions. The energy functional for the Total Generalized Variation of the second order with a robust  $L^1$  data term ( $TGV^2 - L_\delta^1$ ) looks like this:

$$\min_u \left\{ E_{TGV^2-L_\delta^1} = \int_\Omega \alpha_1 |\nabla u - v| dx + \int_\Omega \alpha_0 |\nabla v| dx + \int_\Omega |u - f|_\delta dx \right\}, \quad (2.7)$$

with an auxiliary vector field  $v$  upon which the spatial regularization is imposed and coupled to the gradient vector field of the sought solution. Higher order regularization is achieved by adding more auxiliary tensor fields, each one coupled to the gradients of the previous one and only imposing the spatial regularization upon the last one. In most cases the advantage of higher order regularization yields diminishing returns [11].

## 2.3 Generalization to Multiple Observations

In the last section four basic models for image restoration were introduced: The Tikhonov model, the ROF model, the  $TV_\varepsilon-L_\delta^1$  model and the  $TGV^2-L_\delta^1$  model. In this section a probabilistic reasoning is presented why the  $L^1$  norm is superior to the  $L^2$  norm for multiple observations, then the modified energy models are presented.

### 2.3.1 Probabilistic Modelling of Multiple Data Terms

In theory a quadratic penalty function yields the best estimator for observations contaminated with Gaussian noise: Assuming a true value of  $u$  and that a sample is drawn from a Gaussian distribution centred at  $u$  with a certain variance  $\sigma$ , the probability of observing  $f$  is

$$p(f|u) = \frac{1}{\sqrt{2\pi\sigma^2}} e^{-\frac{(f-u)^2}{2\sigma^2}}. \quad (2.8)$$

Bayes theorem then allows us to link the conditional probabilities and thus express the probability of  $u$  observing a value  $f$  as

$$p(u|f)p(f) = p(f|u)p(u) \quad (2.9)$$

$$p(u|f) = \frac{p(u)p(f|u)}{p(f)}. \quad (2.10)$$

The best estimate of  $u$  is given by the maximum a posteriori probability (MAP) of  $u$ .

If we do not impose any apriori distribution on  $f$  and  $u$  (in order to restrict those values to a certain range for example or prefer certain model parameters of  $u$ ), the MAP estimate of  $u$  simplifies to

$$\arg \max_u p(u|f) = \arg \max_u p(f|u) . \quad (2.11)$$

For multiple observations  $f_i$ , the MAP estimate of  $u$  is obtained by maximizing the product of the individual condition probabilities:

$$\arg \max_u p(u|f_i) = \arg \max_u \prod_i p(f_i|u) = \arg \max_u \prod_i k_1 e^{-(f_i-u)^2 k_2} , \quad (2.12)$$

with  $k_1 = \frac{1}{\sqrt{2\pi\sigma^2}}$  and  $k_2 = \frac{1}{2\sigma^2}$ . Taking the logarithm of the product as well as dropping the constant factors  $k_1$  and  $k_2$  does not change the position of the maximum:

$$\arg \max_u \prod_i k_1 e^{-k_2(f_i-u)^2} = \arg \max_u e^{\sum_i -k_2(f_i-u)^2} = \quad (2.13)$$

$$= \arg \max_u \sum -k_2(f_i - u)^2 = \arg \max_u \sum -(f_i - u)^2 . \quad (2.14)$$

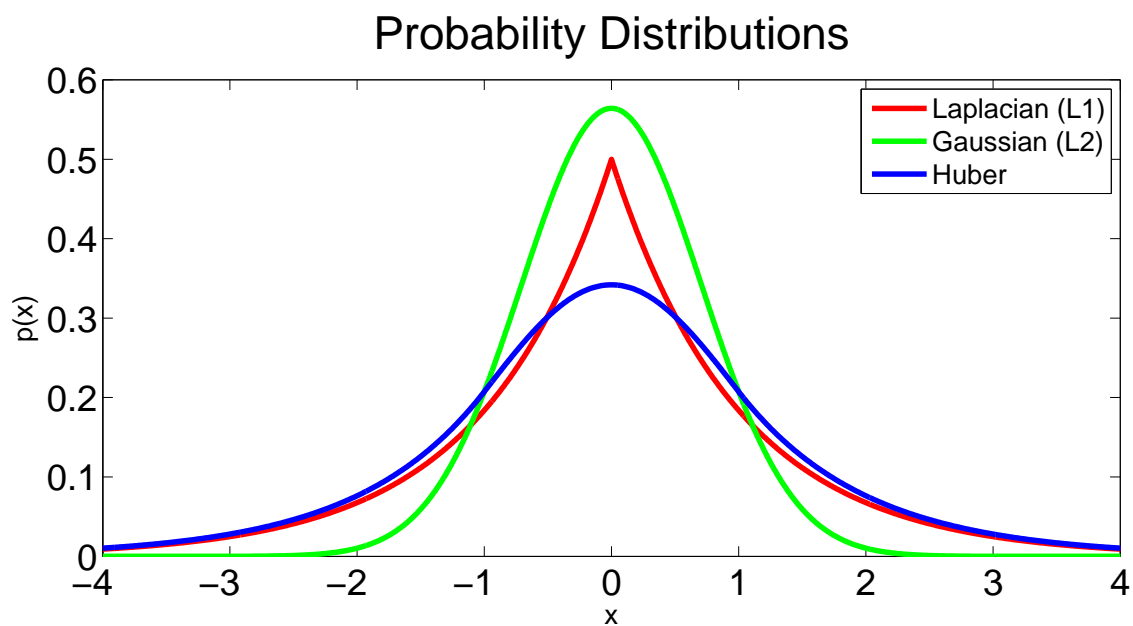
Flipping the sign and taking the minimum instead, yields something already very similar to the quadratic penalty function:

$$\arg \max_u \sum -(f_i - u)^2 = \arg \min_u \sum (f_i - u)^2 . \quad (2.15)$$

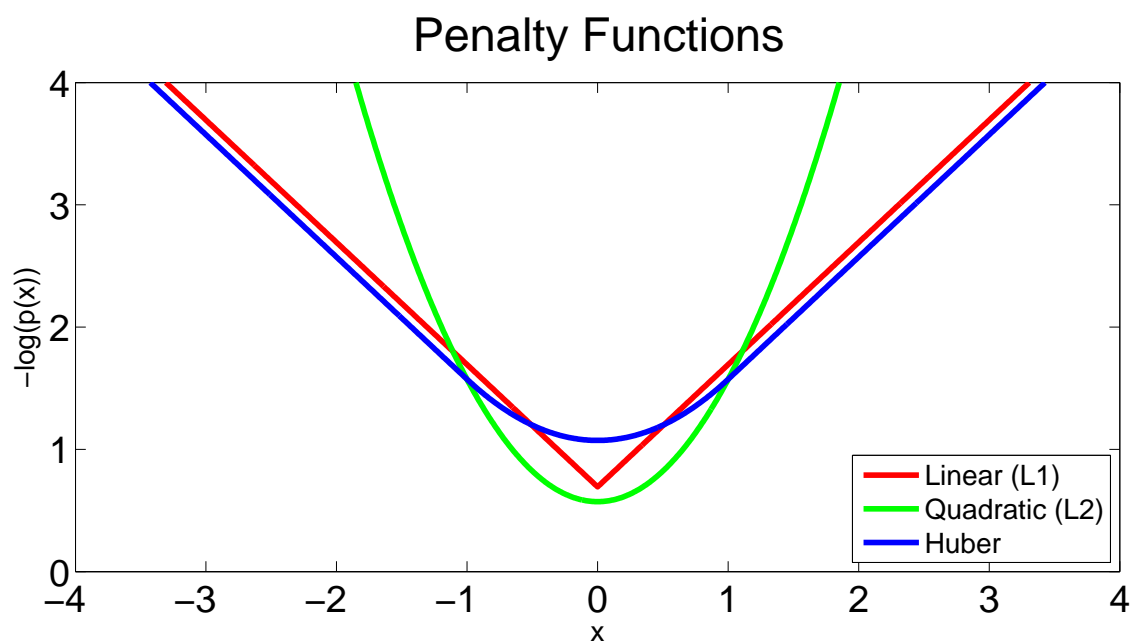
One can now observe, that the value which maximizes the aposteriori probability of  $u$  given the observations  $f_i$  is the same which minimizes the quadratic penalty function.

In reality the observations are rarely only contaminated by Gaussian noise. Often one has to deal with outliers or even cope with completely different distributions of the drawn samples. In theory, the optimal choice of the penalty function can be obtained, if the distortions of the samples is known, by taking the negative logarithm of their distribution. This is often difficult as this distribution is not known in advance and also because for complex distributions the resulting penalty function can not be easily optimized. The  $L^1$  estimator is the border case between efficient (convex) optimization and robustness. The corresponding error distribution in that case is the Laplacian distribution,

$$p(f|u) = k_1 e^{-k_2|f-u|} , \quad (2.16)$$



(a)



(b)

Figure 2.2: Probability distributions like those depicted in (a) have a corresponding analytical penalty norm (linear, quadratic and Huber), as shown in (b).

where  $k_1$  and  $k_2$  are again suitable constant factors to make this function a probability distribution. The robustness against outliers can be explained by the stronger tail of this distribution compared to the Gaussian noise: Outliers are then much more probable and thus receive a lower weight. These relationships are illustrated in Figure 2.2. One notices that the probability of outliers is very low for the Gaussian ( $L^2$ ) distribution (and thus their corresponding weight is very high as indicated by the penalizer). The  $L^1$  norm allows for a higher probability of outliers and thus the weight is much lower.

The same reasoning applies to the penalty function of the derivatives, even though here the assumption that the gradients stem from a Gaussian distribution is already thwarted by looking at natural images. In a Gaussian distribution the probability for strong gradients is too low, such that they are heavily penalized and thus avoided. In natural images, however, those strong gradients have a higher probability than predicted by a simple Gaussian model. The probability distribution corresponding to a  $L^1$  or Huber norm much better matches those requirements.

### 2.3.2 Extending the Energy Models to Multiple Observations

The extension of the the four different energy models to multiple observations is a straight forward summation of the distances to the observed data points. The metric how to compute the individual distance obviously depends on the chosen energy model.

For the Tikhonov energy from Equation (2.1) the modified energy model becomes

$$E_{Tikhonov} = \frac{1}{2} \int_{\Omega} \alpha |\nabla u|^2 dx + \frac{1}{2} \int_{\Omega} \sum_{i=1}^N (u - f_i)^2 dx, \quad (2.17)$$

where the functions  $f_i : \Omega \rightarrow \mathbb{R}$ ,  $i = 1 \dots N$  represent the individual observations.

The extension of the three other energy models to multiple observations goes in analogy to the Tikhonov model.

$$E_{ROF_{\varepsilon}} = \int_{\Omega} \alpha |\nabla u|_{\varepsilon} dx + \frac{1}{2} \int_{\Omega} \sum_{i=1}^N (u - f_i)^2 dx. \quad (2.18)$$

$$E_{TV_{\varepsilon-L^1_{\delta}}} = \int_{\Omega} \alpha |\nabla u|_{\varepsilon} dx + \int_{\Omega} \sum_{i=1}^N |u - f_i|_{\delta} dx. \quad (2.19)$$

$$E_{TGV^{2-L^1_{\delta}}} = \int_{\Omega} \alpha_1 |\nabla u - v| dx + \int_{\Omega} \alpha_0 |\nabla v| dx + \int_{\Omega} \sum_{i=1}^N |u - f_i|_{\delta} dx, \quad (2.20)$$

## 2.4 Efficient Optimization

Convex analysis and optimization is a field with a long history. Accordingly, a host of algorithms exists for minimizing convex functionals. An excellent overview and analysis of such algorithms is given in [16]. Among the approaches presented there, a primal-dual approach is one of the most efficient and versatile algorithms. The algorithm can be easily adapted to each energy functional and shows excellent convergence behaviour. In [51] it is also successfully used for minimizing the Mumford-Shah functional. This scheme can also be applied to all energy models described in the last section.

This algorithm states that a minimization problem of the form

$$\min_x \{F(Ax) + G(x)\} \quad (2.21)$$

with  $F$  and  $G$  being proper convex functions, can be solved with the iterative update steps

$$y^{n+1} = (I + \sigma \partial F^*)^{-1} (y^n + \sigma A \bar{x}^n) \quad (2.22)$$

$$x^{n+1} = (I + \tau \partial G)^{-1} (x^n - \tau A^* y^{n+1}) \quad (2.23)$$

$$\bar{x}^{n+1} = 2x^{n+1} - x^n, \quad (2.24)$$

where  $A^*$  is the adjoint operator of  $A$  and  $F^*$  is the Legendre-Fenchel dual of  $F$ . The step sizes  $\tau$  and  $\sigma$  have to be chosen such that  $\tau\sigma|A| < 1$ .

### 2.4.1 Legendre-Fenchel Conjugate

The Legendre-Fenchel conjugate is defined [56] as

$$f^*(x^*) = \sup_x \{\langle x, x^* \rangle - f(x)\} . \quad (2.25)$$

This transformation can be understood as giving the largest offset of a hyperplane with the normal vector specified by  $x^*$  touching the function  $f(x)$ . The Legendre-Fenchel conjugate has the remarkable property, that the biconjugate of a convex function is equivalent to the original function:

$$f^{**}(x^{**}) = \sup_{x^*} \{\langle x^*, x^{**} \rangle - f^*(x^*)\} = f(x) . \quad (2.26)$$

This transformation is illustrated with the  $\alpha$ -weighted Huber norm as an example, because it will play an important role in the energy functions which are examined in this work. The Legendre-Fenchel conjugate for the  $\alpha$ -weighted Huber norm is obtained by applying its definition from Equation (2.3) to Equation (2.25):

$$\alpha \text{Huber}_{\varepsilon}^*(x^*) = \text{Huber}_{\varepsilon, \alpha}^*(x^*) = \sup_x \begin{cases} \langle x, x^* \rangle - x^2 \frac{\alpha}{2\varepsilon} & : |x| \leq \varepsilon \\ \langle x, x^* \rangle - \alpha|x| + \frac{\alpha\varepsilon}{2} & : |x| > \varepsilon \end{cases} . \quad (2.27)$$

For  $|x^*|$  larger than  $\alpha$  the function grows without bounds and thus the supremum is  $\infty$ . Now only the case for values of  $|x^*|$  less or equal to  $\alpha$  has to be analysed. The maximum value of the quadratic portion of that function is obtained by setting its derivative to zero and solving for  $x$ :

$$x^* - x \frac{\alpha}{\varepsilon} = 0 \quad (2.28)$$

$$x = x^* \frac{\varepsilon}{\alpha} . \quad (2.29)$$

The quadratic part of the Huber norm is only valid for  $|x| \leq \varepsilon$ , which gives rise to the side condition

$$x = x^* \frac{\varepsilon}{\alpha} \leq \varepsilon , \quad (2.30)$$

and thus

$$x^* \leq \alpha , \quad (2.31)$$

which coincides with the premise for the second case and therefore already completely covers it. The Legendre-Fenchel dual of the  $\alpha$ -weighted Huber norm therefore is

$$\text{Huber}_{\varepsilon, \alpha}^*(x^*) = \begin{cases} x^{*2} \frac{\varepsilon}{2\alpha} & : |x^*| \leq \alpha \\ \infty & : |x^*| > \alpha \end{cases} , \quad (2.32)$$

or, alternatively, introducing the indicator function  $\mathcal{I}_S$ , which takes the value of 0 for points in the set  $S$  and  $\infty$  otherwise,

$$\text{Huber}_{\varepsilon, \alpha}^*(x^*) = x^{*2} \frac{\varepsilon}{2\alpha} + \mathcal{I}_{\{|x^*| \leq \alpha\}} . \quad (2.33)$$

Combining Equation (2.26) and (2.33) the Huber norm can therefore be rewritten as

$$\alpha \text{Huber}_\varepsilon(x) = \alpha |x|_\varepsilon = \sup_{x^*} \left\{ \langle x, x^* \rangle - x^{*2} \frac{\varepsilon}{2\alpha} - \mathcal{I}_{\{|x^*| \leq \alpha\}} \right\} . \quad (2.34)$$

One can transform the quadratic norm in a similar fashion and derive the equality

$$\alpha x^2 = \sup_{x^*} \left\{ \langle x, x^* \rangle - \frac{x^{*2}}{4\alpha} \right\} . \quad (2.35)$$

### 2.4.2 Primal-Dual Gap

As stated in the previous section, each convex function can be rewritten as the biconjugate of itself using the Legendre-Fenchel transformation. Applying this transformation to Equation (2.21) yields

$$\min_x \max_y \{ \langle Ax, y \rangle - F^*(y) + G(x) \} . \quad (2.36)$$

The operator  $A$  can swap sides by using its adjoint operator  $A^*$  which results in

$$\min_x \max_y \{ \langle x, A^*y \rangle + G(x) - F^*(y) \} \quad (2.37)$$

after a slight rearrangement of the terms. Negating all terms twice does not change the result, but allows further transformations. Additionally the order of minimization and maximization can be swapped as we are dealing with a convex problem:

$$\max_y \min_x \{ -(\langle x, -A^*y \rangle - G(x) + F^*(y)) \} . \quad (2.38)$$

Calculating the minimum with respect to  $x$  is equivalent to computing the maximum of the negated function:

$$\max_y \max_x \{ \langle x, -A^*y \rangle - G(x) + F^*(y) \} . \quad (2.39)$$

The first two terms are now exactly the Legendre-Fenchel dual of  $G^*(-A^*y)$  as according to Equation (2.26) the biconjugate of a convex function is the function itself,  $G^{**} = G$ . The original primal problem is now transformed into its dual equivalent

$$\max_y \{ G^*(-A^*y) + F^*(y) \} . \quad (2.40)$$

The difference between the primal and dual formulation for a given set of variables is

called the primal-dual gap and is always non-negative:

$$\mathcal{G}(x, y) = F(Ax) + G(x) - G^*(-A^*y) - F^*(y) \geq 0 \quad (2.41)$$

It becomes zero if both the primal and dual problem reach their respective extremal values. It can be shown that the primal-dual gap converges with  $O(1/n)$  to zero for the update steps given in Equation 2.22. The proof of convergence of this scheme bases on [44] and is given in detail in [22] and [16].

The algorithm iteratively updates the primal and dual variables. First, the dual variable is updated via an explicit and implicit gradient ascent. Then, a similar procedure is applied to the primal variable, with the only difference that a gradient descent is performed instead of a gradient ascent.

The implicit gradient descent is performed by finding the argument minimizing a proximal operator, defined as

$$(I + \lambda\partial f)^{-1}(z) = \arg \inf_y \left\{ \frac{1}{2\lambda} (y - z)^2 + f(y) \right\}. \quad (2.42)$$

This infimum can be obtained by setting the derivative to zero:

$$\frac{1}{\lambda} (y - z) + \partial f(y) = 0, \quad (2.43)$$

which simplifies to

$$y - z + \lambda\partial f(y) = 0 \quad (2.44)$$

$$(I + \lambda\partial f)(y) = z \quad (2.45)$$

and finally yields the expression used in Equation (2.42) by bringing the operator on the other side of the equation

$$y = (I + \lambda\partial f)^{-1}(z) \quad (2.46)$$

### 2.4.3 Tikhonov Energy Model

Applying the theory of the Legendre-Fenchel conjugation to the Tikhonov energy model from Equation (2.17), replaces the spatial regularization term with its dual biconjugate derived in Equation (2.35). Traditionally this dual variable is denoted by  $\mathbf{p}$ :



$$E_{Tikhonov} = \sup_{\mathbf{p}} \left\{ \int_{\Omega} \langle \nabla u, \mathbf{p} \rangle - \frac{\mathbf{p}^2}{2\alpha} dx + \frac{1}{2} \int_{\Omega} \sum_{i=1}^N (u - f_i)^2 dx \right\}, \quad (2.47)$$

Before a minimizer  $u$  of Equation (2.47) can be computed using the update scheme from Equation (2.22), the proximal operators have to be calculated for the primal and dual updates. This is done by solving Equation (2.42) for the primal and dual updates

$$\mathbf{p}^{n+1} = \arg \inf_{\mathbf{p}} \left\{ \frac{1}{2\sigma} (\mathbf{p} - \bar{\mathbf{p}})^2 + \frac{\mathbf{p}^2}{2\alpha} \right\} \quad (2.48)$$

$$u^{n+1} = \arg \inf_u \left\{ \frac{1}{2\tau} (u - \bar{u})^2 + \frac{1}{2} \sum_{i=1}^N (u - f_i)^2 \right\}. \quad (2.49)$$

The minima of both functionals are obtained by setting their derivatives to zero,

$$\frac{1}{\sigma} (\mathbf{p} - \bar{\mathbf{p}}) + \frac{\mathbf{p}}{2\alpha} = 0 \quad (2.50)$$

$$\frac{1}{\tau} (u - \bar{u}) + \sum_{i=1}^N (u - f_i) = 0. \quad (2.51)$$

The minimizers can then be calculated by sorting the terms, which yields

$$\mathbf{p} = \frac{\bar{\mathbf{p}}}{1 + \frac{\sigma}{2\alpha}} \quad (2.52)$$

$$u = \frac{\bar{u} + \tau \sum_{i=1}^N f_i}{1 + \tau N}. \quad (2.53)$$

Thus the update rules for the Tikhonov energy model are

$$\mathbf{p}^{n+1} = \frac{\mathbf{p}^n + \sigma \nabla \bar{u}^n}{1 + \frac{\sigma}{2\alpha}} \quad (2.54)$$

$$u^{n+1} = \frac{u^n - \tau \nabla \cdot \mathbf{p}^n + \tau \sum_{i=1}^N f_i}{1 + \tau N} \quad (2.55)$$

$$\bar{u}^{n+1} = 2u^{n+1} - u^n \quad (2.56)$$

#### 2.4.4 $ROF_\varepsilon$ Energy Model

The primal dual formulation of the  $ROF_\varepsilon$  energy model is also obtained by replacing the smoothness term of Equation (2.18) with its biconjugate derived in Equation (2.34).

$$E_{ROF_\varepsilon} = \sup_{\mathbf{p}} \left\{ \int_{\Omega} \langle \nabla u, \mathbf{p} \rangle - \mathbf{p}^2 \frac{\varepsilon}{2\alpha} - \mathcal{I}_{\{|\mathbf{p}| \leq \alpha\}} dx + \frac{1}{2} \int_{\Omega} \sum_{i=1}^N (u - f_i)^2 dx \right\}. \quad (2.57)$$

One notices that this functional has the same data term as the Tikhonov energy model, and therefore the update rules of the optimization scheme for the primal variable can be reused without modifications. The proximal operator for the dual update, however, has to be recalculated.

$$\mathbf{p}^{n+1} = \arg \inf_{\mathbf{p}} \left\{ \frac{1}{2\sigma} (\mathbf{p} - \bar{\mathbf{p}})^2 + \mathbf{p}^2 \frac{\varepsilon}{2\alpha} + \mathcal{I}_{\{|\mathbf{p}| \leq \alpha\}} \right\} \quad (2.58)$$

The indicator function  $\mathcal{I}_{\{|\mathbf{p}| \leq \alpha\}}$  is treated in a special way by restricting  $\mathbf{p}$  to the ball with radius  $\alpha$ . The solution cannot be outside of that ball as the indicator function would give an infinite penalty there. Setting the first derivative to zero yields the minimizer - a re-projection onto that unit ball afterwards guarantees that the solution is not outside.

$$\frac{1}{\sigma} (\mathbf{p} - \bar{\mathbf{p}}) - \mathbf{p} \frac{\varepsilon}{\alpha} = 0 \quad (2.59)$$

Separating the variables and applying the re-projection yields the solution for the proximal operator:

$$\mathbf{p} = \Pi_{\alpha} \frac{\bar{\mathbf{p}}}{1 + \frac{\varepsilon\sigma}{\alpha}}. \quad (2.60)$$

The update steps for the dual formulation of the ROF energy model are therefore

$$\mathbf{p}^{n+1} = \Pi_{\alpha} \frac{\mathbf{p}^n + \sigma \nabla \bar{u}^n}{1 + \frac{\varepsilon\sigma}{\alpha}} \quad (2.61)$$

$$u^{n+1} = \frac{u^n - \tau \nabla \cdot \mathbf{p}^n + \tau \sum_{i=1}^N f_i}{1 + \tau N} \quad (2.62)$$

$$\bar{u}^{n+1} = 2u^{n+1} - u^n. \quad (2.63)$$

### 2.4.5 $TV_\varepsilon$ - $L_\delta^1$ Energy Model

Similar to the ROF model, the Lagrange-Fenchel duality can be used to rewrite the energy functional of the  $TV_\varepsilon$ - $L_\delta^1$  model. This time, multiple separate dual variables have to be introduced to account for the multiple data terms, because those are also not differentiable at zero. The indicator functions are this time already incorporated into the formulation of the supremum by restricting the domain of the dual variables to balls with a certain radius, as explained in the case of the  $ROF_\varepsilon$  energy model:

$$E_{TV_\varepsilon-L_\delta^1} = \sup_{\substack{|\mathbf{p}| \leq \alpha \\ |r_i| \leq 1}} \left\{ \int_{\Omega} \langle \nabla u, \mathbf{p} \rangle - \mathbf{p}^2 \frac{\varepsilon}{2\alpha} dx + \int_{\Omega} \sum_{i=1}^N \langle u - f_i, r_i \rangle - r_i^2 \frac{\delta}{2} dx \right\}. \quad (2.64)$$

The major difference is the appearance of multiple dual variables. Accordingly there are also more update steps, one for each dual variable. The solutions of the proximal operator for  $\mathbf{p}$  and  $r_i$ , however, are already known from Equation (2.60). The update for  $u$  is again derived from the definition of the proximal operator:

$$u^{n+1} = \arg \inf_u \left\{ \frac{1}{2\tau} (u - \bar{u})^2 + \sum_{i=1}^N \langle u - f_i, r_i \rangle \right\}. \quad (2.65)$$

The standard procedure of setting the first derivative to zero yields

$$\frac{1}{\tau} (u - \bar{u}) + \sum_{i=1}^N r_i = 0, \quad (2.66)$$

and by separating the variable

$$u = \bar{u} - \tau \sum_{i=1}^N r_i. \quad (2.67)$$

The updates steps for the  $TV_\varepsilon$ - $L_\delta^1$  energy model are therefore

$$\mathbf{p}^{n+1} = \Pi_{\alpha} \frac{\mathbf{p}^n + \sigma \nabla \bar{u}^n}{1 + \frac{\varepsilon \sigma}{\alpha}} \quad (2.68)$$

$$r^{n+1} = \Pi_1 \frac{r^n + \sigma(\bar{u} - f_i)}{1 + \delta \sigma} \quad (2.69)$$

$$u^{n+1} = u^n + \tau \left( \nabla \cdot \mathbf{p}^n - \sum_{i=1}^N r_i \right) \quad (2.70)$$

$$\bar{u}^{n+1} = 2u^{n+1} - u^n. \quad (2.71)$$

#### 2.4.6 $TGV^2-L_{\delta}^1$ Energy Model

Rewriting the  $TGV^2-L_{\delta}^1$  energy model using the Legendre-Fenchel duality introduces the tensor matrix  $\mathbf{q}$  as the dual to the auxiliary vector field  $\mathbf{v}$ . Additionally, all the dual variables used for the  $TV_{\varepsilon}-L_{\delta}^1$  energy model are also necessary. The indicator functions are again incorporated into the domain over which the supremum is calculated into order to make the equation more readable.

$$E_{TGV^2-L_{\delta}^1} = \sup_{\substack{|\mathbf{p}| \leq \alpha_1 \\ |\mathbf{q}| \leq \alpha_0 \\ |r_i| \leq 1}} \left\{ \int_{\Omega} \langle \nabla u - \mathbf{v}, \mathbf{p} \rangle dx + \int_{\Omega} \langle \nabla \mathbf{v}, \mathbf{q} \rangle dx + \int_{\Omega} \sum_{i=1}^N \langle u - f_i, r_i \rangle - r_i^2 \frac{\delta}{2} dx \right\} \quad (2.72)$$

Most of the updates steps for minimizing Equation (2.72) are similar to the  $TV_{\varepsilon}-L_{\delta}^1$  model, the other ones are derived by applying the templates from Equation (2.22) to the solutions of the proximal operator already seen in the cases of the other energy models.

$$\mathbf{p}^{n+1} = \Pi_{\alpha_1} \mathbf{p}^n + \sigma (\nabla \bar{u}^n - \bar{\mathbf{v}}^n) \quad (2.73)$$

$$\mathbf{q}^{n+1} = \Pi_{\alpha_0} \mathbf{q}^n + \sigma \nabla \bar{\mathbf{v}}^n \quad (2.74)$$

$$r^{n+1} = \Pi_1 \frac{r^n + \sigma(\bar{u}^n - f_i)}{1 + \delta \sigma} \quad (2.75)$$

$$u^{n+1} = u^n + \tau \left( \nabla \cdot \mathbf{p}^n - \sum r_i^n \right) \quad (2.76)$$

$$\mathbf{v}^{n+1} = \mathbf{v}^n + \tau (\mathbf{p}^n + \nabla \cdot \mathbf{q}^n) \quad (2.77)$$

$$\bar{u}^{n+1} = 2u^{n+1} - u^n \quad (2.78)$$

$$\bar{\mathbf{v}}^{n+1} = 2\mathbf{v}^{n+1} - \mathbf{v}^n. \quad (2.79)$$

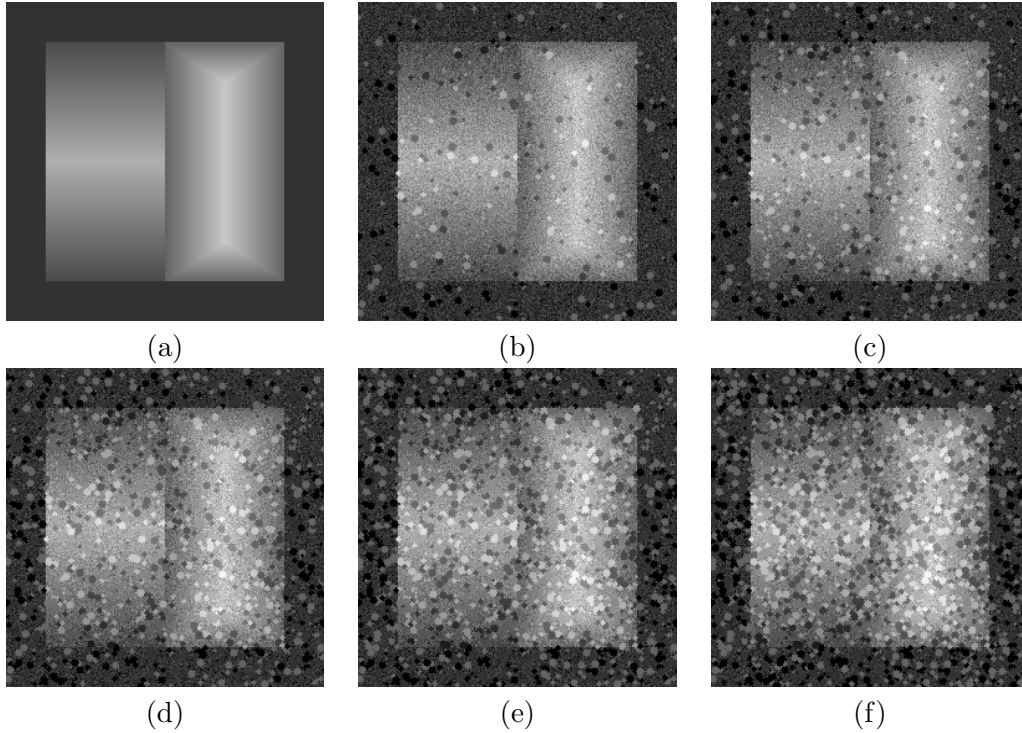


Table 2.1: Illustration of the synthetic dataset used to evaluate and compare the energy models. Image (a) depicts the ground truth, the other images (b) - (f) are distorted samples by adding Gaussian noise and 10%, 20%, 30%, 40% and 50% outliers, respectively.

## 2.5 Experimental Results

A quantitative evaluation of the results of the different energy models is not trivial as each of them optimizes different norms for the data and smoothness term. Therefore the energy models are evaluated with a synthetic dataset and the smallest difference between the obtained approximation and the known ground truth is measured. Then they are applied to a real world dataset obtained from aerial imagery, which can only be judged qualitatively as no ground truth is available. Finally it is shown that the proposed energy models can also be applied to improve range images for a single image pair by fusing the results of various dense image matching algorithms.

### 2.5.1 Synthetic Dataset

The synthetic dataset consists of an idealised building block with roof shapes which are fairly common in urban environments. On the left hand side there is a gabled roof, whereas on the right hand side a hip roof is modelled. The dynamic range of this synthetic signal

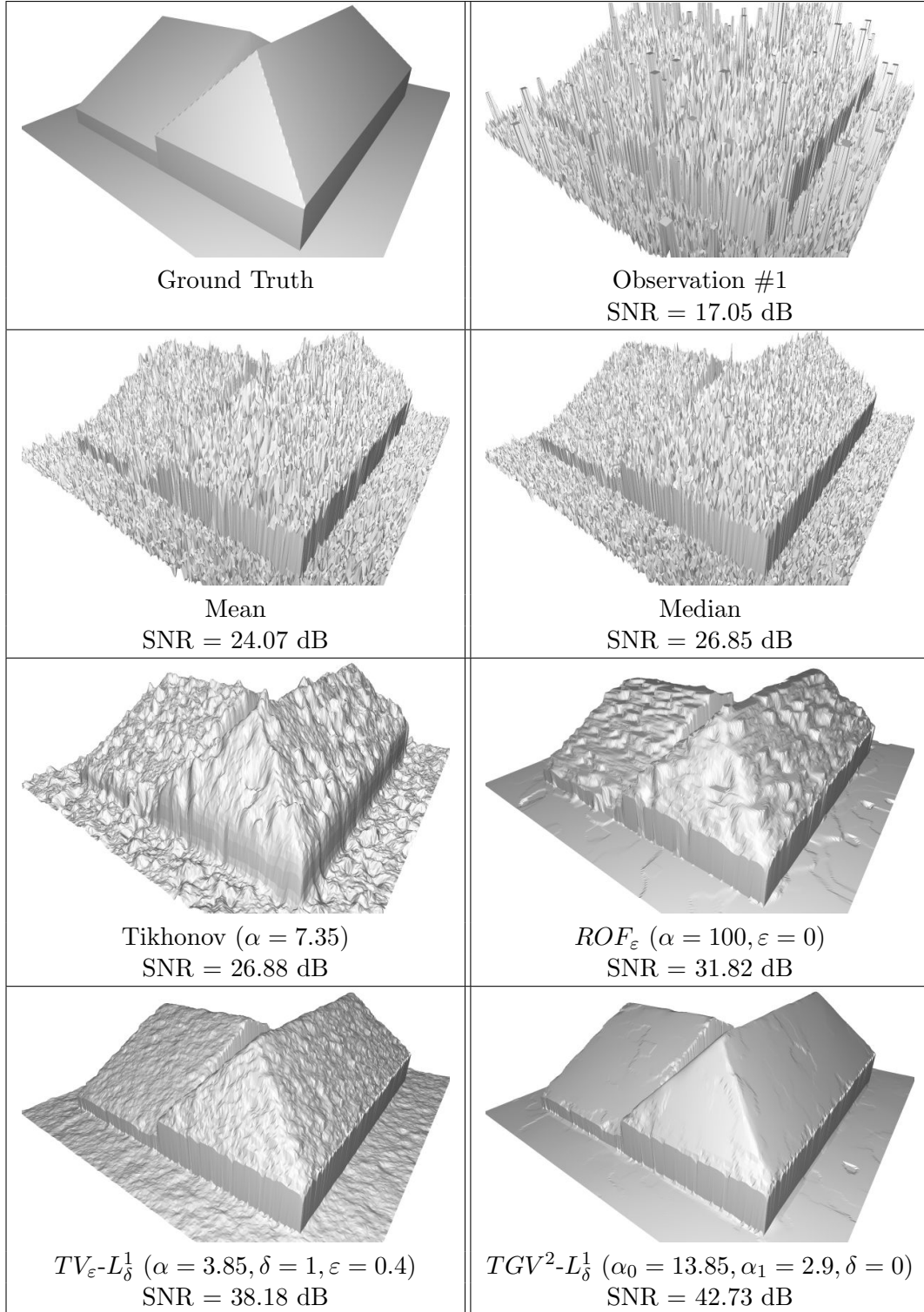


Table 2.2: Experiment with 10% outliers and 5 observations. For each algorithm the best result with respect to the  $L^2$ -distance to the ground truth is depicted. One notices that a robust data term is crucial for a good result. The  $TGV^2-L_\delta^1$  delivers better results than the  $TV_\epsilon-L_\delta^1$  algorithm. The quality of the results is visible in the 3D renderings.

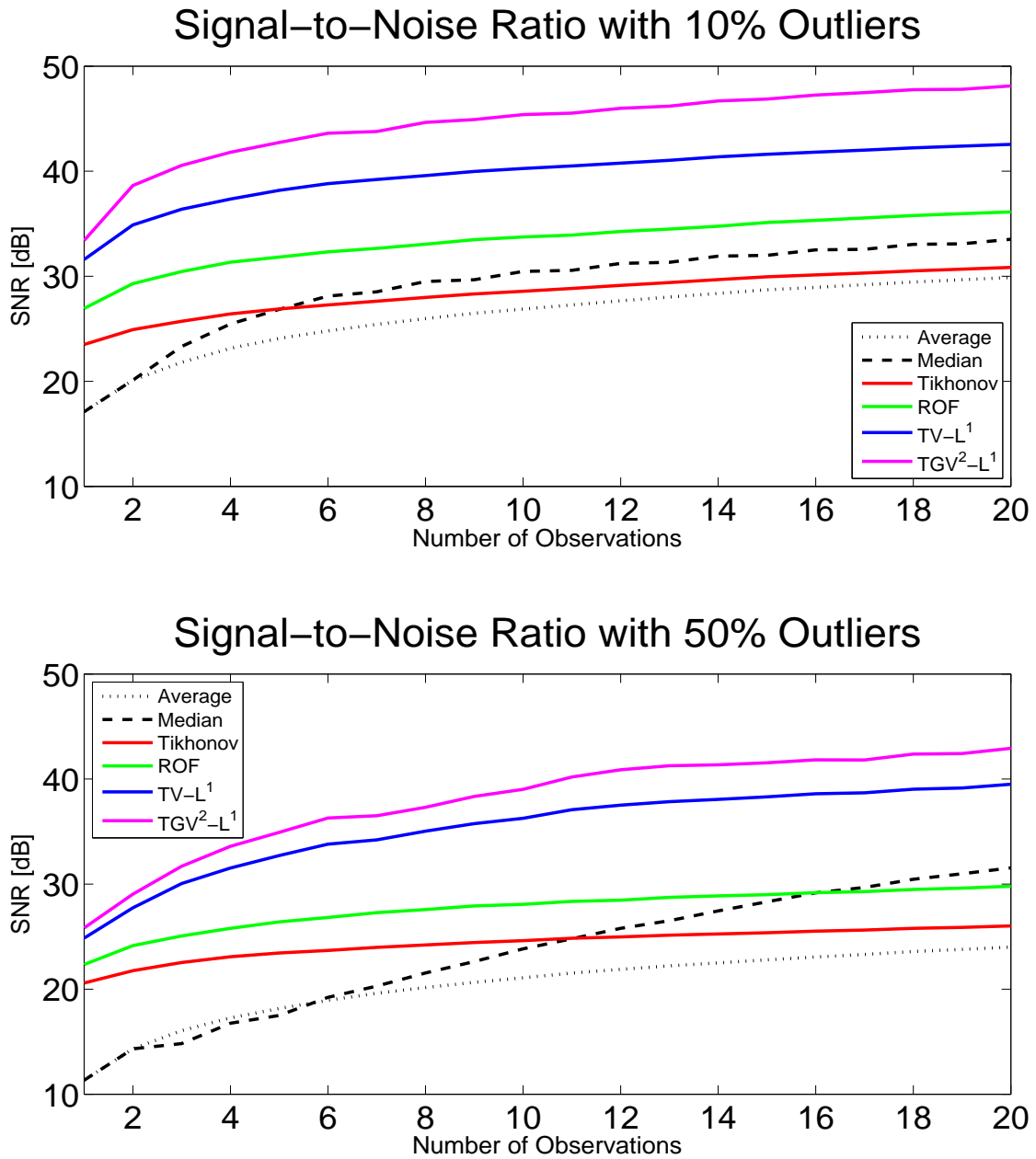


Figure 2.3: Signal-to-Noise ratios depending on number of observations. These plots graph the function  $10 \log \frac{|g|^2}{|u-g|^2}$ , where  $u$  is the minimizer of the energy functional and  $g$  is the ground truth signal.

is  $[50, 200]$ , in the experiments this signal is sampled on a regular grid with a size of 256 by 256 points. In order to evaluate the power of the regularization of the different energy

models, a number of differently distorted versions of the ground truth is generated (see Table 2.1).

For those distorted samples a Gaussian noise with a standard deviation of 10 is added. Additionally a certain amount of pixel groups with variable size are replaced with outliers which have a systematic offset of either plus or minus 50. Those distorted images are subjected to the energy minimization schemes outlined in the previous section. This experiment is performed with a different number of observations, starting with one distorted version of the ground truth. At most twenty different observations are used as this number of observations for a ground point is rarely surpassed in aerial flight missions even with a high overlap. Additionally one notices a diminishing return in the reduction of the remaining noise beyond that many samples (see Figure 2.3).

For each energy model a wide range of parameters is evaluated and thus the best settings are found via an exhaustive search. The exhaustive search makes it possible to find the optimal parameters for a given energy model and evaluation criterion. The evaluation criterion used in this survey calculates the  $L^2$  norm of the difference between each result and the ground truth. The reason for using this evaluation criterion is that the regularization via an energy functional not only has the purpose to eliminate noise, but most importantly outliers. Using the  $L^2$  norm for the evaluation prefers solutions where more outliers are removed, as they would be heavily penalized. The noise suppression ability is given as the ratio of the strength of the remaining noise to the signal strength of the ground truth,

$$10 \log \frac{|g|^2}{|u - g|^2}, \quad (2.80)$$

where  $u$  is the minimizer of the energy functional and  $g$  is the ground truth signal.

The results for those parameter settings which yield the lowest distances are depicted in Table 2.2 for 10% outliers. Please note, however, that the values for the parameters are not comparable as they have a different meaning in each energy and make a trade-off between different kinds of norms.

As a comparison two solutions are included which have no spatial regularization. In one case only the mean of all observations is calculated, in the other one the median. The lack of spatial regularization makes it easy to compute both solutions, but the results are unsatisfactory. The average performs worst of all models for small numbers of observations and then approximates the Tikhonov model as spatial regularization becomes less critical. The median also has a bad performance for few observations, but then quickly surpasses



the Tikhonov model. Afterwards it seems to approximate the performance of the  $ROF_\varepsilon$  model. For higher outlier rates, however, it fares much better than it, because it not features a robust data term guarding against outliers.

The Tikhonov energy model has problems at removing the noise while preserving the discontinuities. If the smoothness term would get an even higher weight, more noise would be removed, but the discontinuity would also be heavily blurred. The numerical results are the worst of all spatially regularized energy model and qualitatively it is also the most displeasing, independent of the amount of outliers used.

The  $ROF_\varepsilon$  energy model makes a very good tradeoff between removing the noise while preserving the discontinuities. However, because of the  $L^2$  data term the systematic errors give the reconstruction a bias and therefore larger differences to the ground truth. One can also still note traces of the characteristic stair-casing effect of the  $L^1$  norm along the gradients, despite the usage of the Huber norm. Higher settings of  $\varepsilon$  reduce this effect even further, but also have a negative impact upon the reconstruction of discontinuities.

The energy models with the quadratic data term collectively fail to handle outliers as expected and predicted by theory. The improved smoothness term of the  $ROF_\varepsilon$  model does help to improve the result, but quantitatively the result is far worse than those obtained with a robust data term.

The  $TV_\varepsilon-L_\delta^1$  energy model already performs much better, as the  $L^1$  norm lends robustness to the data term. Like the  $ROF_\varepsilon$  energy model it also suffers from the stair-casing effect, but it succeeds better at suppressing the outliers and therefore the obtained differences to the ground truth are much smaller.

The best results are achieved by the  $TGV^2-L_\delta^1$  energy model. This is no surprise as the synthetic example consists only of piecewise linear structure which do not incur any penalties for this model. This superiority is reflected in the graph plotting the signal-to-noise ratios for multiple observations. The advantage of the  $TGV^2-L_\delta^1$  energy model even grows at the beginning by adding more observations, then it stays fairly constant, even though the difference becomes smaller with more outliers.

In Table 2.3 the noise reduction ability of each energy model is graphed depending on the number of available observations. A difference of about 3.16 dB is therefore equal to halving the strength of the remaining noise.

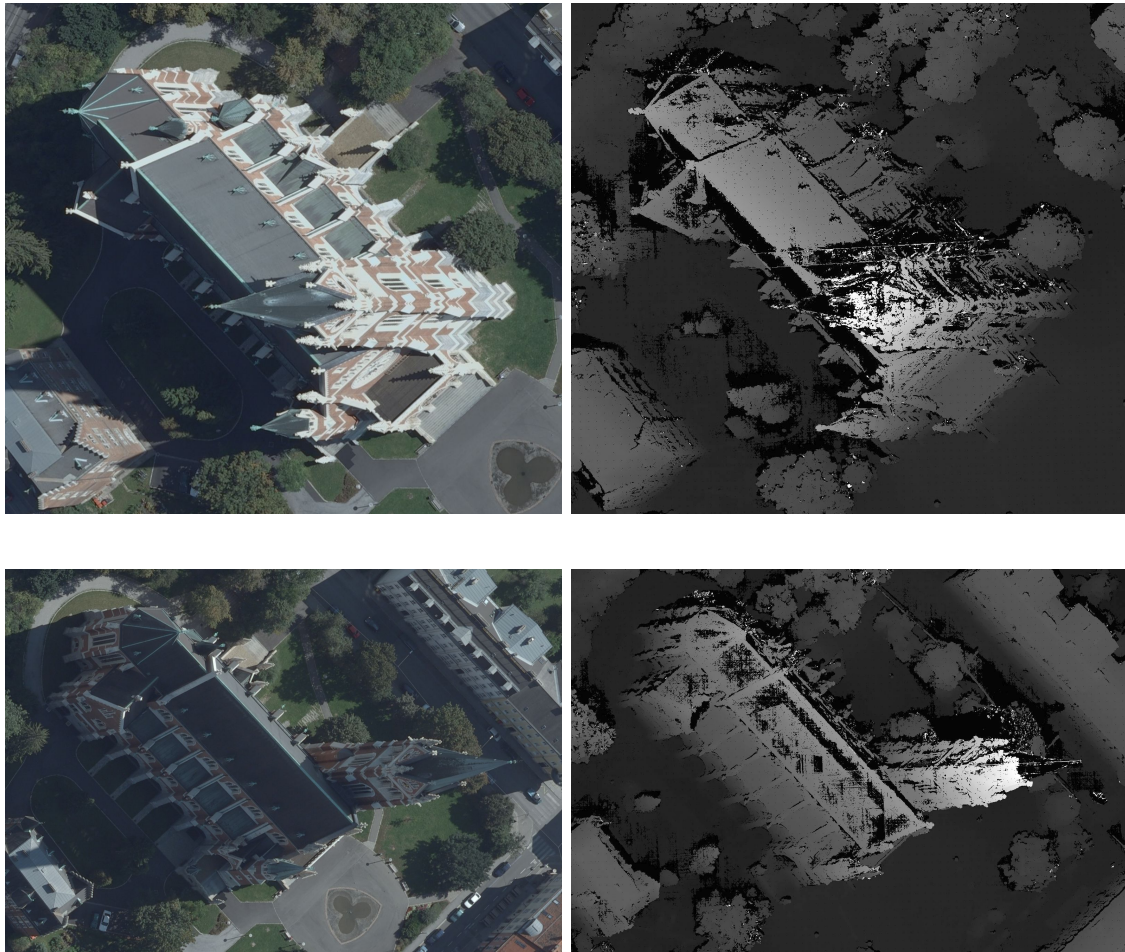


Figure 2.4: Input Data: On the left side the key image is shown, which is combined with two other input images to produce a pixel-synchronous range image (right hand side). Those range images contain outliers and regions where no reliable depth value could be estimated (black pixels in the range image).

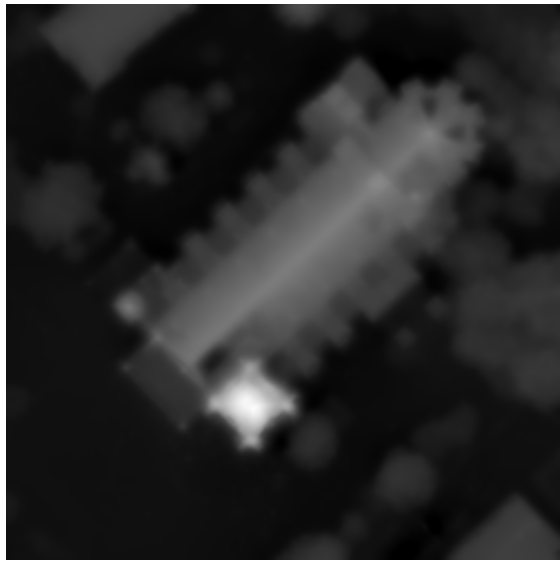
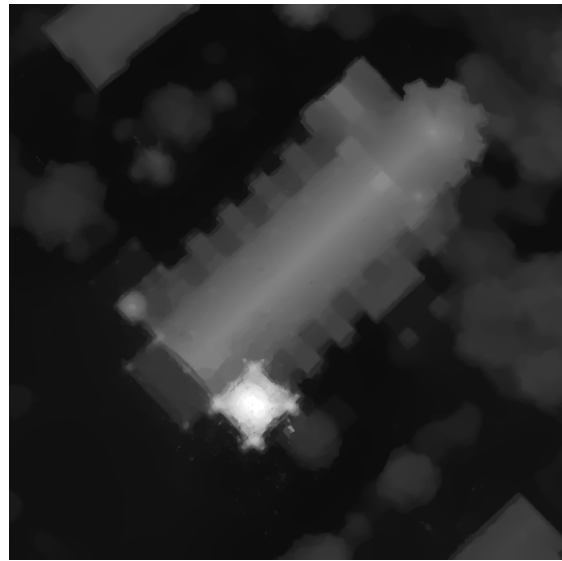
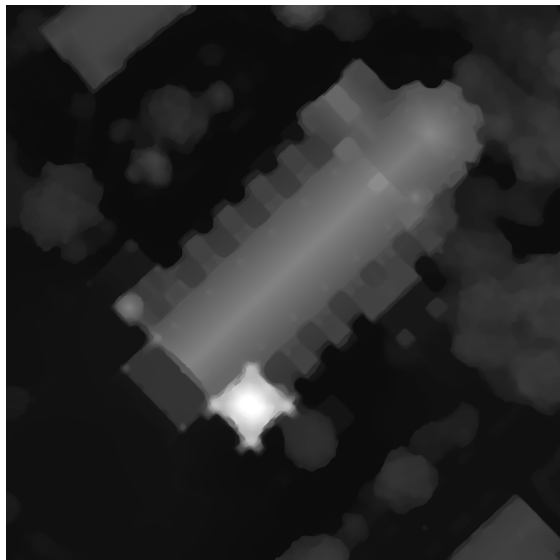
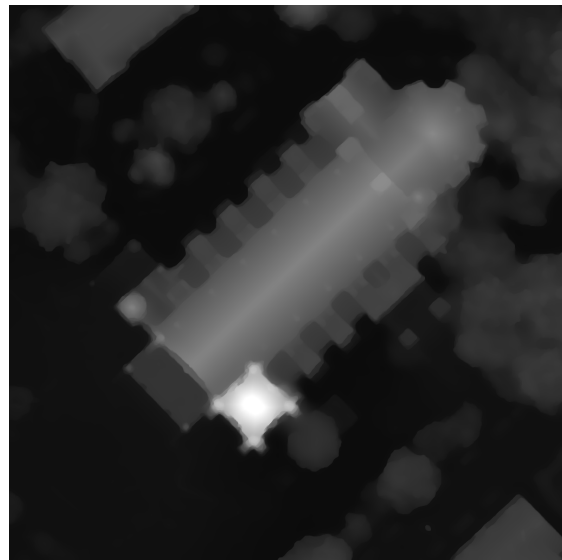
(a) Tikhonov,  $\alpha = 500$ (b)  $ROF_\epsilon$ ,  $\alpha = 750$ ,  $\epsilon = 1.0$ (c)  $TV_\epsilon-L_\delta^1$ ,  $\alpha = 15$ ,  $\epsilon = 0.5$ ,  $\delta = 0.1$ (d)  $TGV^2-L_\delta^1$ ,  $\alpha_1 = 125$ ,  $\alpha_0 = 20$ ,  $\delta = 0.1$ 

Figure 2.5: A robust dataterm is indispensable for real-world datasets: The Tikhonov and  $ROF_\epsilon$  energy models yield very bad results, because they cannot handle outliers. The  $TV_\epsilon-L_\delta^1$  and  $TGV^2-L_\delta^1$  show much better and similar results. Their difference becomes visible in a 3D rendering and plotting a cross-section of the building.

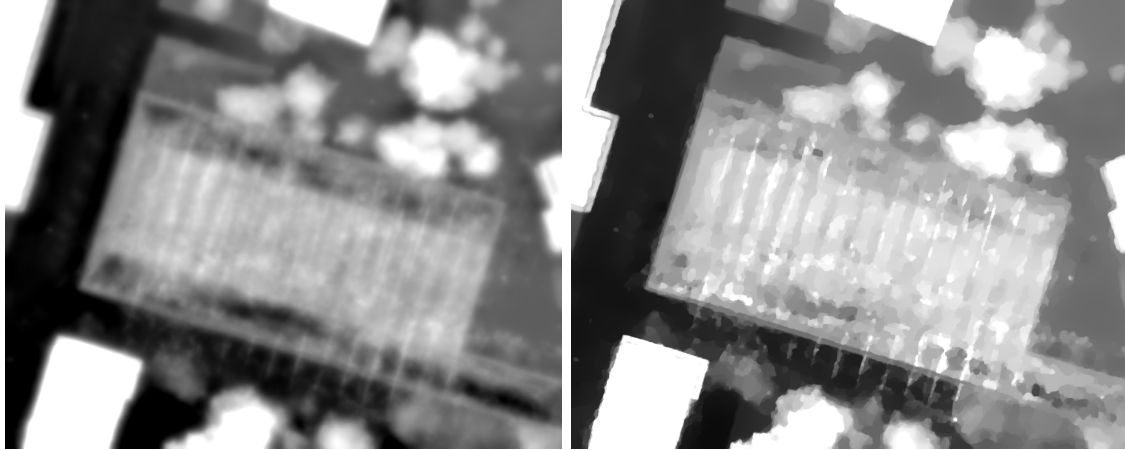
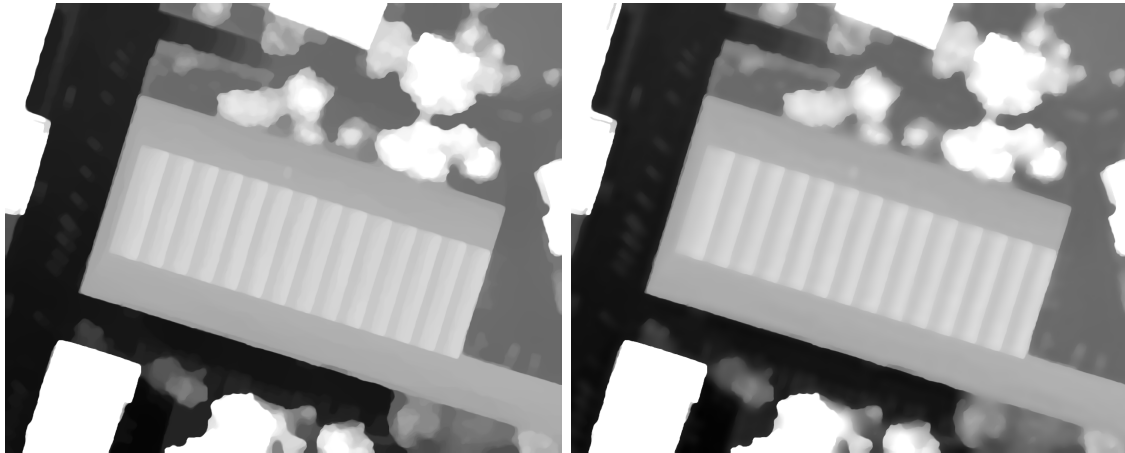
(a) Tikhonov,  $\alpha = 500$ (b)  $ROF_\epsilon$ ,  $\alpha = 750$ ,  $\epsilon = 1.0$ (c)  $TV_\epsilon-L_\delta^1$ ,  $\alpha = 15$ ,  $\epsilon = 0.5$ ,  $\delta = 0.1$ (d)  $TGV^2-L_\delta^1$ ,  $\alpha_1 = 125$ ,  $\alpha_0 = 20$ ,  $\delta = 0.1$ 

Figure 2.6: A warehouse with a very regular roof structure can be used to examine the behaviour of the energy models on sloped surfaces. Again, the results without a robust data term are unsatisfactory. The difference between first and second-order regularization become visible by plotting a cross-section of the roof.

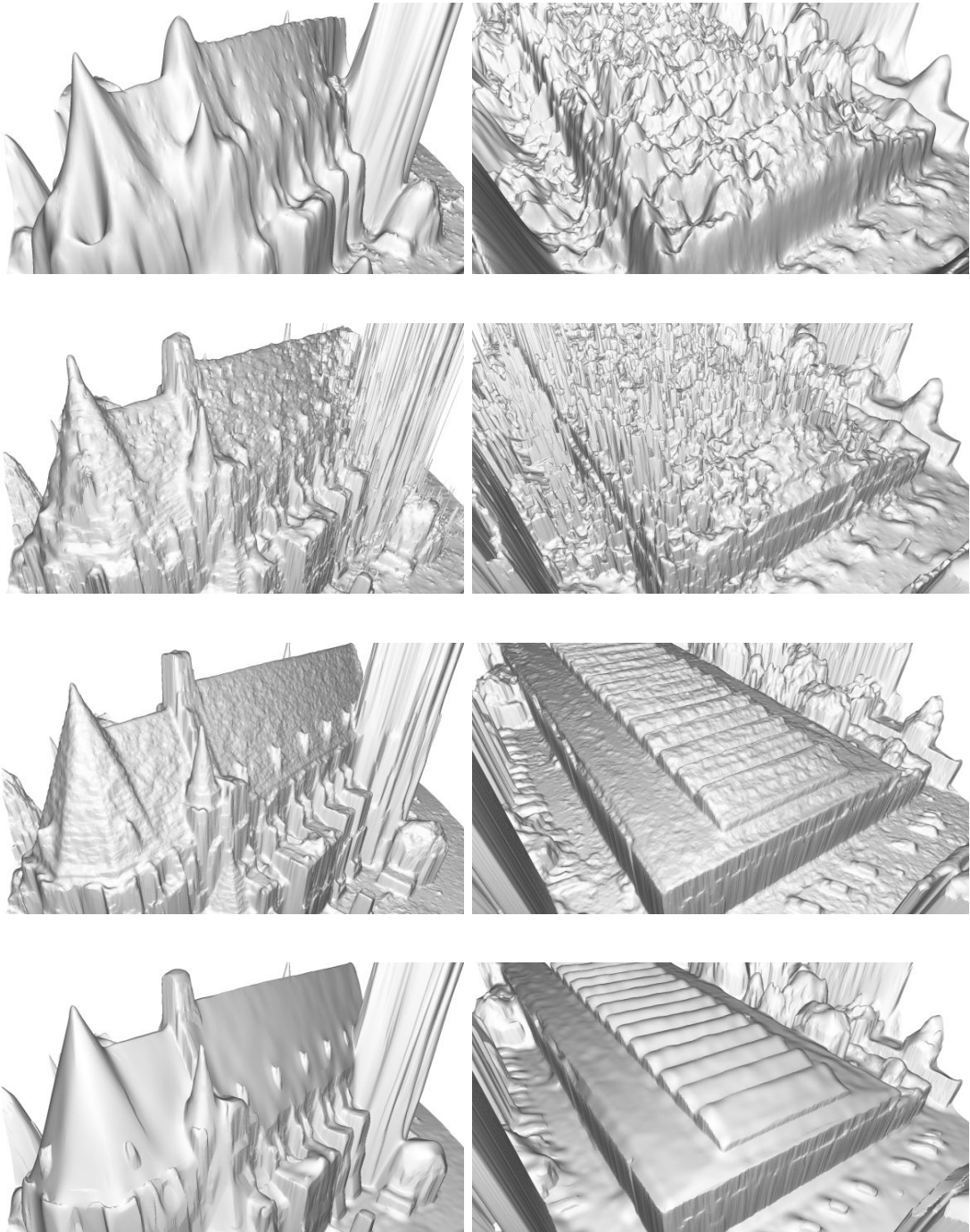


Figure 2.7: 3D renderings of the surfaces reconstructed with the different energy models make it easier to compare the quality as there is no ground truth. From top to bottom: Tikhonov,  $ROF_\varepsilon$ ,  $TV_\varepsilon-L_\delta^1$  and  $TGV^2-L_\delta^1$

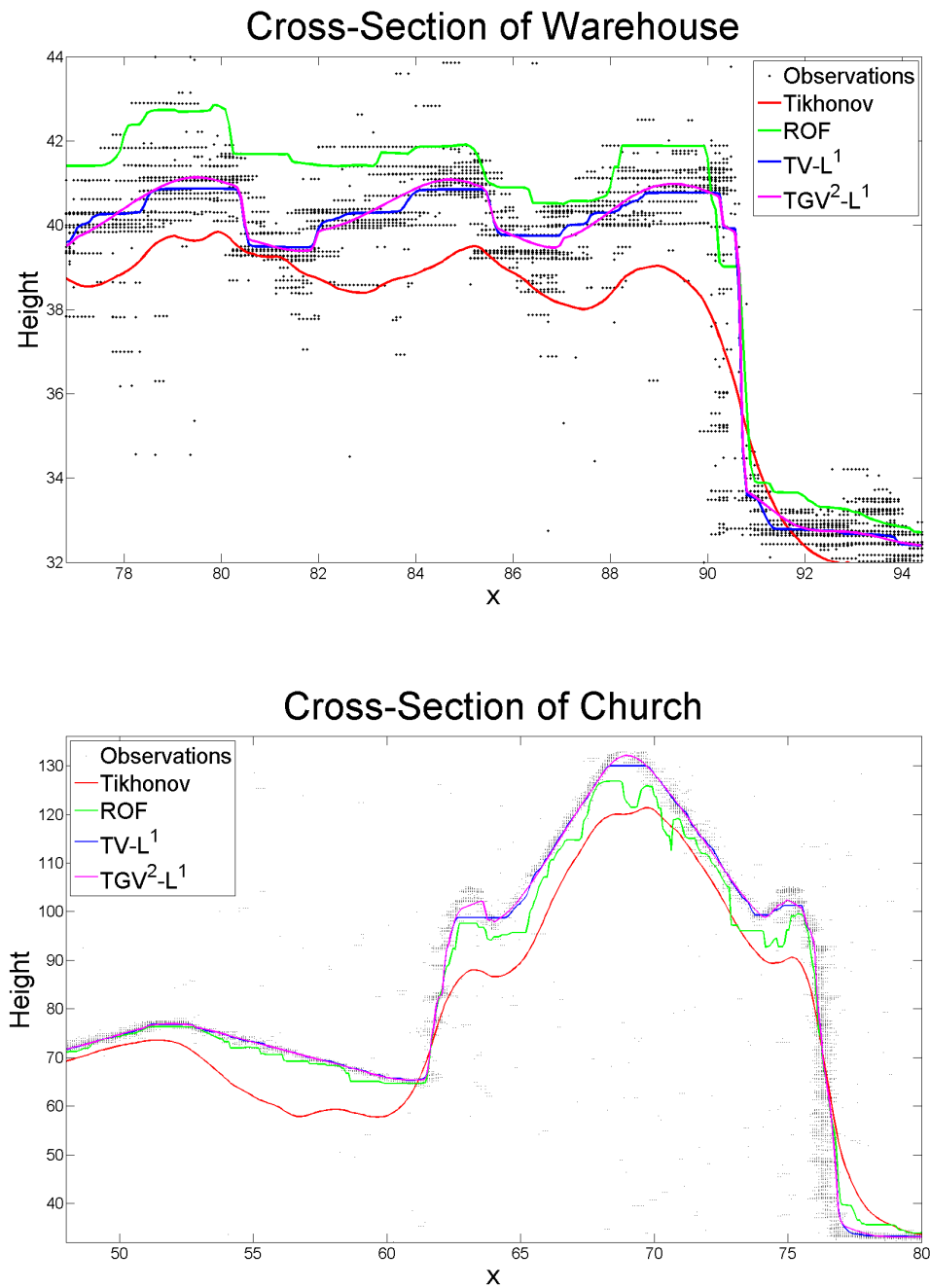


Figure 2.8: A cross-section trough the roof of the warehouse comes close to an idealized saw-tooth function. One notices the staircasing artifacts of the  $TV-L^1$  energy model which approximate the slopes. In comparison to that, the second order regularization of the  $TGV^2-L^1_\delta$  model yields a much better approximation to this sawtooth-like structure. The cross-section of the church highlights how the  $TGV^2-L^1_\delta$  energy model recovers the apex and the small turrets.

### 2.5.2 Aerial Dataset

The real-world datasets were produced by applying a dense image matching algorithm based on cascaded scanline optimization and backmatching similar to [27] to an array of aerial imagery datasets. For each image a pixel-synchronous range image is derived using two adjacent images (see Figure 2.4). Those images have a high degree of along-strip and across-strip overlap - the Graz dataset for example was flown with 85% along-strip and 65% across-strip overlap for example at a ground sampling distance of 8cm which amounts to about twenty observations per ground pixel on average). The perspective range images are then converted into a point cloud, which is projected onto the ground plane, thus giving multiple observations per pixel on the ground. Given that redundancy, a good reconstruction of the surface is possible even with noise and outliers in the range images (see Figure 2.5).

The Tikhonov energy model has very poor noise suppression qualities and also suffers at height discontinuities. One notices in the top row of Figure 2.7 the errors caused by outliers as small spikes in the surface, even though the surface is smoothed considerably as indicated by the structures close to the façade and on the roof which begin to disappear. The  $ROF_\epsilon$  energy model has sharper height discontinuities, but the outliers again cause bumps in the surface model as visible in the second row of Figure 2.7. Clearly the best results are obtained by the  $TV_\epsilon-L_\delta^1$  and  $TGV^2-L_\delta^1$  energy models which give sharp discontinuities and suppress all outliers while fine details are preserved. The difference between those two reconstruction is very small and difficult to assess looking only at the heightfields. A comparison of the two lower rows of Figure 2.7 shows that the surface produced by the  $TGV^2-L_\delta^1$  algorithm is much smoother.

A roof with a regular and known structure like the warehouse depicted in Figure 2.6 is another good example to compare the energy models, The difference between the  $TV_\epsilon-L_\delta^1$  and  $TGV^2-L_\delta^1$  energy models becomes visible by plotting a cross-section through the building (see Figure 2.8). One notices the staircasing artefacts of the  $TV_\epsilon-L_\delta^1$  energy model which approximate the slopes. In comparison to that, the second-order regularization of the  $TGV^2-L_\delta^1$  model yields a much better approximation to the sawtooth-like structure of the roof. In the case of the church the apex and the small turrets on the side of the tower are flattened by the first-order regularization, whereas they are preserved by the second-order regularization.

Larger datasets spanning whole cities cannot be solved in a globally optimal manner, because it is impossible to calculate the whole area at once (or at least prohibitively slow

to cache the data on the hard drive and continuously up- and download the measurements to the graphics card). Therefore the area of interest has to be divided into tiles. Each tile is processed independently, but with a sufficient overlap in order to ensure that the tiles match. The results of three different cities are depicted in the Figures 2.9, 2.10 and 2.11. Those datasets were processed on workstation with four Tesla cards, which were used in parallel.

Manhattan, New York, (Figure 2.9) features a very dense core of high skyscrapers. With normal a normal overlap along and across flight lines, streets between two such high buildings would be often impossible to be reconstructed as there are no points on the street that would be visible in at least two images. Increasing the overlap to 90% and 80% respectively, however, makes it possible to overcome this problem.

Truly three dimensional objects pose a problem in the  $2\frac{1}{2}$ D fusion approach. A crane high above a roof, for example, usually results in many points lying on the same plane above the roof. During the fusion only one level, either the roof or the crane, can be included in the surface model. Under such circumstances often the size of the crane and thus the number of points generated by the dense stereo matching is deciding. An example of this can be seen in the Oakland dataset (Figure 2.10), where one portion of the crane is included in the DSM, at another location the roof surface is picked by the algorithm. Even careful parameter tuning is unlikely to get rid of such artifacts.

Water bodies are usually very difficult to match because it rapidly changes its appearance. This results in inconsistent and often wrong measurements obtained by the dense stereo matcher. The urban core of Pittsburgh (Figure 2.11) is located at a bifurcation of a river with many bridges. The problems arising when dealing with water regions can be observed by the difference in quality of the reconstruction of a bridge (where even the pattern of the narrow cross beams is faithfully recovered) compared to the surface of the water, which is inconsistent and not homogeneous.

### 2.5.3 Middlebury Dataset

The Middlebury Dataset is well known in the dense image matching community as it provides four rectified image pairs with known ground truth to evaluate and compare dense image matching algorithms. At the time of this writing, 76 algorithms and their result are listed on the Middlebury homepage. For the experiment the top 10 results of the list for an error threshold of 1.0 pixels were downloaded. Then each of the described energy models is used to combine them into a new result. An overview of those results



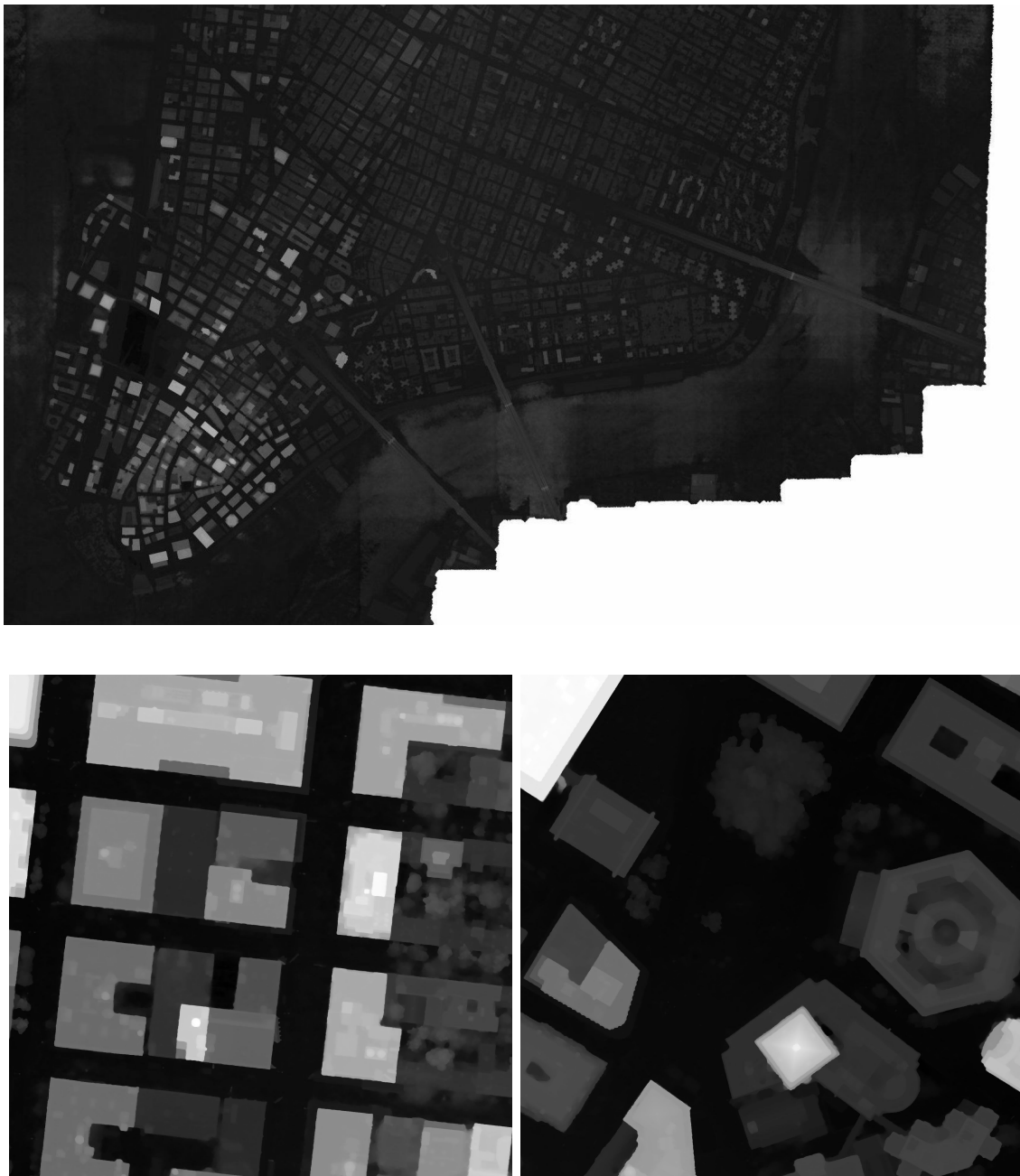


Figure 2.9: Manhattan, New York: The upper image shows lower Manhattan which was covered by approximately 250 images, which were acquired with a very high forward and sideward overlap (90% / 80% at 15cm GSD), because the downtown area features very high skyscrapers which make it difficult to get enough observations on the street level there with lower overlaps. The range image fusion process took about 15 minutes on a workstation with four Tesla cards. The two images at the bottom show details from the fused heightfield.

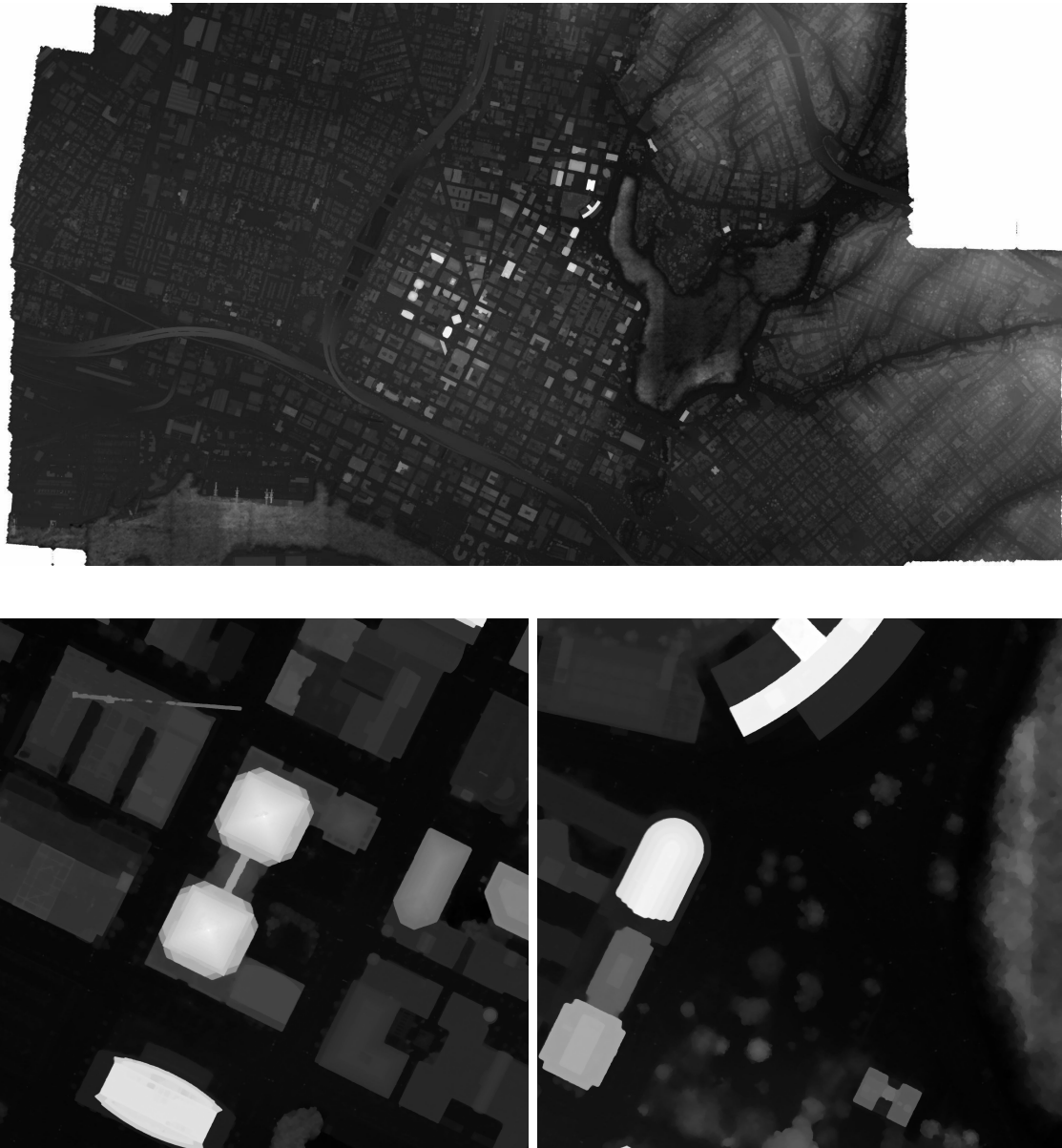


Figure 2.10: Oakland, California: This dataset consists of just 122 images and the range image fusion process takes 8 minutes. The ground sampling distance is 15cm and the overlaps are about 80% and 60%. Truly three dimensional objects pose a problem because they feature multiple valid height values for one point in the DSM. The left detailed view at the bottom shows that the crane is not fully reconstructed, because it has to contest with the roof surface. Often the size and thus the number of points is crucial whether an object gets included in the surface model. The right view shows that the contour of buildings is very sharp independent from the shape.

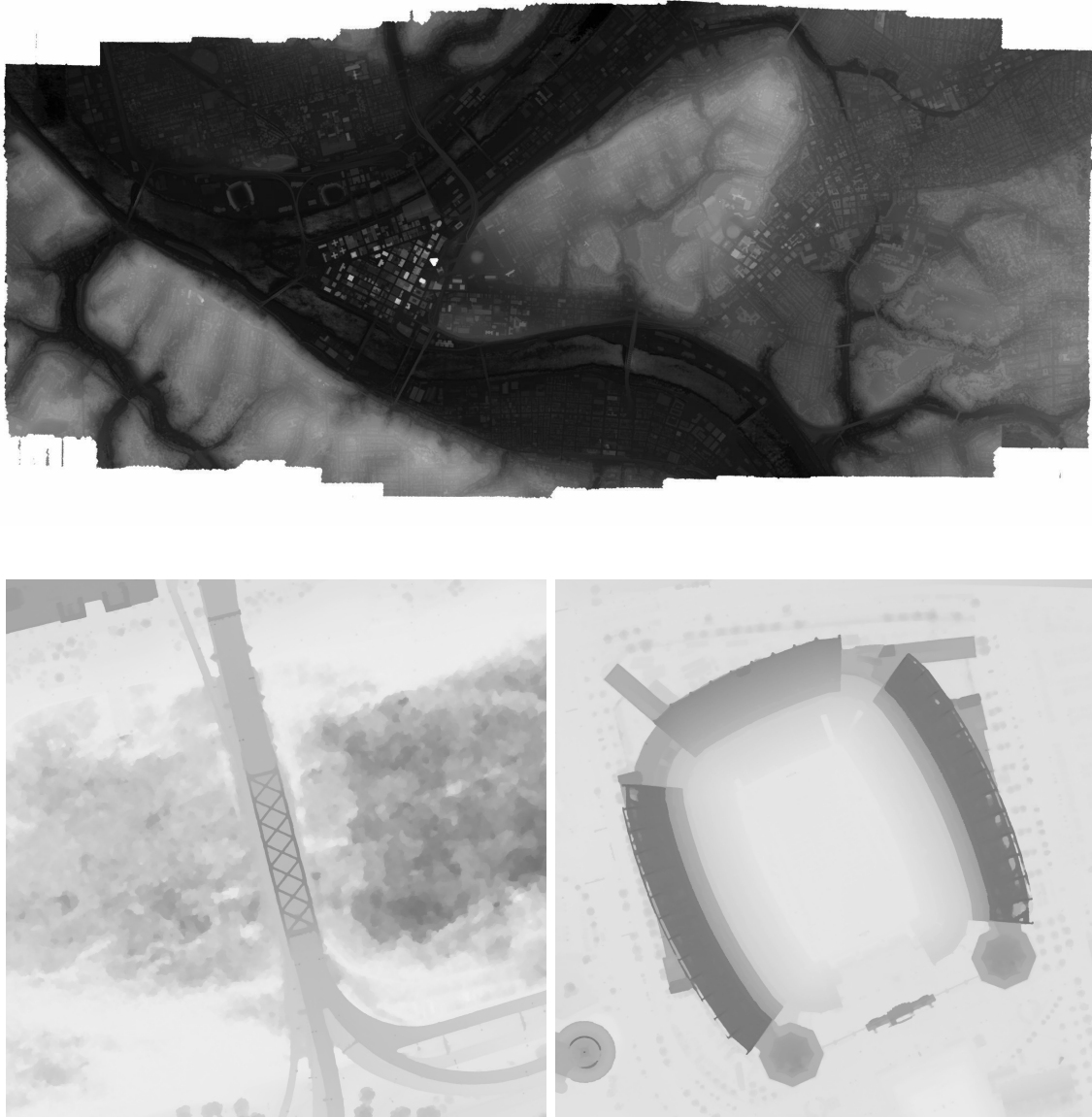


Figure 2.11: Pittsburgh, Pennsylvania: The upper image shows the result obtained for the city of Pittsburgh. The dataset consists of 358 images, the processing time is about 15 minutes. One can see that the hills were nicely recovered. Water is difficult to match, because it constantly moves and has changing reflections, therefore the height samples are not consistent there. This can be seen in the left image at the bottom where the surface is distorted around the bridge. The detailed views show the obtained accuracy for a stadium and a bridge, where even the regular pattern of the cross beams above the street is visible.

is listed in Table 2.3. Energy models with a quadratic data term (Tikhonov and  $ROF_\varepsilon$  energy model) do not improve the result. Compared to the mean disparity value, the additional regularization squeezes no additional improvements out of the top 10 disparity maps. The quadratic smoothness term of the Tikhonov energy does not even improve the result as the best result is obtained with a smoothness weight of zero, which reduces this model to a purely pointwise averaging.

The  $TV_\varepsilon-L_\delta^1$  and  $TGV^2-L_\delta^1$  energy models on the other hand succeed at combining the range images and produce a disparity map which clearly outperforms the results of the top 10 dense image matching algorithms. The difference, however, between the two energy models is very small. This hints at the importance of a robust data term in order to be resilient to outliers, which can not be avoided in most computer vision settings.

It is surprising that the simple median of the top 10 results for the one pixel error threshold gives only slightly worse results, which conveys the high quality of the available disparity maps: for the most part they do not require much spatial regularization. In Table 2.4 the results of the  $TGV^2-L_\delta^1$  are depicted in detail.

It is also noteworthy that using only the result from a single dense image algorithm listed on the Middlebury ranking, the  $TV_\varepsilon-L_\delta^1$  and  $TGV^2-L_\delta^1$  regularization is able to improve it with fitting parameters. This indicates that some of those algorithms could perform even better in the ranking, if they would perform a regularizing post-processing on their disparity maps.

Evaluation with an error threshold of 1.0 pixels														
Energy Model	Parameters	Tsukuba	Venus	Teddy	Cones	% Bad								
$TGV^2-L_\delta^1$	$\alpha_0=5.2, \alpha_1=2.05, \delta=2$	0.81	0.15	0.98	2.23	6.30	6.93	2.11	6.29	6.17	<b>3.12</b>			
$TV_\varepsilon-L_\delta^1$	$\alpha=1.0, \varepsilon=3.5, \delta=2.7$	0.82	1.06	4.43	0.07	0.15	0.97	2.29	6.31	7.18	2.06	6.37	6.04	<b>3.14</b>
Median		0.67	0.88	3.64	0.06	0.14	0.81	2.48	6.78	7.39	2.15	6.62	6.26	<b>3.16</b>
DoubleBP		0.88	1.29	4.76	0.13	0.45	1.87	3.53	8.30	9.63	2.90	8.78	7.79	<b>4.19</b>
AdaptingBP		1.11	1.37	5.79	0.10	0.21	1.44	4.22	7.06	11.8	2.48	7.92	7.32	<b>4.23</b>
SubPixDoubleBP		1.24	1.76	5.98	0.12	0.46	1.74	3.45	8.38	10.0	2.93	8.73	7.91	<b>4.39</b>
CoopRegion		0.87	1.16	4.61	0.11	0.21	1.54	5.16	8.31	13.0	2.79	7.18	8.01	<b>4.41</b>
GC+SegmBorder		1.47	1.82	7.86	0.19	0.31	2.44	4.25	5.55	10.9	4.99	5.78	8.66	<b>4.52</b>
OutlierConf		0.88	1.43	4.74	0.18	0.26	2.40	5.01	9.12	12.8	2.78	8.57	6.99	<b>4.60</b>
WarpMat		1.16	1.35	6.04	0.18	0.24	2.44	5.02	9.30	13.0	3.49	8.47	9.01	<b>4.98</b>
$ROF_\varepsilon$	$\alpha=0.6, \varepsilon=2.5$	1.46	1.97	7.82	0.20	0.39	2.82	4.14	8.69	11.6	3.10	8.64	9.02	<b>5.00</b>
Mean		1.46	1.97	7.82	0.20	0.40	2.82	4.14	8.68	11.6	3.12	8.68	9.07	<b>5.00</b>
Tikhonov		1.46	1.97	7.82	0.20	0.40	2.82	4.14	8.68	11.6	3.12	8.68	9.07	<b>5.00</b>
Undr+OvrSeg		1.89	2.22	7.22	0.11	0.22	1.34	6.51	9.98	16.4	2.92	8.00	7.90	<b>5.39</b>
AdaptOvrSegBP		1.69	2.04	5.64	0.14	0.20	1.47	7.04	11.1	16.4	3.60	8.96	8.84	<b>5.59</b>
GeoSup		1.45	1.83	7.71	0.14	0.26	1.90	6.88	13.2	16.1	2.94	8.89	8.32	<b>5.80</b>

Table 2.3: The Middlebury stereo vision suite consists of four reference datasets which are used by researchers around the globe to evaluate and compare their dense image matching algorithms. This table sorts the algorithms according to their percentage of bad pixels. On the official site a different sorting is used based on the average rank, which is unsuitable for this evaluation, as multiple additional results are evaluated. For each energy model the optimal parameter settings are used. In addition to the energy models studied in this thesis the median and average values of the top 10 disparities maps are also evaluated. The evaluated entries are highlighted in the table.






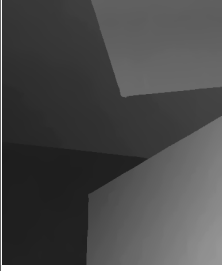

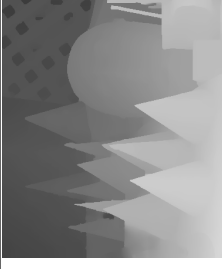
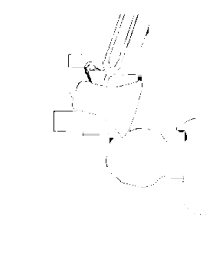
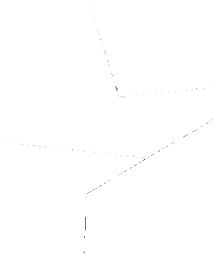

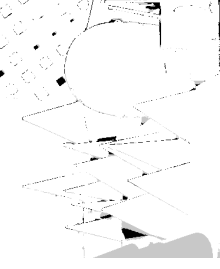
	Tsukuba	Venus	Teddy	Cones
Input Image				
Disparity Map				
Error Image				

Table 2.4: This table shows the results obtained by applying the  $TGV^2-L_0^1$  algorithm to the results of the top 10 algorithms in the Middlebury ranking with an error threshold of 0.5 pixel. The first row shows one image of the stereo pairs of the four data sets. The second row shows the estimate of the disparity map by the  $TGV^2-L_0^1$  energy model. In the third row an error image is depicted which shows where a wrong disparity value was estimated. See Table 2.3 for numerical details of these results.

## 2.6 Conclusions

In this section convex energy models for image regularization are examined. Then a reasoning is presented how to extend those energy models to multiple observations. A flexible and efficient optimization scheme is introduced and adapted to each energy model.

Using a synthetic dataset and the Middlebury stereo evaluation suite (both have a known ground truth) all energy models are quantitatively evaluated. Those experiments uniformly show that a robust data term ( $L^1$  or Huber norm) is critical for obtaining good results in the presence of outliers. The evaluation with a synthetic dataset clearly shows that the  $TGV^2-L^1_\delta$  energy model is the best choice for this problem. The results obtained from the Middlebury evaluation suite confirm this result, even though the difference between the energy models with a robust data term is very small.

Additionally the algorithms are applied to a real world dataset derived from aerial imagery. From the experiments in this section one can conclude that the  $TGV^2-L^1$  energy model has the best regularization properties among the investigated algorithms and thus is the algorithm of choice for DSM generation.





## Chapter 3

# Ortho Image Generation

### Contents

---

<b>3.1</b>	<b>Introduction . . . . .</b>	<b>65</b>
<b>3.2</b>	<b>Warping of Images . . . . .</b>	<b>66</b>
<b>3.3</b>	<b>Compositing Images . . . . .</b>	<b>68</b>
<b>3.4</b>	<b>Experimental Results . . . . .</b>	<b>79</b>
<b>3.5</b>	<b>Conclusions . . . . .</b>	<b>85</b>

---

### 3.1 Introduction

Ortho images are a valuable derivative product obtained from aerial images. They are created by warping the perspective aerial images onto the terrain surface and then stitching and/or blending them together. Traditionally ortho images are generated by warping the perspective aerial imagery onto a low resolution digital terrain model and blending them together along some seamlines. This procedure, however, causes visible artifacts in the resulting ortho image because the seams do not always align (for example if the height is not correctly estimated at the seam location) and portions of the façade are still visible because the geometry used for warping is not accurate enough to compensate for those height discontinuities.

The surface model obtained from the robust integration of range images as presented in Chapter 2 provides the geometric accuracy for generating a "true" ortho image, where all vertical structures disappear.

The problem of combining the individual images into one large ortho image can be addressed in two ways: first, it can be viewed as a labeling problem as in the case of

the traditional ortho image. This requires to solve a multi-class labeling problem and then to blend the warped patches together. Alternatively, a continuous color-surface can be estimated from the warped patches as the digital surface model is accurate enough to make those warped patches pixel- synchronous. In this chapter both methods are presented and analysed. Their runtime and the quality of the results are evaluated and discussed.

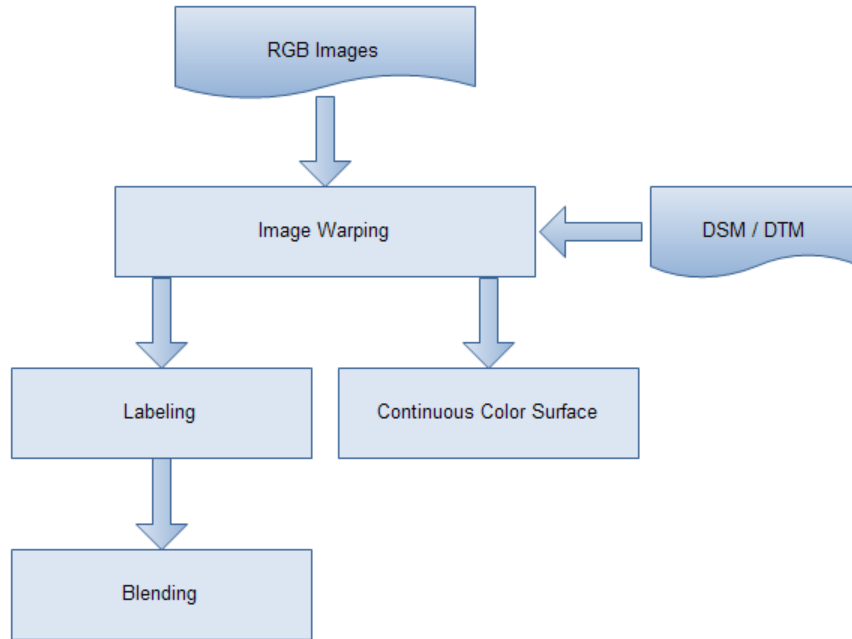


Figure 3.1: This Figure gives a schematic overview of the two possible ways how to derive an ortho image from aerial imagery. Both ways rely on a surface model to warp the images. The traditional approach is then to compute a labeling which minimizes the color differences between warped patches along some seams where a blending is applied. Alternatively a continuous color surface can be estimated which approximates all available samples at the same time.

## 3.2 Warping of Images

The input images available for composing the ortho image naturally feature a perspective distortion. Before they can be combined into a contiguous ortho image, this distortion has to be removed. This is achieved by projecting them on the same three dimensional surface. The accuracy of this three dimensional surface directly influences the quality of the resulting ortho image. A flat plane induced by a homography for example, could

already serve as the three dimensional surface. However, only points lying on that plane will match after warping. Traditionally, ortho images from aerial imagery are derived using a DTM. For rural areas this works very well as buildings (and other objects which are not included in the DTM geometry) are normally quite low and do not cause severe depth discontinuities unlike in urban environments. Especially in urban cores with high buildings the result of projecting the perspective imagery on the DTM is unsatisfactory. For those areas the automatically derived DSM provides the necessary details to allow an accurate warping.

Warping the image itself is done by sampling the three dimensional surface and projecting the points back into the perspective imagery. In Figure 3.2 perspective images, together with the corresponding warped patches for two different surface models, are shown. It becomes clear that without a high quality surface model, it is more challenging to stitch different images together. Modelling the height discontinuities is also the only way to generate a true ortho images where vertical structures disappear. Using only a DTM results in leaning buildings in the final ortho image.

Additionally the visibility of each sample point on three dimensional surface is checked. This information is vital later on, when the images are stitched together, as the borders should be placed in such a manner that no part of the ortho is textured from an image which does not see the respective pixels.

Depending on the sampling distance and the slope of the three dimensional surface with respect to the viewing angle, the above mentioned scheme can lead to aliasing artifacts. This effect can be mitigated by either supersampling the affected parts or filtering the input image to remove high frequencies. In practice one builds a image pyramid and access a level corresponding to the sampling density in the input image.

As mentioned before, the warping step already yields important information for the stitching, as each pixel is assigned a score which determines how well an image is suited for texturing a certain point on the surface. This score depends on the viewing angle, distance and other image quality parameters (avoiding over- or underexposed images for example), but practise has shown that for aerial images it is sufficient to take the viewing angle into account, as the distance is for all images approximately the same:

$$score(p, i) = \begin{cases} -\infty & \text{if not visible} \\ \mathbf{view}_i \cdot (\mathbf{p} - \mathbf{center}_i) & \text{if visible} \end{cases} \quad (3.1)$$

where  $p$  is the point on the surface,  $i$  is the index of the image to be considered. The

two camera parameters which influence the score are given by  $center_i$  and  $view_i$  which represent the camera center and viewing angle respectively. Texturing from an image which does not see a surface point incurs high costs and will therefore be avoided in the ortho image composition.

### 3.3 Compositing Images

After warping each input image, the resulting patches have to be merged together to form a contiguous ortho image. As mentioned above there exist two different approaches to this problem.

On the one hand it is possible to phrase the problem as a multi-class labeling problem to stitch the images together. Ideally the borders between individual patches should not be noticeable, thus a blending algorithm tries to diffuse the differences along the seamlines. This approach has the advantage that as much content as possible is extracted from a single image thus restricting the possibility of disturbing visual artifacts along the patch boundaries.

The other possibility is to estimate a continuous color-surface through all available observations. This exploits the redundancy inherent in the input images and allows to suppress moving objects for example. Additionally this approach is perfectly suited for parallelization on graphics hardware which drastically reduces the runtime of the algorithm.

#### 3.3.1 Composition via Labeling

Much research has already been dedicated to image stitching and composition by labeling and blending the input images together.

The placement of the borders of the images is formulated as a multi-class labeling problem. Each sampled point is visible in a set of images, but depending on the viewing angle and distance of the image, one is better suited for texturing that pixel than others, thus yielding a data term. Additionally one would like to avoid long and complicated border configurations as this makes the blending step later more difficult. Ideally those borders should be placed at locations where the images have similar intensity values. These constraints can be framed as a pairwise pixel interaction term and finally yields an energy function like

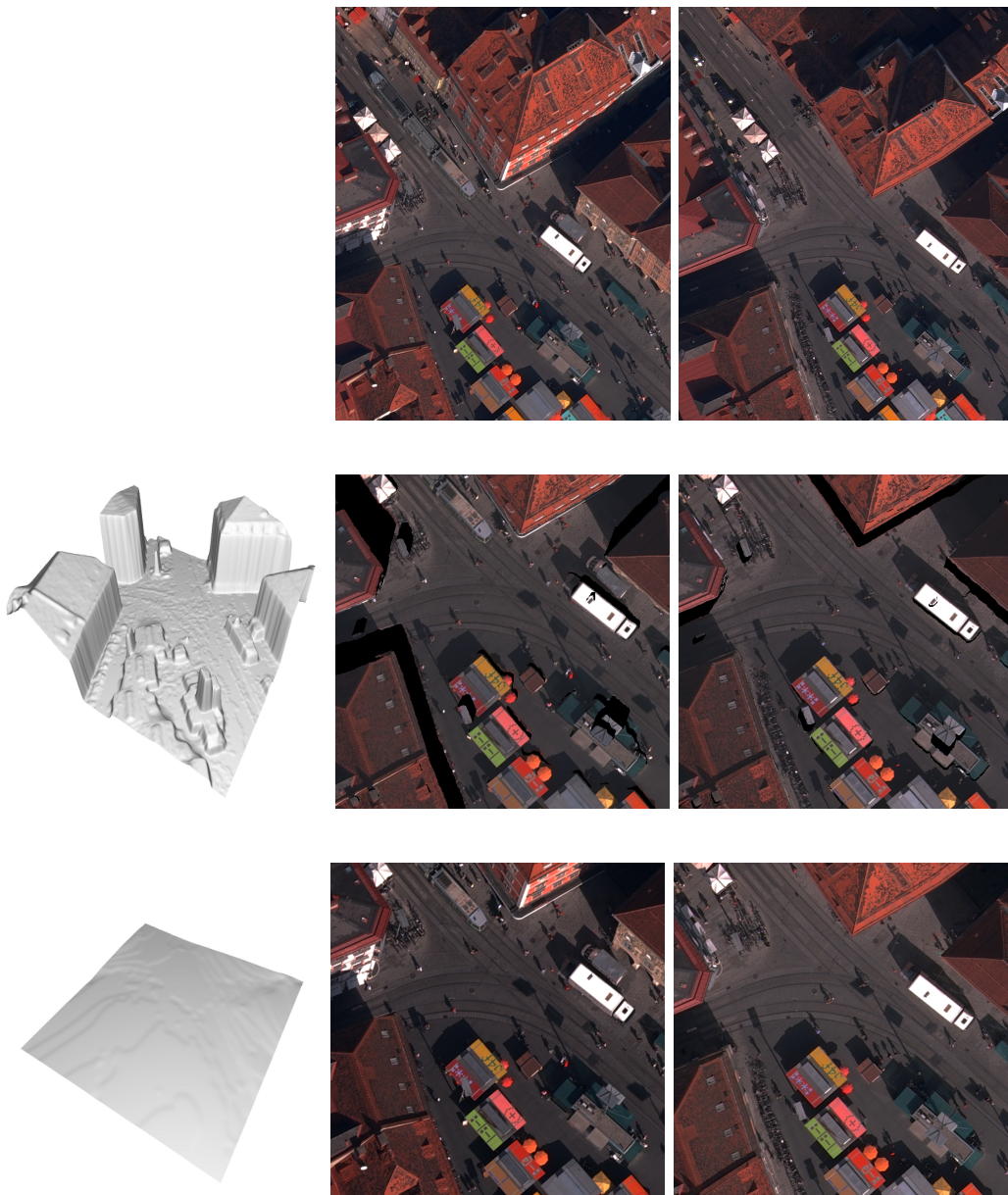


Figure 3.2: This Figures shows the result of warping aerial images onto different geometries: in the top row two different aerial images are shown, which are warped on a DSM (middle row) and a DTM (bottom row). The left column depicts the geometry which is used. The accuracy of the DSM results in two pixel-synchronous warped images - some areas are of course not visible (black). The DTM only gives correct correspondences on ground level; all structures above the ground do not match in the warped views. As a benefit, there are no occluded areas as the geometry has no height discontinuities.

$$E(L) = \sum_{p \in \mathcal{N}} D_p(L(p)) + \lambda \sum_{(p,q) \in \mathcal{E}} S_{p,q}(L(p), L(q)) \quad (3.2)$$

where the neighbourhood system of the pairwise interaction terms defines a graph by a set of nodes and edges  $(\mathcal{N}, \mathcal{E})$ . The functions  $D_p$  and  $S_{p,q}$  give correspondingly the costs connected to a certain label configuration indicated by the label function  $L(p)$  for pixel  $p$  and  $q$ . It is important to note that the neighbourhood systems encoded in  $\mathcal{E}$  impacts the solution as a grid bias is introduced, also called metrication error.

Even though there exist algorithms to solve this problem globally optimal [52], it is often not necessary and faster to approximate the solution via repeatedly solving binary labeling problems with the above mentioned structure. The multi-class result is then obtained via alpha expansion moves or alpha-beta swap moves ([10, 37]).

The project areas are routinely much too large to handle it all at once, therefore it has to be subdivided into smaller tiles. Each tile features an overlap to allow for consistent borders.

### 3.3.1.1 Graph Cuts

In general the energy which can be minimized by Graph Cuts in conjunction with alpha expansion moves look like Equation 3.2.

Additionally this algorithm relies on the pairwise penalty function  $S$  to be submodular. In the case of the alpha expansion move this means that for all labels and pixels the following equation has to hold:

$$S_{p,q}(\alpha, \alpha) + S_{p,q}(\beta, \gamma) \leq S_{p,q}(\alpha, \gamma) + S_{p,q}(\beta, \alpha) \quad (3.3)$$

Even though algorithms have been proposed which are able to deal with non-submodular terms, it has been shown ([35]) that those terms do not noticeably improve the results of image stitching.

For submodular terms there exists a construction technique [37], which allows to translate the energy minimization problems 3.2 into finding the maximum flow in a graph in polynomial time.

The Graph Cuts algorithm has two major drawbacks. First, depending on the neighbourhood system it has a metrication error associated with it. This is visible in the results which show a preference for horizontal and vertical edges in the case of the 4-connected graph. By extending the neighbourhood system to 8 or more nodes, this effect can be

mitigated but at the cost of increased runtime and memory consumption. The second drawback is the sequential nature of the algorithm which makes it difficult to parallelize to take advantage of multicore machines or modern graphics cards. The algorithm with the best performance so far iteratively finds the shortest path between source and sink and pushes flow along this path. Once all edges between source and sink are saturated the algorithm terminates.

### 3.3.1.2 CUDA Cuts

In recent years the computational power of CPU and GPU diverged. Nowadays the flagship products of leading graphic card producers excel the best CPUs by more than an order of magnitude in terms of operations per second and those indicators will most probably continue to drift apart. It is therefore self-evident that it is promising to port algorithms to the GPU in order to exploit and benefit from this development.

Taking into consideration that the computational power on the GPU is provided by a high degree of parallelism (currently the most powerful architecture of Nvidia features 30 multiprocessors each with 8 cores), it is obvious that filter operations or similar per-pixel processes are able to greatly benefit from the processing power given such a data parallel architecture. On the other hand, it is more difficult to speed up sequential algorithms like finding the minimum or sorting an array of numbers, but by exploiting mechanisms like the prefix sum algorithm [8], a decent speedup is possible.

Recently the Graph Cuts algorithm, which is difficult to implement in a parallelized fashion, was successfully ported to the GPU [71]. There exist various algorithms how to compute the max-flow/min-cut solution for a graph. The Edmonds-Karp algorithm for example repeatedly finds the shortest path between source and sink and adjusts the weights correspondingly. The push-relabel algorithm originally proposed by Anderson and Setubal [2] is better suited for the parallel architecture of the GPU.

The idea behind the algorithm is to exploit the parallelism of the GPU by letting each node in the graph push excessive flow to its neighbours. In order to direct those excessive flows from the source towards the sink, a potential is computed for each node, such that flow is only push from higher nodes to those below. At the beginning all nodes connected to the source have the highest potential, whereas those connected to the sink have the lowest potential.

There exist two ways how to estimate the potential of each node: a global breadth-first search starting from the sinks assigns each node the correct potential, but may take many

iterations depending on the graph. Therefore this global search is used only intermittently, as it is faster to try to estimate the potential locally from the neighbours. This local estimate does not guarantee that the flow gets really pushed closer to the sink, but in conjunction with the global assignment of potentials, it is a good heuristic which speeds up the convergence.

### 3.3.1.3 Continuous Cuts

In the recent years continuous formulations of many problems from such diverse areas as image denoising, stereo matching, optical flow estimation and segmentation surfaced, which were previously attacked primarily in a discrete setting. Pock et al. [52] for example presented a continuous solution of the globally optimal labeling problem with submodular pairwise interaction term. The nature of continuous approaches which involve partial differential equations, often result in algorithms which are much better suited for data parallel architectures and do not suffer from a grid bias (metrication error). In comparison with the Graph Cuts algorithm, the continuous formulation has the drawback that it is more difficult to tell if the algorithm has converged as often only minimal changes occur in the final iterations of the algorithm when the global optimal state is approached.

First only the binary problem is analyzed and solved. Let  $\Omega$  be a subset of  $\mathbb{R}^2$  representing the image domain. The foreground and background data term are given by the two functions  $f, g : \Omega \mapsto \mathbb{R}$ . Then the binary labeling problem can be state as a minimization of the following functional with respect to  $u : \Omega \mapsto \{0, 1\}$ :

$$\min_u \left\{ \int_{\Omega} f u \, dx + \int_{\Omega} b(1 - u) \, dx + \lambda \int_{\Omega} |\nabla u| \, dx \right\} \quad (3.4)$$

where the function  $u$  induces a segmentation of  $\Omega$  into the non-overlapping regions  $S_{Foreground}$  and  $S_{Background}$  and thus indicates to which label each point belongs: It takes the value 1 for points belonging to the foreground and 0 for points belonging to the background.

$$u(x) = \begin{cases} 1 & \text{if } x \in S_{Foreground} \\ 0 & \text{if } x \in S_{Background} \end{cases} \quad (3.5)$$

Strang [64] showed that a solution to 3.4 can be obtain by solving the problem on the convex set of functions  $u$  with values in the interval  $[0, 1]$  and thresholding the solution at any value in  $[0, 1]$ . Compared to 3.2 the difference one notes is that the binary interaction term is replaced with the length of the gradient of  $u$ . Depending on the metric which is used



to measure this length, different neighbourhood systems of the Graph Cuts algorithm can be simulated. The Manhattan distance for example resembles the neighbourhood system of a 4-connected grid. The properties of the Euclidean distance on the other hand can theoretically not be achieved in the discrete setting as that would require infinite many neighbours (even though in practice 16 and more already achieve results which are very similar).

Olsson et al. [49] recently introduced anisotropic continuous cuts. The anisotropic metric allows to have an image-driven smoothness term and thus a finer control over where the label boundaries are placed. If  $Q$  denotes the diffusion tensor, then 3.4 becomes

$$\min_u \left\{ \int_{\Omega} f u \, dx + \int_{\Omega} b(1-u) \, dx + \lambda \int_{\Omega} |Q \nabla u| \, dx \right\} \quad (3.6)$$

The data term functions  $f$  and  $g$  can be combined into one integral and the constant part may be then ignored, because it does not change the minimum.

$$\min_u \left\{ \int_{\Omega} u(f-b) \, dx + \lambda \int_{\Omega} |Q \nabla u| \, dx \right\} \quad (3.7)$$

In [15] it was shown that if  $J$  is a convex, positively one-homogeneous functional fullfilling

$$J(u) = \int_{-\infty}^{+\infty} J(u^t) dt \quad (3.8)$$

with

$$u^t = \begin{cases} 1 & \text{if } u(x) > t \\ 0 & \text{otherwise} \end{cases} \quad (3.9)$$

then a solution of the minimization problem

$$\min_u \left\{ \alpha \int_{\Omega} u w \, dx + J(u) \right\} \quad (3.10)$$

can be found by thresholding the solution of

$$\min_u \left\{ \frac{\alpha}{2} \int_{\Omega} (u-w)^2 \, dx + J(u) \right\} \quad (3.11)$$

at zero. Thus the binary labeling problem 3.7 is solved by the solution of the mini-

mization problem

$$\min_u \left\{ \int_{\Omega} (u - f + b)^2 dx + \lambda \int_{\Omega} |Q \nabla u| dx \right\} \quad (3.12)$$

This equation is the ROF [57] model augmented with the anisotropic diffusion tensor. Resorting to the Lagrange-Fenchel dual introduced in Equation 2.25, one can derive the dual equation similar to 2.57:

$$\min_u \left\{ \sup_{|\mathbf{p}| \leq 1} \int_{\Omega} (u - f + b)^2 dx + \lambda \int_{\Omega} (Q \nabla u) \cdot \mathbf{p} dx \right\} \quad (3.13)$$

In [75] the projected gradient descent of Chambolle was extended to account for the anisotropic diffusion tensor and a two step algorithm derived which solves Equation 3.13:

$$u^{n+1} = f^n - b^n + \lambda \nabla \cdot (Q \mathbf{p}^n) \quad (3.14)$$

$$\mathbf{p}^{n+1} = \frac{\mathbf{p}^n + \frac{\tau}{\lambda} (Q \nabla u^n)}{\max \{1, |\mathbf{p}^n + \frac{\tau}{\lambda} (Q \nabla u^n)|\}} \quad (3.15)$$

where  $\tau$  is the step size which must be smaller than  $\frac{1}{4}$ .

Extending the solution of the binary labeling problem for continuous cuts to multi-class labeling problems can be done in a similar fashion as with the alpha-expansion move.

First, a new indicator function for each label is introduced, which delineates the area where the respective label is set.

$$\chi_{\alpha} = \begin{cases} 1 & \text{if } x \in S_{\alpha} \\ 0 & \text{otherwise} \end{cases} \quad (3.16)$$

and the energy functional to be minimized becomes

$$\min_u \left\{ \int_{\Omega} u d_{\alpha} dx + \lambda \int_{\Omega} |Q \nabla u| dx + \sum_{\beta \neq \alpha} \int_{\Omega} (1 - u) \chi_{\beta} d_{\beta} dx + \lambda \int_{\Omega} |Q \nabla ((1 - u) \chi_{\beta})| dx \right\} \quad (3.17)$$

where  $d_{\alpha}$  gives the dataterm for associating a point with label  $\alpha$ . The solution to the multi-class labeling problem can now be approximated by solving a series of binary problems of the form of Equation 3.17. Each label is in turn expanded against all other

labels. The indicator function  $u$  from 3.5 now takes on a different meaning:

$$u(x) = \begin{cases} 1 & \text{if } x \in S_\alpha \\ 0 & \text{if } x \notin S_\alpha \end{cases} \quad (3.18)$$

The only remaining problem is the term  $|Q \nabla((1-u)\chi_\beta)|$  which cannot be properly considered in the binary cut. Using the triangle inequality

$$|Q \nabla((1-u)\chi_\beta)| = |Q \nabla(u\chi_\beta)| \leq |Q \nabla\chi_\beta|u + \chi_\beta|Q \nabla u| \quad (3.19)$$

this term can be split up into a contribution to the dataterm and a regular smoothness term. Equation 3.17 thus simplifies to

$$\min_u \left\{ \int_{\Omega} u d_\alpha dx + \lambda \int_{\Omega} |Q \nabla u| dx + \sum_{\beta \neq \alpha} \int_{\Omega} (1-u)\chi_\beta d_\beta + \lambda |Q \nabla\chi_\beta|u dx + \lambda \int_{\Omega} \chi_\beta |Q \nabla u| dx \right\} \quad (3.20)$$

### 3.3.2 Helmholtz Blending

Apart from the very simplistic linear interpolation between two images, more advanced blending techniques have been introduced in the past. Burt and Adelson [13] developed a multi-band approach, which decomposes the input images into band-pass filtered component images. Those filtered images are linearly blended in a transition zone whose size is proportional to the wave lengths represented by the corresponding layer. Perez et al. [50] demonstrated how the Poisson equation can be used to achieve a convincingly seamless combination of two images. In this work an extension of this algorithm is used which bases on [66] and generalizes the Poisson to the Helmholtz equation.

One obtains the Poisson equation as a special case of the Helmholtz equation

$$\sigma u(x, y) - u_{xx}(x, y) - u_{yy}(x, y) = f(x, y) \quad (3.21)$$

by setting the parameter  $\sigma$  to zero. The Poisson equation is often used in conjunction with Dirichlet boundary conditions in order to transfer the gradients from one image region to another region while preserving the boundary values there. Thus, the following constraints are used

$$-\Delta u(x) = f(x) \text{ for } x \in \Omega \quad (3.22)$$

$$u(x) = g(x) \text{ for } x \in \partial\Omega \quad (3.23)$$

which are reasonable for image inpainting and cloning. One drawback is that one image always stays fixed and all other patches are radiometrically attached to this one image. In the case of aerial imagery each image will slightly deviate because of illumination and exposure changes. Other radiometric differences are caused by hazing (intensity variation dependent on the distance and medium between camera center and scene), vignetting (intensity variation dependent on the non-uniform illumination of the sensor) and changes in reflection due to not symmetrical bidirectional reflectance distribution function (BRDF). Normally one would therefore like to avoid to single out one image and register all others to it. Another problem is that one patch is attached after the other, thus forming a large carpet of warped image patches. Using the pure Poisson equation it could happen that by integrating the gradients starting from one initial image, the intensity range drifts away from the original radiometric range.

The Helmholtz equation, on the other hand, allows to tie the result of the blending to the original intensity values of the used patches. Depending on the parameter  $\sigma$  a tradeoff is made between approximating the gradients and approximating the existing intensity values.

By discretizing Equation 3.21 on a grid  $\Omega^h$  with spacing  $h$  one obtains the following equality:

$$\sigma v(i, j) + \frac{-v(i-1, j) + 2v(i, j) - v(i+1, j)}{h^2} + \frac{-v(i, j-1) + 2v(i, j) - v(i, j+1)}{h^2} = f(i, j) \quad (3.24)$$

where  $v$  denotes the discretized approximation of  $u$ . Equation 3.24 can be further simplified and then reads

$$v(i, j) = \frac{h^2 f(i, j) + v(i-1, j) + v(i+1, j) + v(i, j-1) + v(i, j+1)}{4 + \sigma h^2} \quad (3.25)$$

This system of linear equations is solved by a successive overrelaxation scheme (SOR) with Gauss-Seidel iterations. For solving this series of systems of linear equations one

obtains the following algorithm

$$v^{n+1}(i, j) = \frac{h^2 f(i, j) + v^n(i-1, j) + v^n(i+1, j) + v^n(i, j-1) + v^n(i, j+1)}{4 + \sigma h^2} \quad (3.26)$$

where the superscript indicates the number of iteration.

### 3.3.2.1 Multigrid Implementation

The algorithms presented in the previous section are guaranteed to reach a globally optimal solution, if enough iterations of them are calculated. The convergence behaviour of the Poisson equation under various numerical schemes for example has been studied extensively [12]. It was shown that the decrease of the residual error was dependent on the constituent frequencies. High frequencies (relative to the grid discretization) disappear rapidly, whereas low frequencies take much longer to vanish. Informally this can be explained by thinking about the diffusion of information: for lower frequencies the information has to be diffused much farther until the error is recognized and thus the convergence takes longer. It is therefore important to incorporate the above mentioned algorithms in a multigrid framework.

The strategy of a multigrid algorithm can be explained by the following procedure:

- Relax  $Au = f$  on a very coarse grid and interpolate the solution to the next finer grid
- Relax  $Au = f$  on  $\Omega^{2^n h}$  and interpolate the solution to  $\Omega^{2^{n-1} h}$
- Relax  $Au = f$  on  $\Omega^{4h}$  and interpolate the solution to  $\Omega^{2h}$
- Relax  $Au = f$  on  $\Omega^{2h}$  and interpolate the solution to  $\Omega^h$
- Relax  $Au = f$  on  $\Omega^h$

The important aspect is that the solution converges very rapidly on the coarse grid. Then the solution is interpolated to the next finer level and details are added by calculating some more iterations. This procedure is repeated until the finest grid resolution is reached.

### 3.3.3 Composition via a Continuous Color Surface

An alternative solution for compositing the multiple images into one contiguous ortho image is to estimate a continuous color-surface. The prerequisite for this approach to work is

that all images are very well aligned, otherwise the different samples from the input images for one and the same ground point do not correspond to each other. This assumption is supported by the high quality DSM obtained from the fusion process described in Chapter 2.

The idea is to extend the existing methods for fusing multiple observations to vectorial functions. Instead of a scalar function  $\Omega \rightarrow \mathbb{R}$  with  $\Omega \subset \mathbb{R}^2$ , a slightly more complex function  $\Omega \rightarrow \mathbb{R}^3$  is studied which describes a color surface on the image domain. In Chapter 2 it is shown that a second-order regularization is the superior choice for range image fusion. In the case of ortho image generation, however, a first-order regularization is sufficient because a piecewise affine function is not a natural representation of an ortho image. The piecewise constant model in contrast is the natural choice as regions tend to have a homogeneous color. The variational formulation of the problem is therefore

$$E(\mathbf{u}) = \int_{\Omega} \lambda |\mathbf{u}| dx + \int_{\Omega} \sum_{i=1}^N |\mathbf{u} - \mathbf{f}_i| dx , \quad (3.27)$$

where  $\mathbf{u}$  is the color surface which is estimated and should pass through the collected samples  $\mathbf{f}_i$ . The vector norm in this context is defined as the Euclidean length

$$|\mathbf{x}| = \sqrt{\sum_{i=1}^3 x_{(i)}^2} , \quad (3.28)$$

where  $x_{(i)}$  described the  $i$ -th channel of the color vector. In this work the colors are represented as tuples in the RGB color space, however, other representations like the Lab color space would also be possible. The Lagrange-Fenchel conjugate (see Equation 2.25) is used to transform the problem statement into a form which is easier to handle. The Lagrange-Fenchel transform of the vectorial norm in Equation 3.28 is given by

$$f^*(\mathbf{x}^*) = \sup_{\mathbf{x}} \{ \langle \mathbf{x}, \mathbf{x}^* \rangle - |\mathbf{x}| \} . \quad (3.29)$$

The inner product of two vectors can be written in terms of their enclosed angles  $\alpha$ :

$$\langle \mathbf{x}, \mathbf{x}^* \rangle = |\mathbf{x}| |\mathbf{x}^*| \cos(\alpha) \quad (3.30)$$

It is easy to see that this expression is maximized if the cosine of alpha is one, and thus the enclosed angle is zero, which makes the two vectors point in the same direction. The supremum of Equation 3.29 is thus reached if  $\mathbf{x}$  points in the same direction as  $\mathbf{x}^*$ .

Therefore, the supremum depends only on the length of  $\mathbf{x}^*$ , as  $\mathbf{x}$  is always aligned with it:

$$f^*(\mathbf{x}^*) = \begin{cases} 0 & : |\mathbf{x}^*| \leq 1 \\ \infty & : |\mathbf{x}^*| > 1 \end{cases}, \quad (3.31)$$

which can equivalently be expressed using the indicator function as

$$f^*(\mathbf{x}^*) = \mathcal{I}_{|\mathbf{x}^*| \leq 1}. \quad (3.32)$$

For convex functions the biconjugate is the function itself, as stated in Equation 2.26. In the case of the vectorial norm this yields

$$f(\mathbf{x}) = \sup_{\mathbf{x}^*} \{ \langle \mathbf{x}, \mathbf{x}^* \rangle - \mathcal{I}_{|\mathbf{x}^*| \leq 1} \}, \quad (3.33)$$

which can be simplified by constraining the set over which the supremum is calculated to the unit ball:

$$f(\mathbf{x}) = \sup_{|\mathbf{x}^*| \leq 1} \{ \langle \mathbf{x}, \mathbf{x}^* \rangle \}. \quad (3.34)$$

It turns out, that the update scheme derived in Equation 2.68 can be used with the only change, that one dual variable per channel needs to be tracked. Those dual variables are then together reprojected on a three dimensional unit ball.

This formulation is very well suited for parallelization on the graphics hardware - the only drawback is the comparably high memory requirement: for each layer of observations three additional dual variables are necessary, thus limiting the area which can be processed simultaneously. As for the labeling approach, the complete project area therefore has to be subdivided into tiles with a sufficient overlap to ensure matching borders.

## 3.4 Experimental Results

First the three labeling algorithms are evaluated with a synthetic binary labeling problem in order to compare their runtime as well as their results. Afterwards both the labeling as well as the approach using a continuous color surface for automatic generation of ortho images are applied to realword datasets. Their results are analysed, compared and discussed.

### 3.4.1 Comparison of Labeling Algorithms

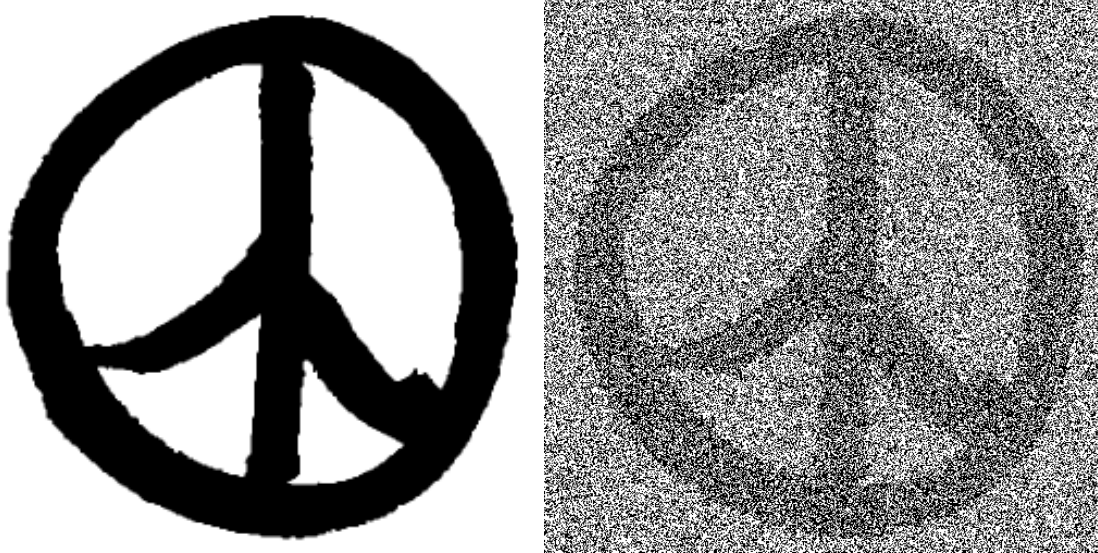


Figure 3.3: On the left hand side the original, unmodified image is shown. 40% of the pixels are then flipped, yielding a distorted image which is used to evaluate the labeling algorithms.

In Figure 3.3 a binary image is shown which is distorted with 40% salt-and-pepper noise (40% of all pixels are flipped), which serves as a synthetic evaluation dataset for the different approaches to solving an isotropic, binary labeling problem.

This noisy image is denoised using each of the algorithms with varying weights for the data and smoothness term. The results for three different smoothness weights are depicted in Figure 3.4. One notices that the Graph Cuts algorithm suffers from metrication errors - increasing the neighbourhood system mitigates this effect, but does not completely get rid of it. The Graph Cuts implementation on the graphics card obviously has exactly the same problems concerning the metrication error as it computes the same result with a different algorithm. The continuous cuts on the other hand yields very smooth results even for higher smoothness weights. This behaviour is beneficial as no border orientation is preferred, whereas the grid-based methods would incur a preference for horizontal and vertical structures.

However, as it becomes visible in the graph of Figure 3.5, the runtime performance of both Graph Cut implementations on the graphics card do not offer the speed-up expected from a specialized implementation on dedicated graphics hardware.



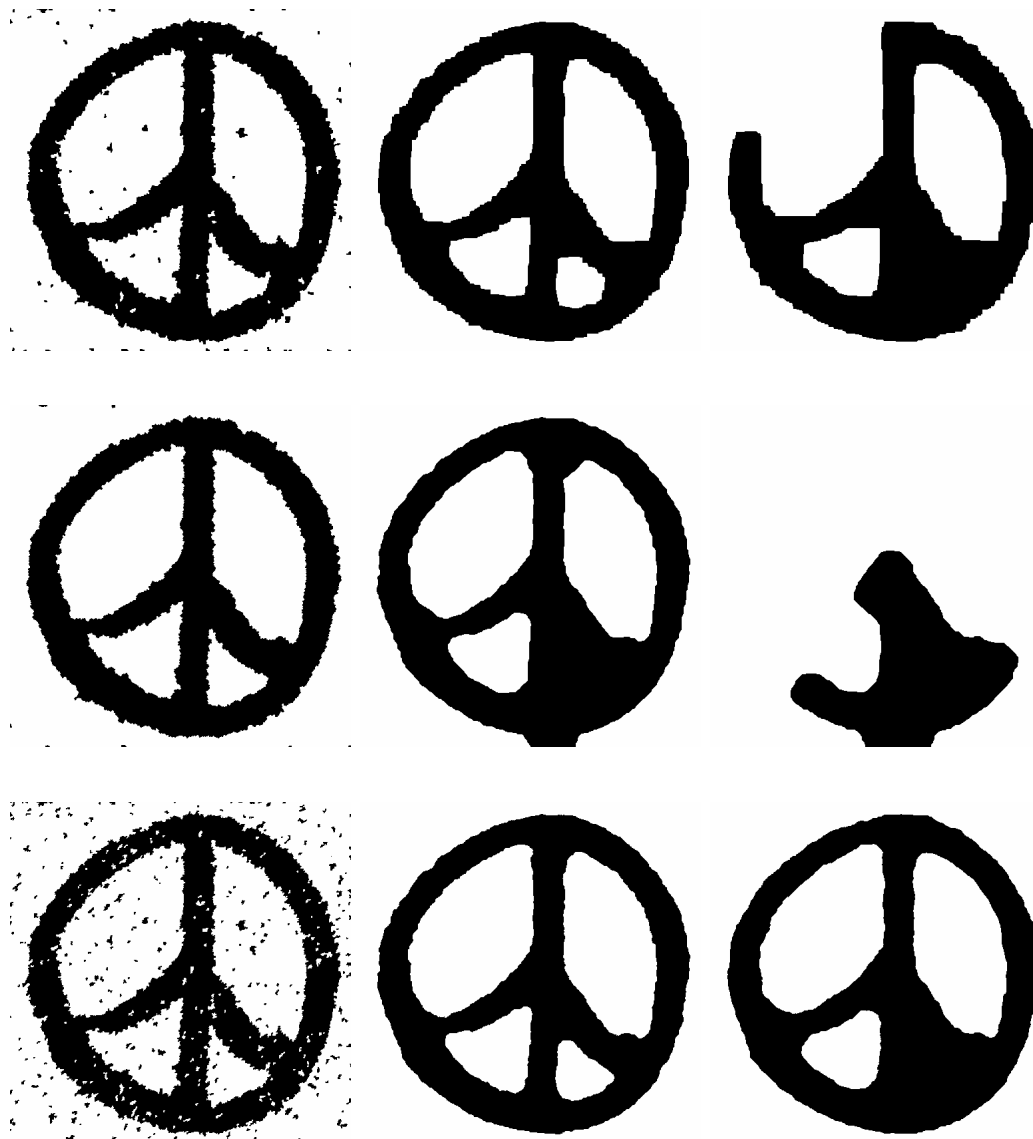
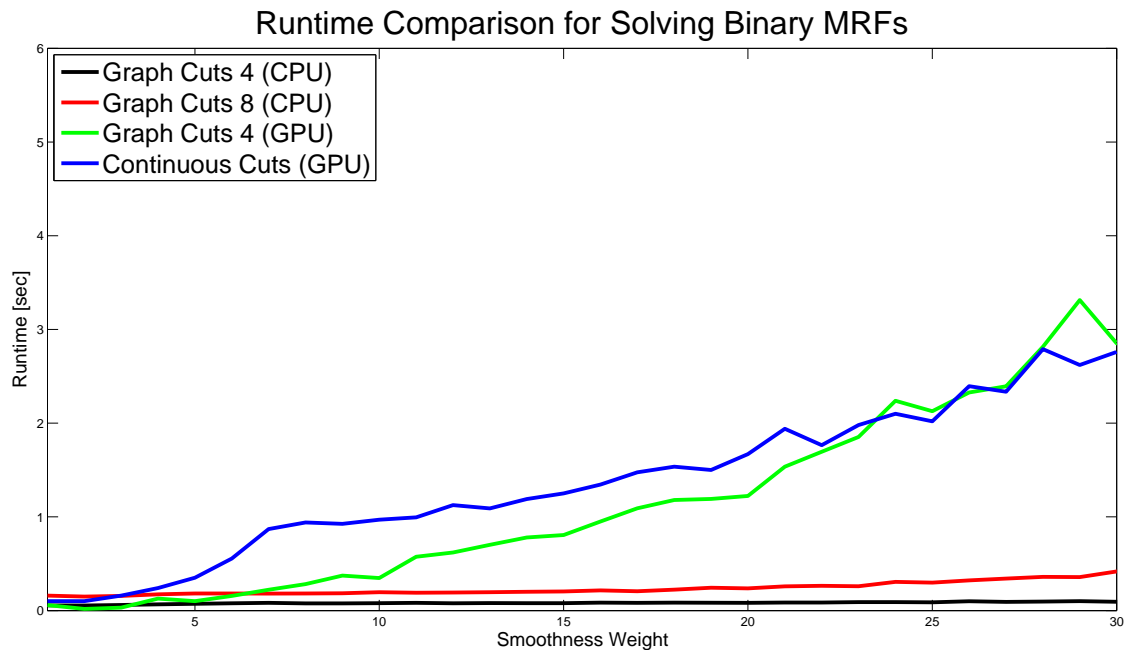


Figure 3.4: Each row shows the results for a different algorithm: for the top row the 4-connected Graph Cuts algorithm was used, in the middle row the 8-connected Graph Cuts algorithm and in the bottom the Continuous Cuts algorithm. From left to right the smoothness weight is increased. It is noticeable that the Graph Cuts algorithm suffers from metrication errors - increasing the neighbourhood system mitigates this effect, but does not completely get rid of it.



(a)

Figure 3.5: This graph illustrates the advantage of the CPU for solving a binary MRF. The smoothness weight has a very limited impact on the runtime for the CPU as the algorithm focuses on remaining excessive flows which are rapidly pushed to available sinks. The implementation on the GPU on the other hand is good at processing pixels in parallel, but for higher smoothness weights only a fraction of the pixels need to be updated and push the flow further. The continuous cuts algorithm has the problem that it is difficult to estimate when a stable state is reached and no more iterations are necessary. In each iteration the update rules are applied to every pixel even if the resulting change is minimal. For high smoothness weights the diffusion process realized in the total variation takes long to converge. Note that using a different hardware or squeezing some cycles out of one specific implementation does not change the linear behaviour of the algorithms on the GPU.

For comparatively small pairwise interaction terms the push-relabel algorithm on the GPU is very efficient and fast, because only few and weak cross links exist among the nodes. But once the excessive flow has to travel far in the graph, the runtime increases. The problem is, that at the beginning most of the connections to the sink are saturated and only remote sinks remain. On the other hand many pairwise interaction terms are also saturated and therefore the remaining excessive flow has to be pushed through several bottlenecks. On the way towards those chokepoints it often happens that they get saturated and some

other chokepoint further away is the closest link to the remaining sinks. This behaviour slows down the convergence and increases the runtime for higher smoothness weights. This situation is depicted in Figure 3.6: The left column highlights remaining capacities to the sink to which the excessive flow from the sources is pushed. This excessive flow is illustrated in the middle column. One notices the frontiers of excessive flow which form inside saturated areas, which travel to the sources. This can take several hundred iterations, as in each iteration the flow moves just one pixel. On the right column the potentials are depicted; black and white pixels already have their final label, whereas the various shades of gray indicated that those pixels are still connected to sources as well as excessive flow.

The runtime performance of the continuous cuts increases close to linearly with the weight of the smoothness term. The stronger relationship with neighbouring pixels encourages the diffusion processes and thus increases the dependence among pixels. Additionally this dependence stretches farther and thus takes longer to reach a stable state.

On the first glance it is surprising that the CPU implementation of the Graph Cut algorithm has the best performance for increasing smoothness weights. However, thinking about the problem both of the above mentioned implementations have, namely their inability to selectively focus on pushing flow or information along a greater distance, it becomes clear, that the trump of the CPU is exactly the lack of parallel execution. Pushing excessive flow over hundreds from pixels thus takes only a fraction of the time than on the graphics card, where it can travel only one pixel per iteration.

It becomes clear, that for processing large areas from aerial images the performance is an important factor and thus the CPU implementation of the Graph Cut seems to be the best choice for solving the labeling problem.

### 3.4.2 Manhattan

The qualitative evaluation is done using real world datasets. The Manhattan dataset poses difficult circumstances as the buildings are very high (and thus the view on the streets is often obstructed) and the city is quite crowded on street level. In Figure 3.7 and 3.8 two examples from that datasets are depicted. Especially the first one is interesting as it highlights the drawback of the approach estimating a continuous color surface. In that case there is that much traffic that the background of the crossing cannot be reliably estimated even given all input samples. The reason of that is that a typical flight consists of many flight lines. In each flight line images are taken in rapid succession, thus if there

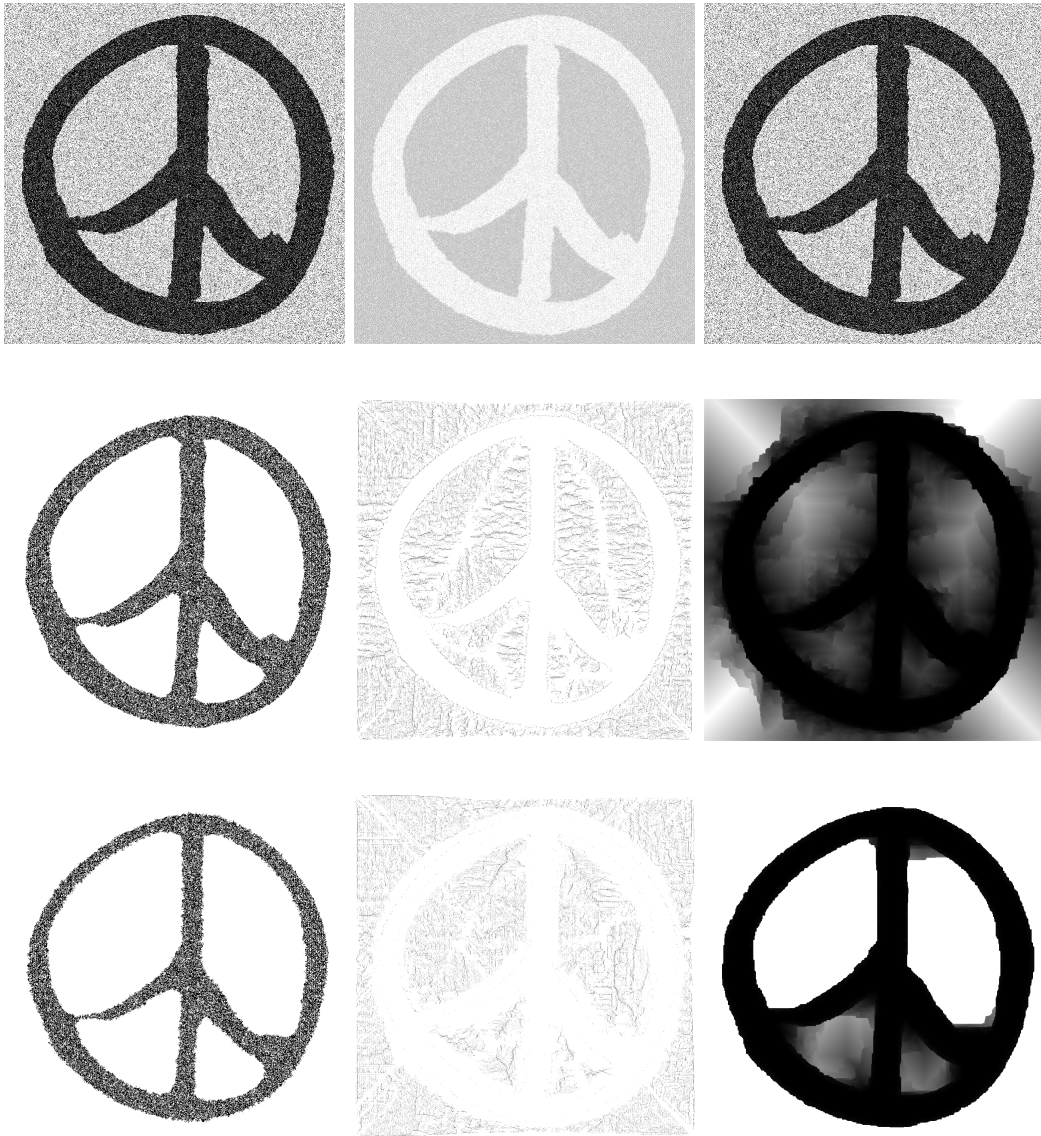


Figure 3.6: This Figure shows the intermediate results of the push-relabel algorithm computed on the GPU. The left column shows where remaining capacities to the sink exist, the middle column indicates where excess flow exists which needs to be pushed to the sink. The right column depicts the estimated potential for each pixel. The top row shows the state after the first iteration, in the middle 50 iterations have passed and at the bottom 100. One notices that towards the end some pixels are already labeled and thus do not participate anymore in the optimization (thus reducing the potential of parallelization). It also occurs that the remaining paths from excess flows to sinks form a bottleneck which is not suited for parallel processing.

is a traffic jam or cars are waiting at the traffic light, it is very probable that all images in one flight line show the same queue of cars. If the aircraft returns in the next flight line at the same crossing usually dozens of minutes have passed. Then again a number of images are taken in succession. If it happens that there is again a queue of waiting cars at that moment, the algorithm fails to reconstruct the street because in each image some car is occluding it. In such cases the result of the labeling approach looks better because the ortho image is consistent in regions coming from the same input image and artifacts can only occur at the stitching boundaries. As those boundaries are designed to minimize the radiometric differences, good results are normally obtained. Additionally the blending is able to hide slight radiometric misalignments present in the input images.

### 3.4.3 Graz

The second dataset which is evaluated consists of 155 images with 85% forward and 65% sideward overlap. The ground sampling distance is about 8cm. Typical results of both approaches are presented in Figure 3.9 and 3.10. In comparison with the Manhattan dataset one notices that because of smaller buildings and less traffic on the street, the continuous color surface is able to consistently reconstruct the correct background and suppress moving objects. Even the most popular scene like the main square in front of the town hall is void of vehicles and pedestrians.

One notices that moving objects like people and the tram are removed by the continuous color surface. This is possible because for each pixel all observations are considered and similar to a majority voting scheme (considering the neighbours of course) outliers are removed.

## 3.5 Conclusions

In this chapter two methods are proposed which demonstrate that using modern computer vision algorithms it is possible to automatically derive high quality ortho images from aerial imagery.

Composing the ortho image via labeling and blending allows to use large portions of the input images on continuous surfaces and thus preserve as many details and fidelity as possible. On the other hand this approach is prone to semantic errors, one and the same moving car could be included twice for example under certain circumstances. Additionally the labeling problem is not well suited for parallel processing on graphics cards.

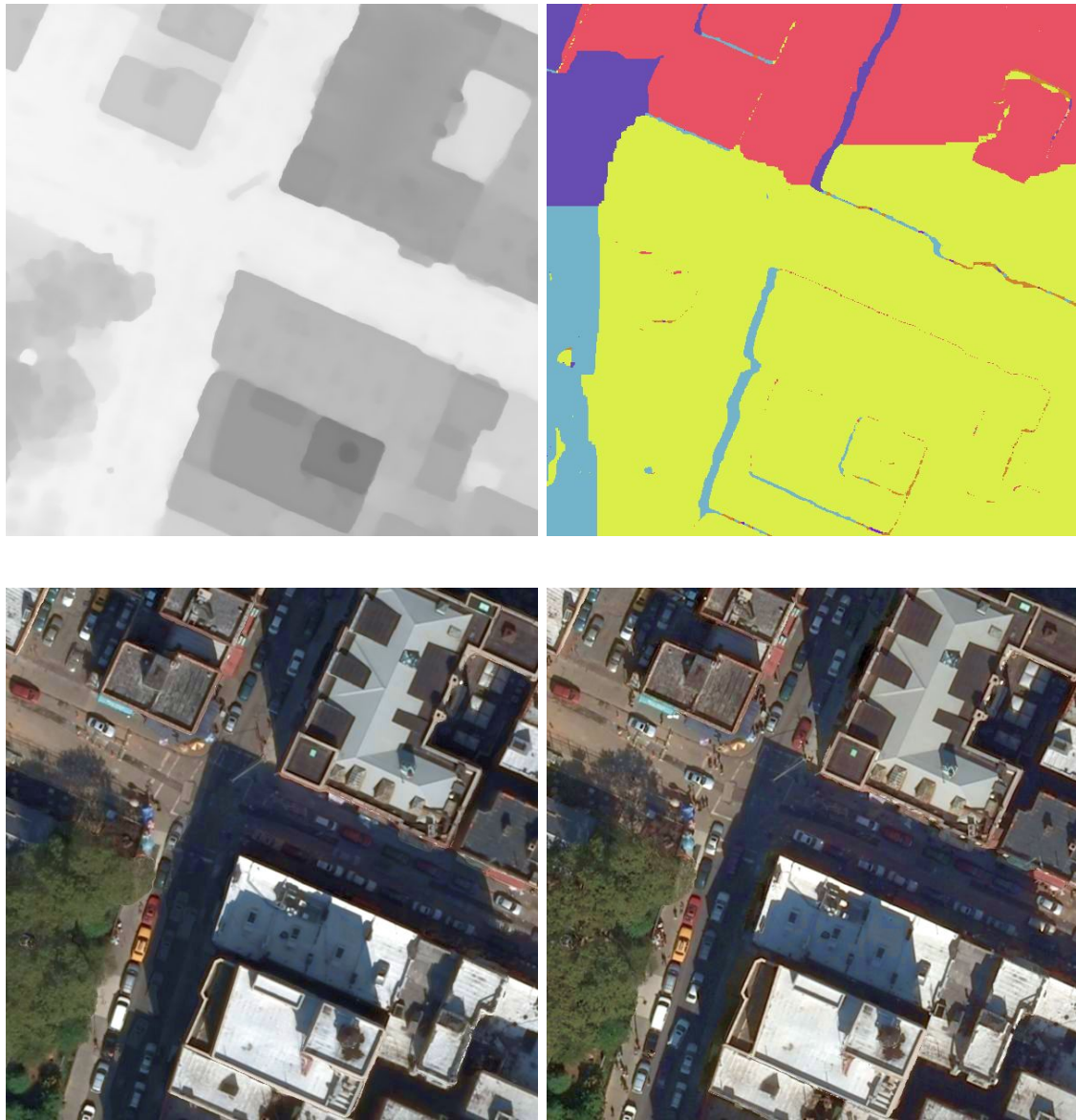


Figure 3.7: A typical result for the ortho image generation with the proposed algorithms is shown. In the top left corner the DSM is shown which is used to warp the input images. In the top right corner the color coded result of the labeling obtained from the Graph Cut algorithm is depicted. Each color represents a different input image. One notices how regions close a building are occluded in one image and need to be filled up by another one. In the bottom row, finally, the ortho images are given which are produced by the continuous color surface algorithm (left) and the labeling (right). Not all moving objects can be removed on the left hand side, because the crossing is too crowded.

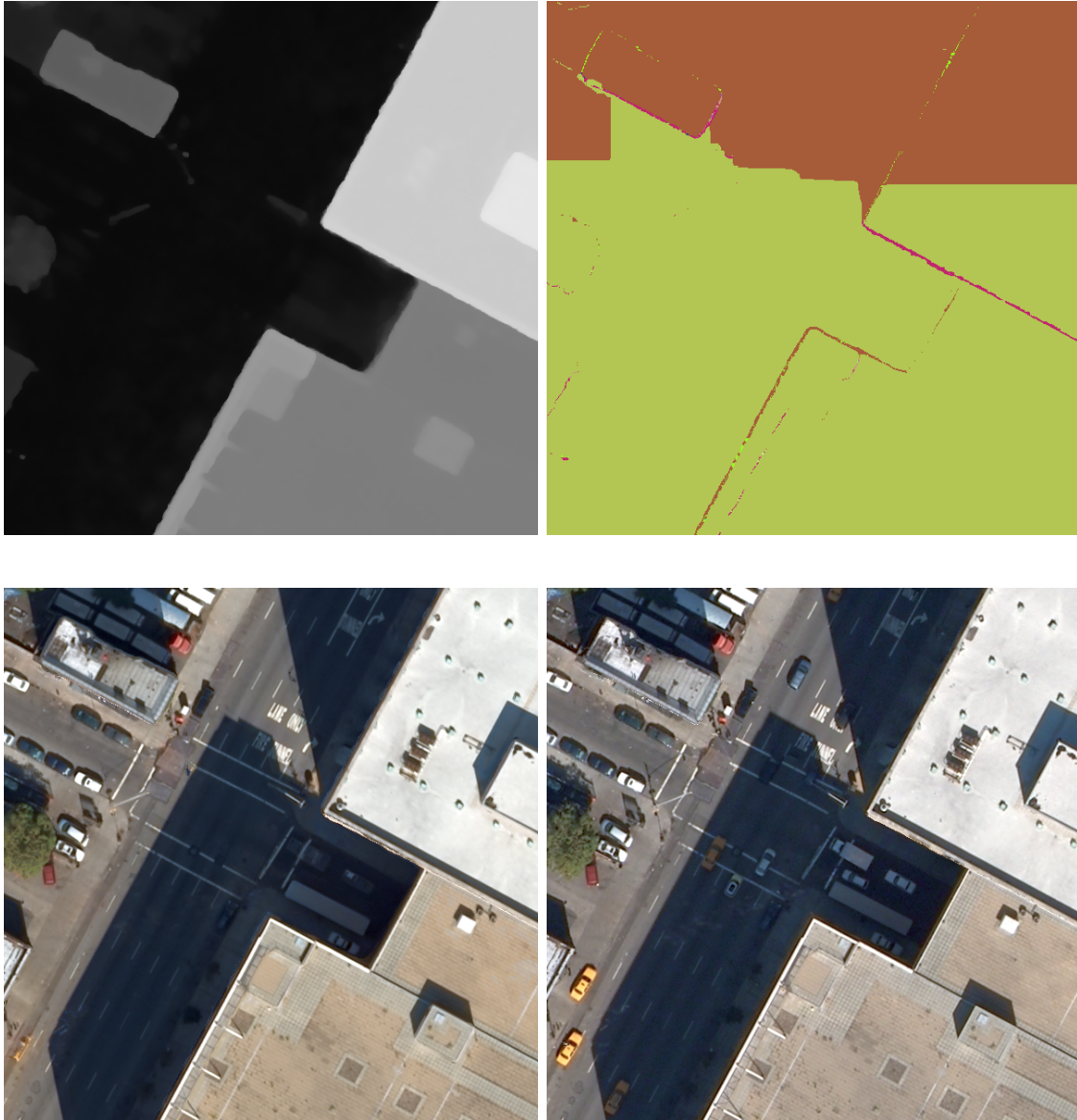


Figure 3.8: In this figure the same layout is being used as in Figure 3.7. In the top row the DSM is shown on the left hand side, and the labeling result on the right hand side. In the bottom row ortho images are shown generated by both algorithms. In this scene the continuous color surface effortlessly removes all moving objects, as there is less traffic.



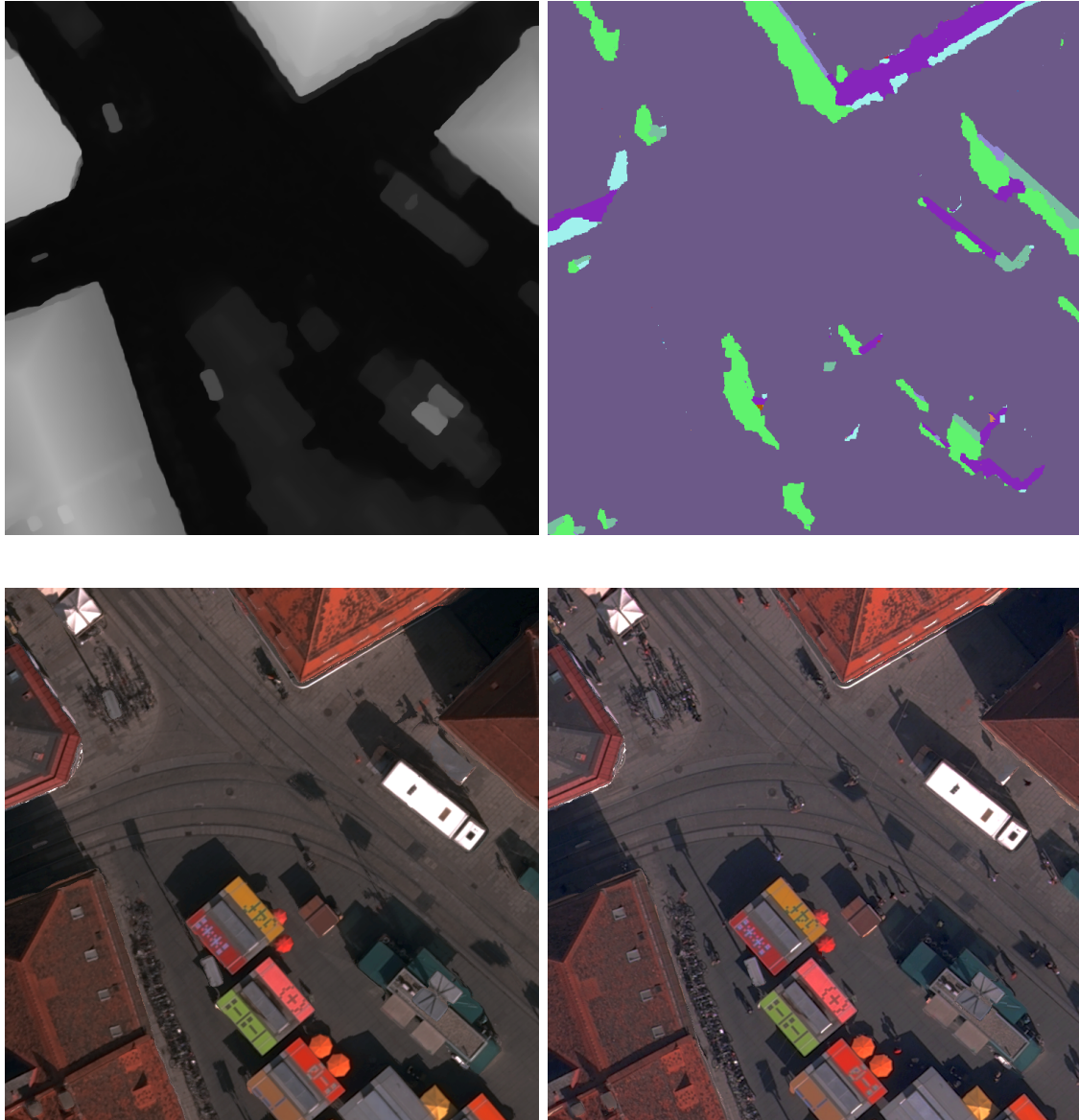


Figure 3.9: In this figure the same layout is being used as in Figure 3.7. In the top row the DSM is shown on the left hand side, and the labeling result on the right hand side. This scene shows the main square in Graz in front of the town hall. The continuous color surface removes all moving objects and presents a very clean ortho image. The purely stitched ortho still contains many pedestrians.



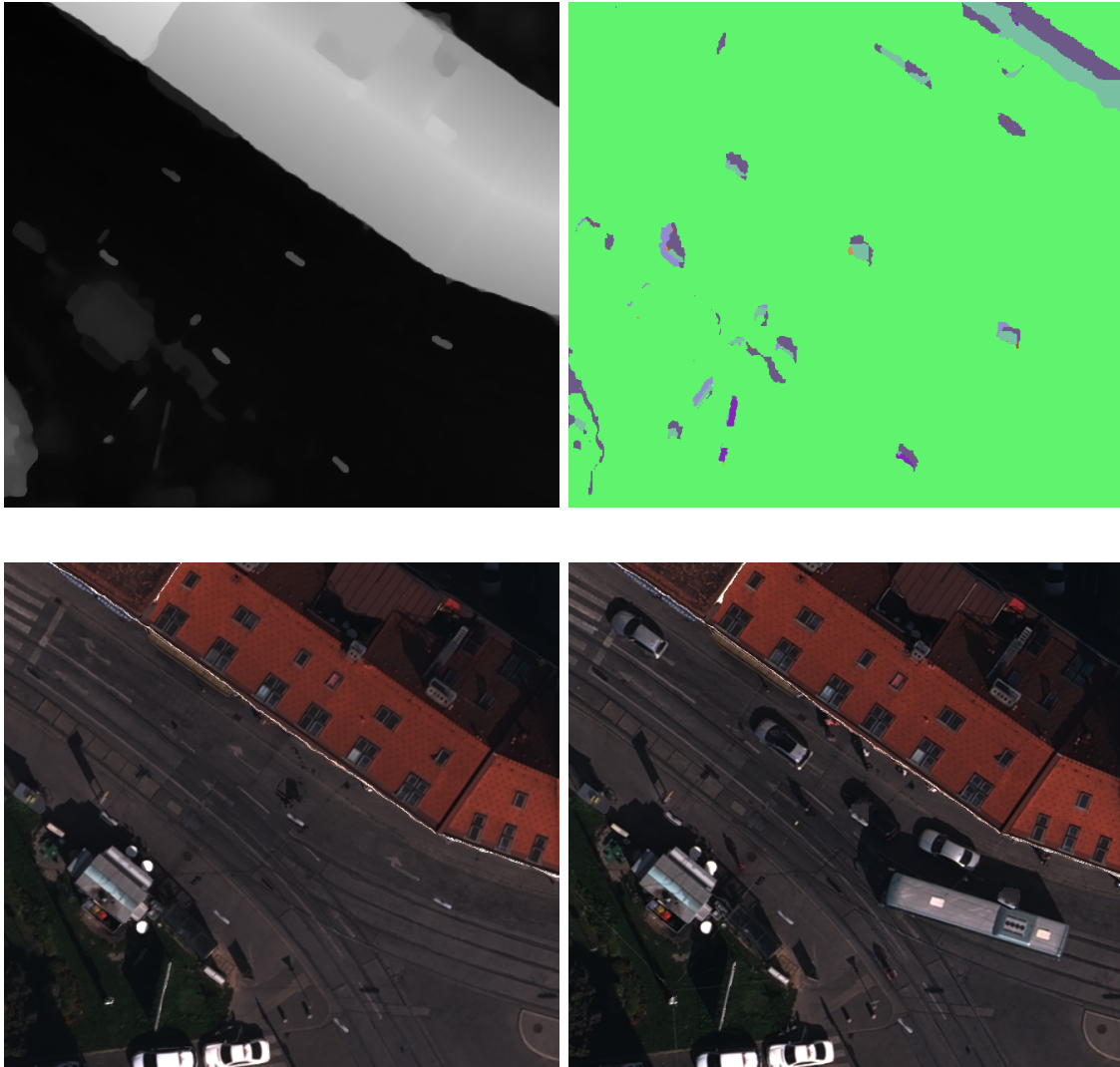


Figure 3.10: This scene shows a different drawback of the labeling approach: overhead street lamps are included in the DSM and severely occlude the street. This forces the labeling to change more often as there the algorithm has to compensate those occlusions. The result looks less natural because of the distortions, one car even has a hole in the middle. The regularization term employed by the continuous color surface successfully removes the cars and also hides the occlusions caused by the street lamps.

Estimating a continuous color surface allows to use all observations to deduce the ortho image. This makes it possible to recognize and suppress moving objects like cars, bikes and people, thus offering a cleaner ortho image. The redundancy in the imagery would even allow to compute a super-resolution ortho image. Estimating the optical flow between the warped patches could compensate for the errors induced by any inaccuracies in the orientation of the images and errors in the estimated geometry upon which the images are warped.

## Chapter 4

# Building Reconstruction

### Contents

---

<b>4.1</b>	<b>Introduction</b>	<b>91</b>
<b>4.2</b>	<b>Overview of the Method</b>	<b>93</b>
<b>4.3</b>	<b>Geometric Primitives</b>	<b>94</b>
<b>4.4</b>	<b>Segmentation</b>	<b>97</b>
<b>4.5</b>	<b>Information Fusion</b>	<b>98</b>
<b>4.6</b>	<b>Experimental Results</b>	<b>101</b>
<b>4.7</b>	<b>Conclusions</b>	<b>105</b>

---

### 4.1 Introduction

Algorithms for the semi- or fully automatic generation of realistic 3D models of urban environments from aerial images are subject of research for many years. Such models were needed for urban planning purposes or for virtual tourist guides. Since the advent of web-based interactive applications like Virtual Earth and Google Earth and with the adoption of 3D content for mashups the demand for realistic models has significantly increased. The goal is to obtain realistic and detailed 3D models for entire cities.

This poses several requirements for the algorithm: First, it should not require any manual interaction because this would induce high costs. This restriction also dissuades the use of cadastral maps as they vary in accuracy, are not readily available everywhere and require careful registration towards the aerial data. Additionally such a dependency increases the cost at large scale deployment. Second, the algorithm should be flexible

enough to generate accurate models for common urban roof structures without limiting itself to one specific type, like gabled roofs or rectangular outlines for example. This also includes the requirement to be able to deal with complex compositions of roof shapes if those happen to be adjacent. Third, the algorithm should have a certain degree of efficiency as it is targeted at thousands of cities with millions of buildings in total. Last, the algorithm should be robust: the visual appearance should degrade gracefully under the presence of noise or bad input data quality.

In the following a survey and assessment of existing algorithms is given, which fail to meet one or more of the above mentioned requirements.

Among the early approaches are feature based modelling methods ([5, 7, 23, 67, 73]) which show very good results for suburban areas. The drawback of those methods is their reliance on sparse line features to describe the complete geometry of the building. The fusion of those sparse features is very fragile as there is no way to obtain the globally most consistent model.

The possibility of using additional data (cadastral maps and other GIS data in most cases) to help in the reconstruction task is apparent and already addressed in many publications ([4, 24, 65]). Such external data, however, is considered manual intervention in our work and thus not used.

A different group of algorithms concentrates on the analysis of dense altimetry data obtained from laser scans or dense stereo matching ([25, 41]). Such segmentation approaches based solely on height information, however, are prone to failure if buildings are surrounded by trees and require a constrained model to overcome the smoothness of the data at height discontinuities. Guhno and Downman ([63]) combined the elevation data from a LiDAR scan with satellite imagery using rectilinear line cues. Their approach was, however, limited to determining the outline of a building. In our work we develop this approach further and embed it into a framework which overcomes the problems described above.

The recovery of three dimensional building models is attracting very much attention of research groups in recent years. Manual efforts are not viable because of the huge costs and sheer size of the problem for even moderately sized cities, therefore an highly automatic procedure is mandatory.

The approach presented in this thesis combines two complimentary sources of information: First, the DSM as an area-based information gives a very good estimate of the overall height and shape of the building - depending on the GSD, this information is available on

a dense grid. The drawback of the DSM is that height discontinuities are often smeared and the buildings are in general not delineated by straight lines. These disadvantages can be remedied by incorporating feature information, which often have a higher accuracy, but are only sparsely available. The combination of those two types is achieved by using the lines to create a segmentation upon which a global optimization algorithm is computed which assigns heights compatible with the DSM.

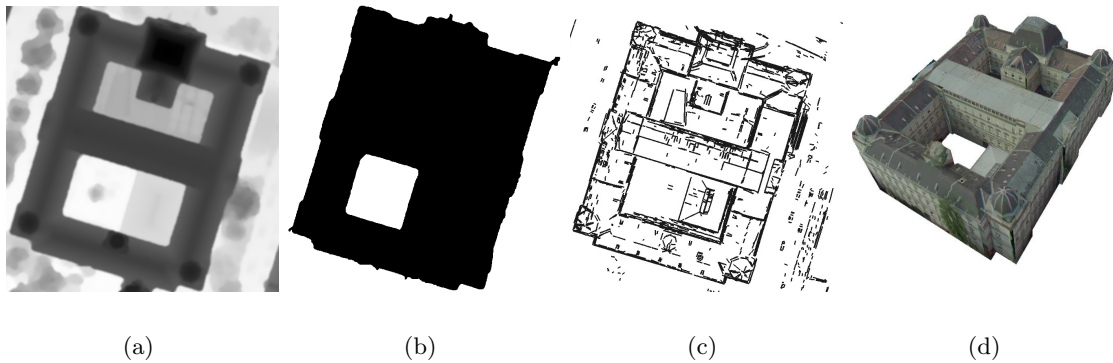


Figure 4.1: These figures depict the data which is used for the reconstruction process: (a) height field, (b) building mask and (c) 3D line segments. Image (d) shows the obtained model by the proposed method.

## 4.2 Overview of the Method

The workflow of the proposed method is outlined in Figure 4.2. Three types of information are necessary as input for the algorithm: Dense height data is generated by a dense image matching algorithm ([33]) (Figure 4.1a, represented as a height field) and gives a good estimate of the elevation, but suffers from oversmoothing at height discontinuities ([58]). Additionally a rough segmentation of the building is required (Figure 4.1b) which could be directly deduced from the height data for example. The third component are sparse 3D line segments (Figure 4.1c) which are obtained from line matching over multiple views ([5]).

The building mask is combined with the dense height data, thus filtering out all 3D points which do not belong to the building. Afterwards the remaining points are grouped into geometric primitives. The geometric primitives are the basic building blocks for assembling the roof shape.

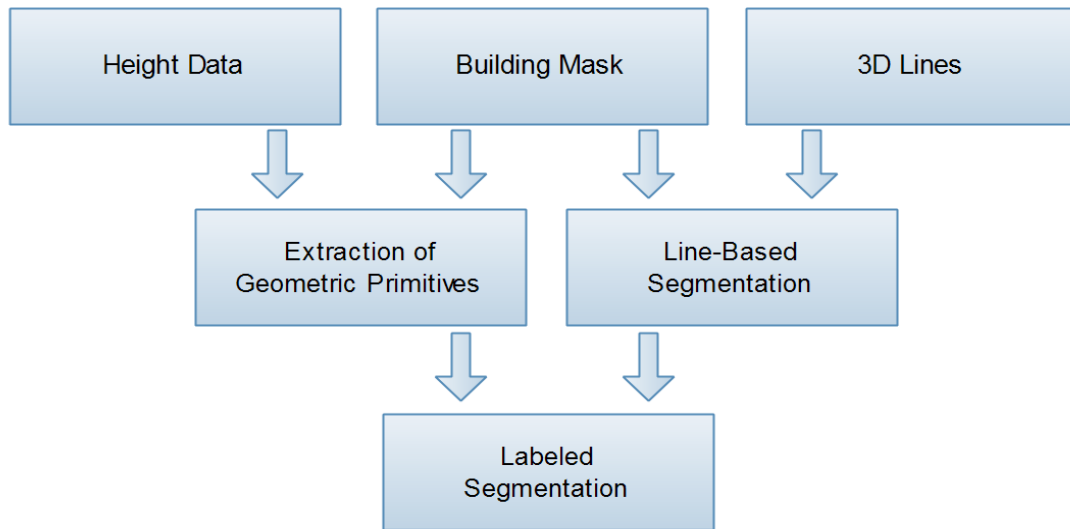


Figure 4.2: Illustration of the single steps of the proposed method: height data and building mask are used to obtain a set of geometric primitives; In parallel the 3D lines are used to generate a segmentation of the building. Finally, a labeled segmentation is produced.

The 3D line segments are projected into the height field and used to obtain a line-based segmentation of the building. The 2D lines of the segmentation form polygons which are then assigned to one of the geometric primitives. Therefore, it is important that the 3D lines capture the location of the height discontinuities as each polygon is treated as one consistent entity which can be described by one geometric primitive. By extruding each of the 2D polygons to the assigned geometric primitive a 3D model of the building is generated.

Note that the algorithm presented in this paper makes no assumptions about the roof shape. Façades are modeled as vertical planes, because the oblique angle of the aerial images does not allow a precise reconstruction of any details.

### 4.3 Geometric Primitives

Geometric primitives form the basic building blocks which are used to describe the roof shape of a building. Currently two types of primitives, namely planes and surfaces of revolution, are used, but the method can be trivially extended to support other primitives. It is important to note, that the detection of geometric primitives is independent from the

composition of the model. This means that an arbitrary amount of hypotheses can be collected and fed into later stages of the algorithm. As the order of discovery of the primitives is not important, weak and improbable hypotheses are also collected as they will be rejected later in the fusion step. If a primitive is missed, the algorithm selects another detected primitive instead which minimizes the incurred reconstruction error.

#### 4.3.1 Planes

Efficiently detecting planes in point clouds for urban reconstruction is well studied and robust algorithms are readily available ([25]). Thanks to the independence of hypothesis discovery and model selection, a region growing process is sufficient in our workflow for the discovery of planes. Depending on the size of the building a number of random seed points are selected, for which the normal vector is estimated from the local neighbourhood. Starting from the seed points, neighbours are added which fit the initial plane estimate. This plane is regularly refined from the selected neighbours. Small regions are rejected to improve the efficiency of the optimization phase. Due to their frequency, close to horizontal planes are modified to make them exactly horizontal, the other oblique ones are left unchanged.

#### 4.3.2 Surfaces of Revolution

Planar approximations of certain roof shapes (domes and spires for example) obtained from plane fitting algorithms, however, are not robust, visually displeasing and do not take the redundancy provided by the symmetrical shape into account. Therefore it is necessary to be able to deal with other shapes as well and combine them seamlessly to obtain a realistic model of the building.

Surfaces of revolution are a natural description of domes and spires and can be robustly detected. Mathematically such surfaces can be described by a 3D curve which moves in space according to an Euclidean motion. Instantaneous kinematics gives a relationship ([54]) between that Euclidean motion parameters and the corresponding velocity vector field. Using that connection it is possible to estimate the parameters of the Euclidean motion in a least squares sense given the normal vectors of the resulting surface.

The equation

$$v(x) = \bar{c} + c \times x \tag{4.1}$$

describes a velocity vector field with a constant rotation and constant translation

defined by the two vectors  $c, \bar{c} \in \mathbb{R}^3$ . If a curve sweeps along that vector field, the normal vectors of all points on the resulting surface have to be perpendicular to the velocity vector at the associated point. Thus

$$n(x)v(x) = 0 \quad (4.2)$$

$$n(x)(\bar{c} + c \times x) = 0$$

holds, where  $n(x)$  gives the normal vector at point  $x$ . With equation (2) it is possible to estimate the motion parameters given at least six point and normal vector pairs  $(x, n(x))$  lying on the same surface generated by such a sweeping curve. In the case of point clouds describing an urban scene the parameter can be constrained by requiring the rotation axis to be vertical. This already reduces the degrees of freedom to two (assuming that  $z$  is vertical) and makes the problem easily solvable:

$$\bar{c} = (0, x, y)^T \quad c = (0, 0, 1)^T$$

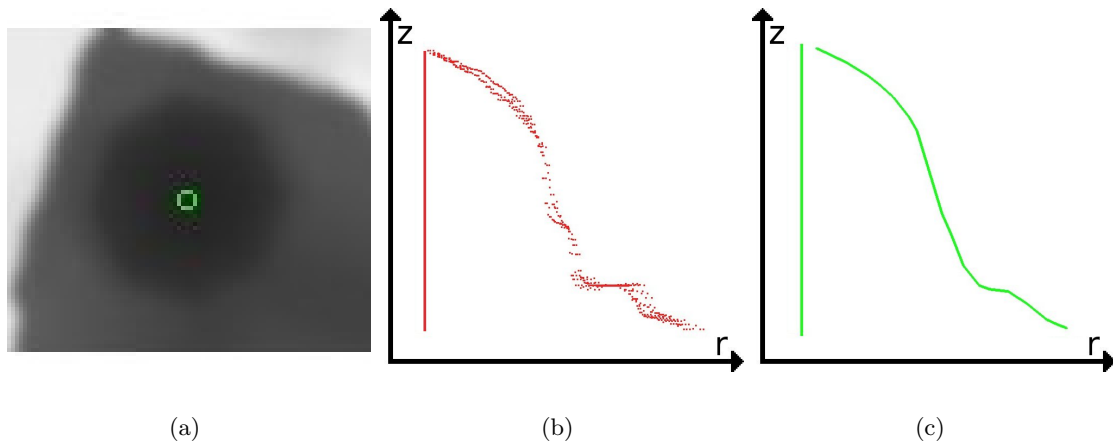


Figure 4.3: Illustrations how starting with the dense height data the 3D curve is derived which generates the dome if it rotates around a vertical axis. (a) Raw height field with the detected axis, (b) all inliers are projected into the halfplane formed by axis and a radial vector, (c) the moving average algorithm produces a smooth curve.

where  $\bar{c}$  gives the position of the axis and  $c$  denotes the vertical rotation axis. The remaining two unknown parameters are estimated by transforming each 3D point with the estimated normal vector  $(x, n(x))$  into a Hough space ([31]). Local maxima in the accumu-



lation space indicate axes for surfaces of revolution. For each axis all inliers are computed and projected into the halfplane spanned by the rotation axis and an arbitrary additional radial vector. The redundancy of the symmetrical configuration can be exploited by a moving average algorithm in order to estimate a smooth curve which generates the surface containing the inliers. Figure 4.3 illustrates those steps with a point cloud describing the shape of a spire.

## 4.4 Segmentation

The goal of the segmentation is to represent the general building structure - not only a rectangular shape - as a set of 2D polygons.

The approach of Schmid and Zisserman ([60]) is used for the generation of the 3D line set that is then used for the segmentation of the building into 2D polygons. A 3D line segment must have observations in at least four images in order to be a valid hypothesis. This strategy ensures that the reliability and geometric accuracy of the reported 3D line segments is sufficiently high. The presence of outliers is tolerable since the purpose of the 3D lines is to provide a possible segmentation of the building. Any 3D line that does not describe a depth discontinuity can be considered as an unwanted outlier which will contribute to the segmentation, but will be eliminated in the fusion stage.

The matched 3D line segments are used to obtain a 2D segmentation of the building into polygons by applying an orthographic projection. The 2D lines cannot be used directly to segment the building, however, as the matching algorithm often yields many short line segments describing the same height discontinuity. A grouping mechanism merges those lines to obtain longer and more robust lines. A weighted orientation histogram - the weights correspond to the length of each line - is created. The principal orientations are detected by finding local maxima in the histogram. Along those directions quasi parallel lines are grouped and merged thus refining their position.

Each grouped line is extended to span the whole building in order to simplify the segmentation process. The lines are splitting the area into a number of polygons. Each polygon is considered to be one consistent entity where the 3D points can be approximated by one geometric primitive.

Figure 4.4 illustrates this concept. The advantage of this approach is that no assumption or constraint of the shape, angles and connectivity of the building is necessary.

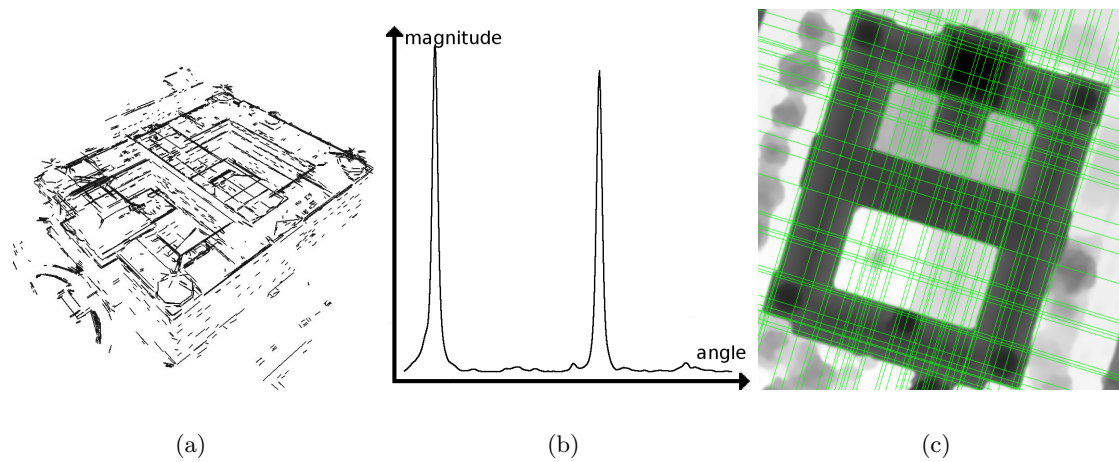


Figure 4.4: Segmentation into polygons: (a) The matched 3D lines are projected into the  $2\frac{1}{2}$ D height field, (b) outliers are eliminated by a weighted orientation histogram which helps to detect principal directions of the building. (c) Along those directions lines are grouped, merged and extended to span the whole building.

## 4.5 Information Fusion

Each polygon resulting from the segmentation is assigned to one geometric primitive (plane or surface of revolution, see Chapter 3). This labeling allows to create a piecewise planar reconstruction of the building - surfaces of rotation are approximated by a rotating polyline and therefore also yield piecewise planar surfaces in the polyhedral model.

The goal of the fusion step is to approximate the roof shape by the geometric primitives in order to fulfill an optimization criterion. In this paper we use the Graph Cuts algorithm with alpha-expansion moves ([10, 37]), but other techniques like belief propagation are suited as well. The goal of this optimization is to select a geometric primitive for each polygon of the segmentation and to find an optimal trade-off between data fidelity and smoothness.

### 4.5.1 Graph Cuts Optimization

The Graph Cuts algorithm finds a very good approximation of the globally optimal solution for a broad range of tasks which can be stated as an energy minimization problem of the following form:

$$E(f) = \sum_{p \in P} D_p(f(p)) + \lambda \cdot \sum_{\{p,q\} \in N} V_{p,q}(f(p), f(q)) \quad (4.3)$$

where  $V_{p,q}(f(p), f(q))$  is called the smoothness term for the connected nodes  $p$  and  $q$  which are labeled  $f(p)$  and  $f(q)$  and  $D_p(f(p))$  is called the data term which measures a data fidelity obtained by assigning the label  $f(p)$  to node  $p$ .

In our approach the segmentation induces a set  $P$  of polygons, where each polygon represent a node of the graph. The neighborhood relationship is reflected by the set  $N$ , which contains pairs of adjacent polygons, ie. polygons sharing an edge. The set of labels used in the optimization process represent the geometric primitives (planes and surfaces of revolution):

$$L = \{\text{plane}_1, \text{plane}_2, \dots, \text{surface-of-revolution}_1, \text{surface-of-revolution}_2, \dots\} \quad (4.4)$$

Thus  $f_p \in L$  reflects the label (current geometric primitive) assigned to node (polygon)  $p \in P$ .

The optimization using polygons is much faster than optimizing for each individual pixel because there are much fewer polygons than pixels. On the other hand it also exploits the redundancy of the height data because it is assumed that all pixels in one polygon belong to the same geometric primitive.

In our context the smoothness term measures the length of the border between two polygons and the data term measures the deviation between the observed surface (obtained from the dense image matching algorithm) and the fitted primitive. The following formulae are used to calculate those two terms:

$$D_p(f_p) = \sum_{x \in p} |\text{height}_{obs}(x) - \text{height}_{f_p}(x)| \quad (4.5)$$

$$V_{p,q}(f_p, f_q) = \begin{cases} \text{length}(\text{border}(p, q)) & \text{if } f_p \neq f_q \\ 0 & \text{if } f_p = f_q \end{cases} \quad (4.6)$$

where  $p$  and  $q$  denote two polygons and  $f(p)$  is the current label of polygon  $p$ . The preset constant  $\lambda$  can be used to weight the two terms in the energy functional. The data term  $D_p$  calculates an approximation of the volume between the point cloud ( $\text{height}_{obs}(x)$ ) and primitive  $f_p$  ( $\text{height}_{f(p)}(x)$ ) by sampling points  $x$  which lie within the polygon  $p$ . This sampling strategy allows to treat all geometric primitives similarly. because they are

reduced to the incurred difference in volume and induced border to other polygons assigned to another geometric primitive. The smoothness term  $V_{p,q}$  penalizes neighbouring polygons with different labels depending on their common border, thus favouring homogeneous regions.

The alpha-expansion move is used in order to efficiently optimize the labeling of all polygons with respect to all discovered primitives. The initial labeling can either be random or a labeling which minimizes only the data term for each individual polygon. After a few iterations (usually less than 5), the optimization converges and all 2D polygons can be extruded to the respective height of the assigned primitive to generate a polyhedral model of the building.

### 4.5.2 Levels of Detail

The second term in Equation (3) regularizes the problem and favors smooth solutions. Depending on the actual value of  $\lambda$  in Equation (3) different results are obtained. Higher values result in fewer and shorter borders at the cost of larger volumetric differences between observed height values and reconstructed models. This feature can be used to generate different models with varying smoothness, trading data fidelity for geometric simplification as smaller details of the building are omitted. An example of such a simplification is shown in Figure ???. The relevant numbers for that building are given in Table 1.

### 4.5.3 Pixel-Based Graphcut

In order to emphasize the importance of incorporating the line information into the reconstruction process, the labeling procedure is applied on the pixel level. The energy functional which is then minimized does not change

$$E(f) = \sum_{p \in P} D_p(f(p)) + \lambda \cdot \sum_{\{p,q\} \in N} V_{p,q}(f(p), f(q)) \quad (4.7)$$

only the construction of the graph is different. The binary interaction term is also simplified to

$$V_{p,q}(f_p, f_q) = \begin{cases} 1 & \text{if } f_p \neq f_q \\ 0 & \text{if } f_p = f_q \end{cases} \quad (4.8)$$

as all edges now have an same length.

## 4.6 Experimental Results

First a comparison between pixel-based and segmentation-based reconstruction is given. The result of both reconstruction techniques is depicted in Figure 4.5. It is noticeable that the pixel-based result dramatically suffers at height discontinuities. Buildings often feature straight lines and sharp corners - both of these are encouraged and highlighted in the segmentation of the building and thus present in the reconstruction. From a point of view of energy minimization, however, the pixel-based approach allows to reach lower energies, as the optimal solution can be better approximated because there are more pixels than polygons in the segmentation. The prior knowledge induced by the segmentation improves the result, but raises the energy. For the pixel-based approach it would therefore be necessary to incorporate the segmentation results in the binary interaction term to encourage label boundaries along those lines. Still, the subpixel accuracy of the segmentation would be lost. The runtime performance also suffers as the pixel-based Graph Cuts algorithm has many more nodes.

### 4.6.1 Graz

The first illustrative experiment was conducted on a test data set of a Graz. The ground sampling distance of the aerial imagery is 8cm. The examined building features four small cupolas at the corners. Additionally one façade is partially occluded by trees. Figure 4.6 shows the results of the reconstruction process. The texture of the façades is well aligned, implying that their orientation was accurately estimated by the 3D line matching. The domes are smoothly integrated into the otherwise planar reconstruction. Even the portion occluded by the tree has been straightened by the extension of the matched 3D lines.

### 4.6.2 Manhattan

The next example is taken from a data set of Manhattan, New York. This building shows that the reconstruction algorithm is not limited to façades perpendicular or parallel to each other. Figure 4.7 illustrates the effect of the smoothness term in the global optimization energy function. Various runs with different values for  $\lambda$  yield a reduced triangle count as the geometry is progressively simplified. Table 4.1 gives details about the solution for different values of  $\lambda$ . The Graph Cuts algorithm allows to find a globally optimal tradeoff between data fidelity and generalization. Those properties are expressed by the decreased length of borders and number of labels (which translate in general to fewer triangles) at the

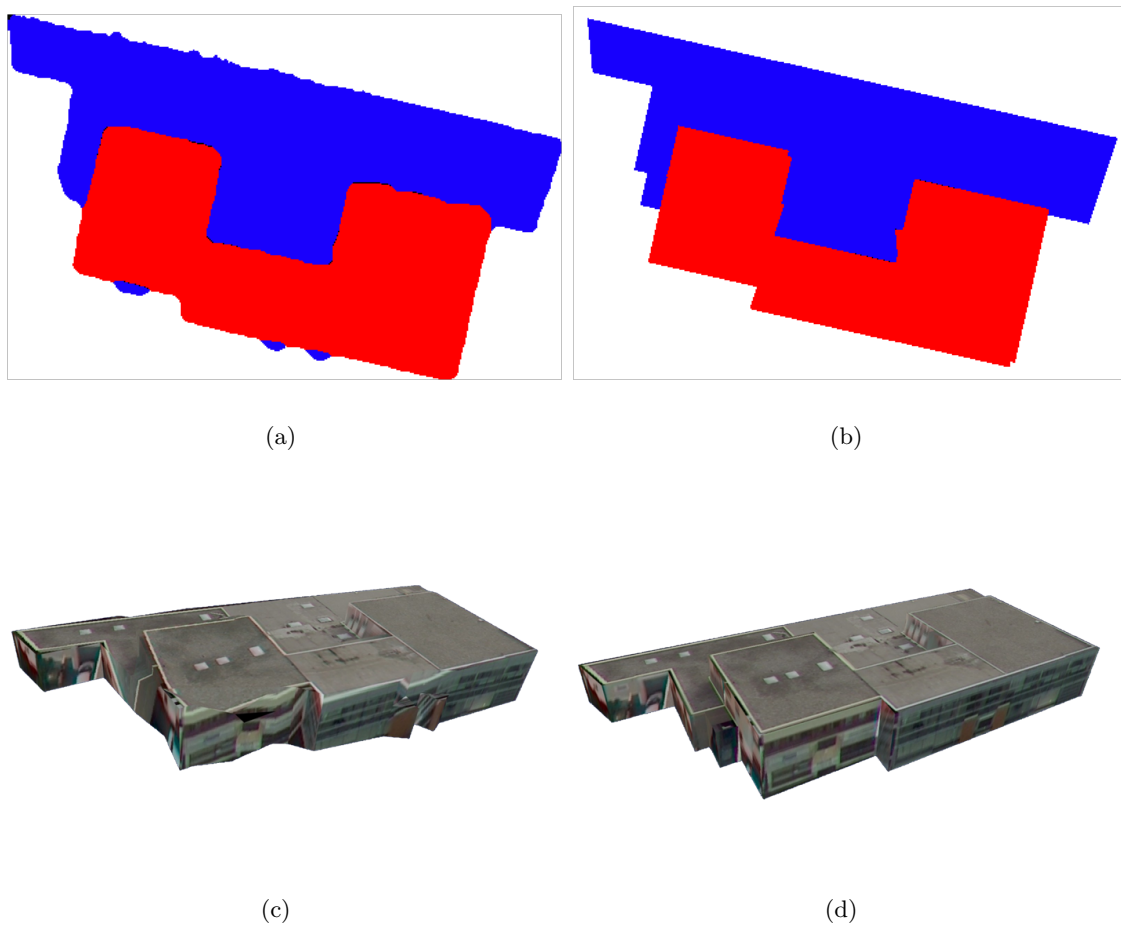


Figure 4.5: This Figure compares the results of the pixel-based (a) reconstruction with the segmentation-based (b) one. The façades of the pixel-based approach are not straight thus degrading the visual appearance, which can be observed very well by comparing (c) and (d).

cost of an increase of the average difference between reconstructed and observed surface.

Detailed views of typical results from the Manhattan data set are shown in Figure 4.10 and 4.11. The reconstruction of rectangular buildings is very successful, even though huge portions of their façades are occluded by trees. The integration of surfaces of revolution realistically models domes and spires (see ??b and ??b). It is important to note that for the purpose of visualization the surfaces of revolution are converted to triangle meshes by sampling them regularly (2m radially with 45 degrees of angular separation).

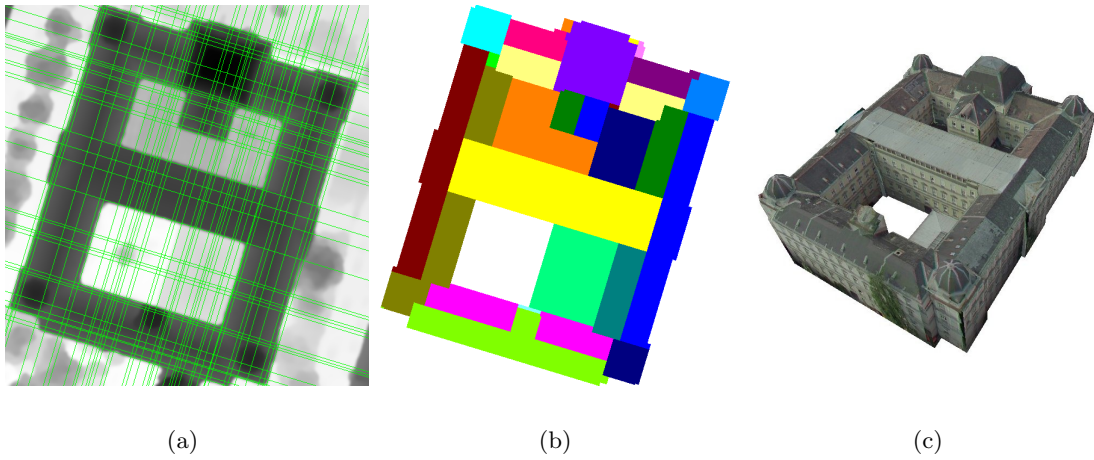


Figure 4.6: The stages of the reconstruction are illustrated by means of the building of the Graz University of Technology: (a) Segmented height field, (b) labeled polygons after the Graph Cuts optimization, (c) screenshot of the reconstructed model ( $\lambda = 5$ )

$\lambda$	#Labels	#Triangles	$\Delta$ Volume [ $m^3$ ]	Border Length [ $m$ ]
5	7	79	1210.79	710.4
10	6	69	1677.19	349.4
20	4	42	1699.31	337.0
100	3	33	2293.36	290.4

Table 4.1: The impact of the smoothness parameter  $\lambda$  on the reconstructed model. The number of unique labels used after the Graph Cuts optimization iterations decreases as well as the number of triangles in the polygonal model.  $\Delta$  Volume denotes the estimated difference in volume between the surface obtained by dense image matching and the reconstructed model (data term). The last column refers to the accumulated length of all borders in the final labeling (smoothness term).

### 4.6.3 Quantitative Evaluation

Apart from judging the visual appearance of the resulting models, we assess the quality of the reconstructed models by comparing them to a ground truth which was obtained manually from the same imagery. For this purpose we use a stereoscopic device to trace the roof lines in 3D. Those roof lines are connected to form polygons and then extruded to the ground level. Those manually reconstructed models are considered ground truth data in this paper. Using this procedure the whole data set from Manhattan (consisting of 1419 aerial images at 15cm ground sampling distance) was processed yielding 1973 buildings.

A comparison of manual and automatic reconstruction for one building is illustrated in

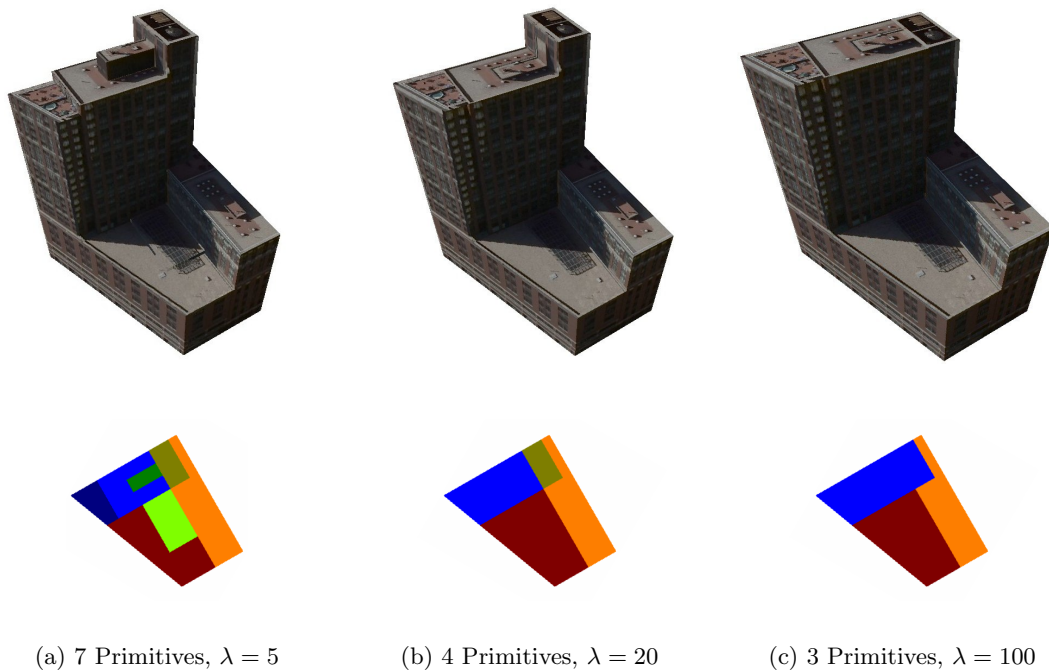


Figure 4.7: Levels of Detail: The same building was reconstructed with different values for  $\lambda$ . The number of geometric primitives used to approximate the shape of the roof is decreasing with higher values for  $\lambda$ . In the upper row a screenshot of the reconstruction is depicted, below are illustrations of the matching labeling obtained by the Graph Cuts optimization.

Figure 4.8. Both building models are converted into a height field with a ground sampling distance of 15cm. This makes it easy to determine and illustrate their differences. Figure 4.9 gives a break down of the height differences as a cumulative probability distribution. Those graphs give the percentage of pixels where the height difference between manual and automatic reconstruction is lower than a certain threshold. Analysis of this chart shows that for the whole data set of Manhattan (1973 buildings) 67.51% of the pixels have a height difference smaller than 0.5m, 72.85% differ by less than 1m and 86.91% are within 2m. There are two main reasons for discrepancies of height values: On the one hand there are displacement errors of roof edges which lead to large height differences, depending on the height of the adjacent roof. On the other hand the human operator is able to recognize small superstructural details on the roofs like elevator shafts and air conditioning units which cause height differences usually below 2m. Those small features are sometimes missed by the automatic reconstruction.



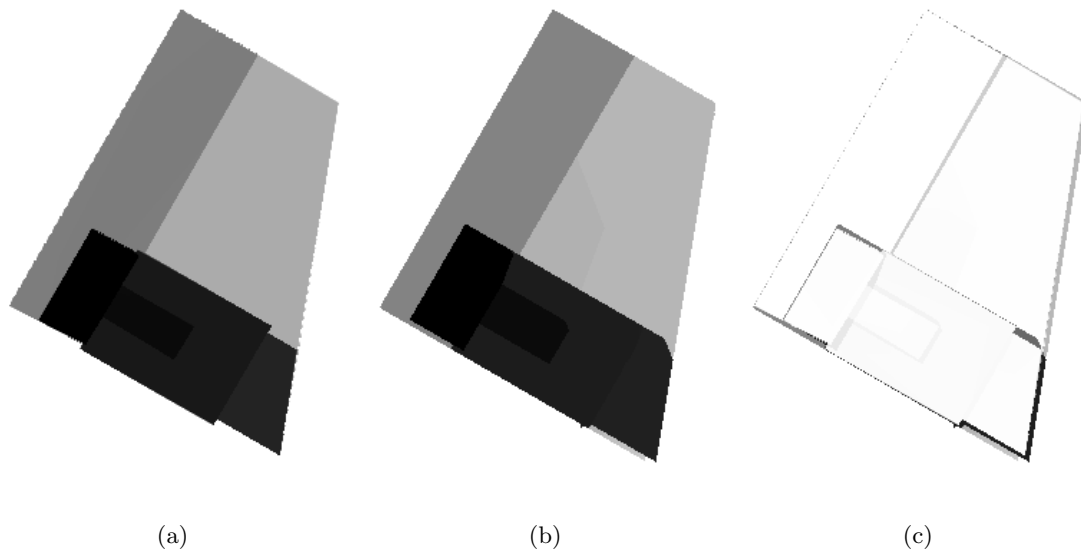


Figure 4.8: Quality assessment with a manually generated ground truth: In (a) and (b) the height fields for the manually and automatically reconstructed building are shown, in (c) the height differences are shown. The largest difference in the placement of edges is about two pixels, which is about 30cm.

## 4.7 Conclusions

In this section a novel approach for reconstructing building models from aerial images was presented by combining 3D line segments and dense image matching algorithms with a global optimization technique. The framework is able to use arbitrary basic geometric building blocks to describe the roof shape. The proposed surfaces of revolution elegantly describe domes and spires which are difficult to recover with an approach based on planes only. The combination of line based features and dense image matching algorithms using a global optimization technique is very promising and is not restricted to the reconstruction of urban scenes from aerial imagery. Additionally it allows for the generation of different globally optimal levels of detail.

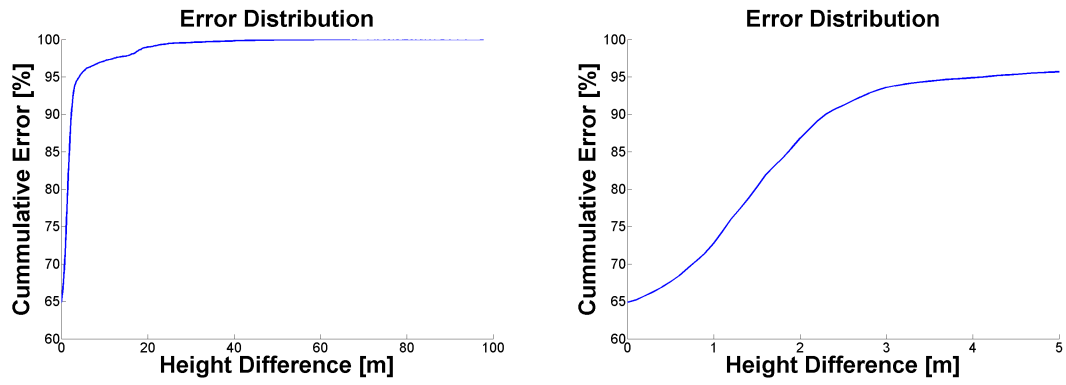


Figure 4.9: The cumulative probability distribution of the height difference for manual and automatic reconstruction. The graph shows the error distribution for 1973 buildings from a data set of Manhattan, New York. The left image shows the graphs for height differences up to 100 meters; the right graph zooms on differences up to five meters.

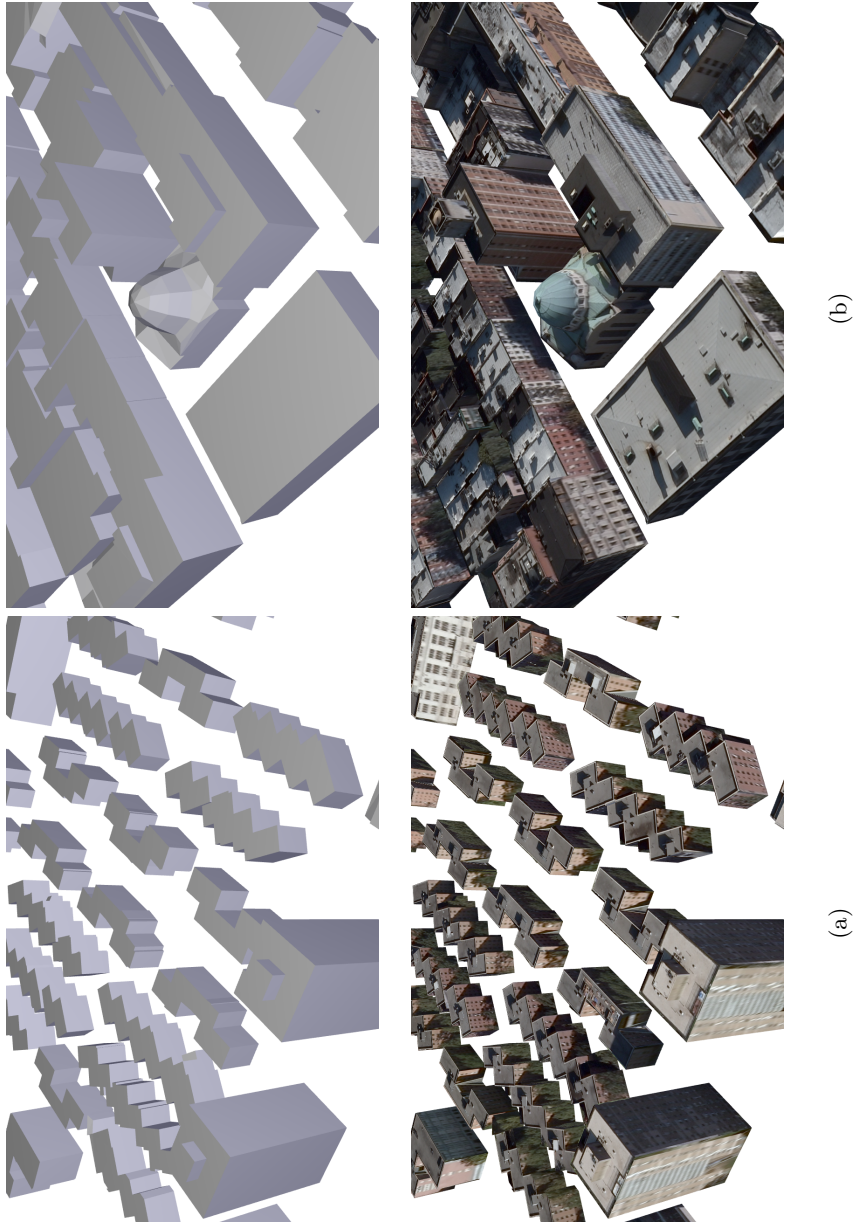


Figure 4.10: Two detailed views of typical results for different types of buildings from the Manhattan data set: (a) rectangular buildings, (b) rectangular building with nicely integrated dome.

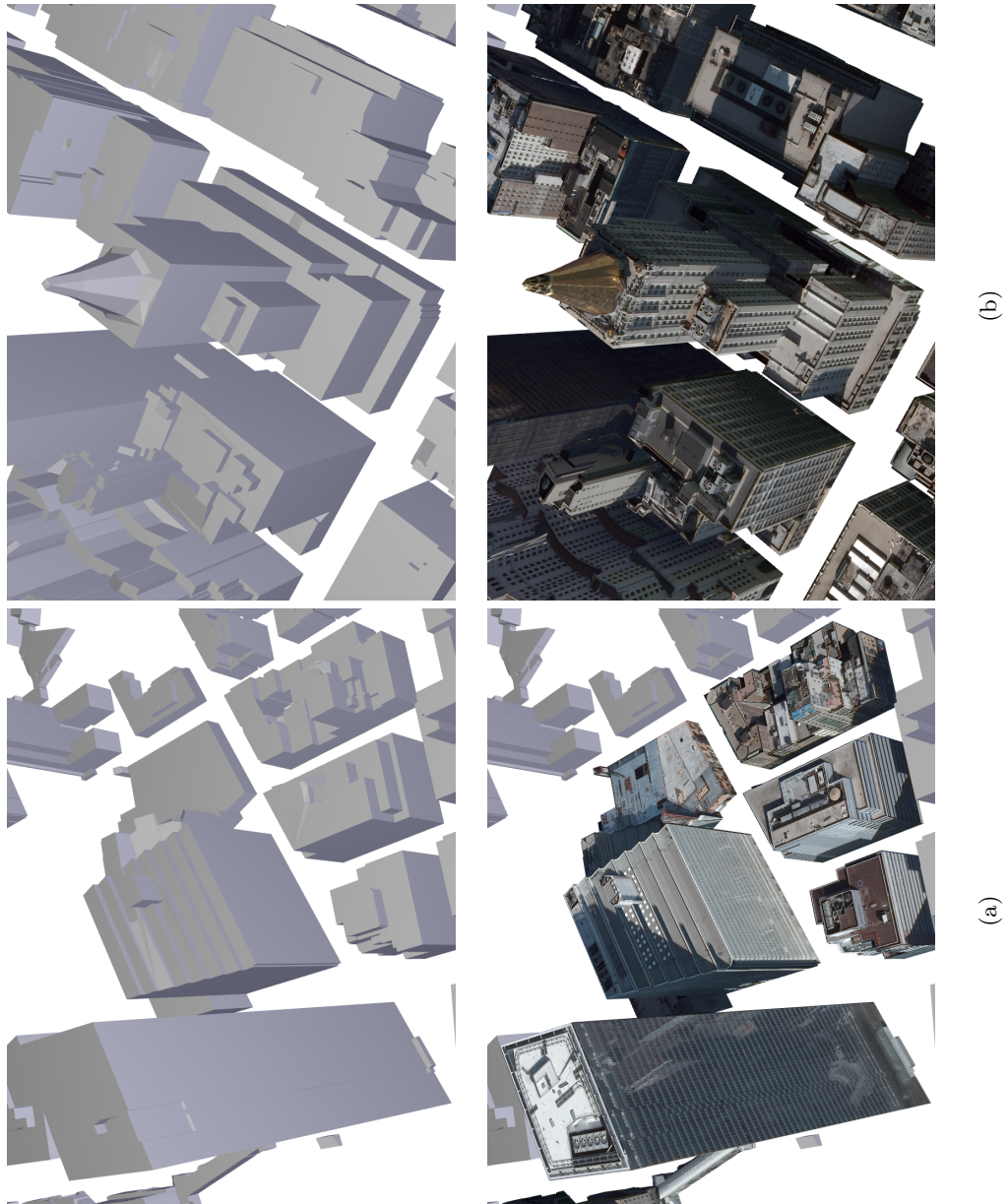


Figure 4.11: Two detailed views of typical results for different types of buildings from the Manhattan data set: (a) skyscrapers in downtown and (b) skyscraper with a spire.

# Chapter 5

# Conclusions

## Contents

---

<b>5.1 Summary and Conclusions . . . . .</b>	<b>109</b>
<b>5.2 Future Work . . . . .</b>	<b>110</b>

---

## 5.1 Summary and Conclusions

In this thesis algorithms for a fully automatic pipeline for processing aerial images are presented. The focus of the contribution lies on the reconstruction of urban environments and the extraction and simplification of individual building models. The general approach of these algorithms is to formulate the encountered problem as an energy minimization task. Continuous and graph-based methods are then used to efficiently find minimizers to these energies.

In the second chapter a detailed comparison of convex energies for range image fusion is presented. Common convex formulations of the image denoising problem are extended to include multiple dataterms which in turn can then be used for robust range image fusion. An efficient primal-dual algorithm is presented for each energy function which is guaranteed to converge to the globally optimal result. Among the investigated methods the Total Generalized Variation approach performs quantitatively and qualitatively best in the evaluations with synthetic and real datasets. This is no surprise as this methods approximates the collected range samples with a piecewise affine function which fits the scenario of an urban surface very well. First-order methods persistently exhibit stair-casing effects which degrade their performance. The robust  $L^1$  dataterm on the other hand is necessary to lend resilience against outliers to the method. Energy models lacking

this robustness collectively fail in the presence of outliers.

The third chapter focuses on the generation of true ortho images. Those ortho images are obtained by ortho-rectifying the input images using the geometry induced by the DSM. Afterwards the warped patches are stitched together to form a contiguous true ortho image. The labeling problem is formulated as an energy minimization task in order to reduce the length and radiometric differences along the label boundaries. Alpha expansion moves are used to reduce the multi-class labeling problem to the binary decision problem. Continuous and graph-based methods are compared for solving this binary problem. A runtime analysis of the various algorithms to solve the binary decision problem shows that the GPU does not offer compelling advantages over the CPU algorithms. The well known metrication error of graph-based methods is demonstrated which is not present in the continuous formulation. This metrication error, however, is not noticeable in the blended ortho images. This approach achieves visually pleasing results by reusing as large as possible portions of images which. On the other hand this approach does not exploit the redundancy inherent in the imagery. The variational algorithms are then adapted to cope with vectorial functions. This way the generation of an ortho image can be mathematically treated in a similar fashion as the range image fusion. The redundancy can then be exploited to filter out moving objects.

Extracting individual building blocks and simplifying them into polygonal meshes is discussed in chapter four. The footprint of each building block is partitioned into polygons by a line-based over-segmentation. The motivation for this is to reduce the complexity of the problem as there are commonly significantly more pixels than polygons for a given building block. Secondly, those lines provide sub-pixel accuracy and encourage straight façades. The shape of the roof is determined by assigning a geometric primitive to each polygon of the footprint. Those geometric primitives are independently discovered and retrieved from the DSM. The assignment itself takes the volumetric difference between DSM and reconstructed building as well as the smoothness of the roof structure into account. The amount of complexity of the roof shape can be controlled by a single parameter. Comparing the automatically obtained results with manual ground truth data, it is shown that the majority of the buildings incur only a small volumetric error.

## 5.2 Future Work

The proposed algorithms feature a high degree of robustness and require only a small amount of parameters (for which often there exists a reasonable default value). However,

there are still areas where improvements are possible.

Even though the range image fusion is performed with a robust data term, it still depends on the range samples obtained from a dense image matching algorithm. For water bodies, for example, no matching procedure is known to yield accurate and consistent samples (as the surface of the water is in perpetual motion and changes appearance in each image). Therefore the surface of water areas is often erroneously estimated. This problem could be addressed in two ways: first, the classification information could be used to identify samples from water areas which are then rejected from the fusion step. Even though this would not give any samples to work with for that area, the regularization term of the energy functional would allow to infer a smooth interpolation between the shores. Another possibility would be to exploit the scattered nature of the range samples. This could be achieved by introducing a spatially varying weight for the data term, which could depend on the variance of the samples for a given location in the DSM for example. One drawback of this approach is that a large variance of the range samples could also be caused by other factors, like reflective roofs (causing many mismatches) or truly three dimensional objects like the Eiffel tower or a bridge with a complicated pattern cross beams.

A problem neglected in this thesis is the radiometric balancing of the input images for ortho image generation. The labeling and blending processes aim to reduce radiometric differences along image boundaries, but a different hue or brightness is not compensated for by those algorithms. In order to reduce the radiometric differences among all available images, a procedure similar to the geometric bundle adjustment for obtaining the camera positions and orientations has to be applied. Such a radiometric bundle adjustment would collect radiometrically corresponding samples (via the DSM for example) and try to balance the images compensating for different exposures and lighting conditions. This radiometric aligning not only helps in hiding differences between images in the ortho image, but the classification is also facilitated because the intra-class variability of the spectral features is reduced.

Future work for the building reconstruction and simplification involves the investigation of other geometric primitives for decomposing the roofs and methods to exploit symmetries encountered in common roof shapes like gabled roofs. Further research will be needed to evaluate the possibilities of this approach in other applications like streetside imagery.





# Appendix A

## Acronyms and Symbols

### List of Acronyms

AT	Aerial Triangulation
BRDF	Bidirectional Reflectance Distribution Function
CUDA	Compute Unified Device Architecture
DTM	Digital Terrain Model
DSM	Digital Surface Model
GCP	Ground Control Point
GIS	Geographical Information System
GPU	Graphics Processing Unit
GSD	Ground Sampling Distance
IBMR	Image Based Modeling and Rendering
LiDAR	Light Detection and Ranging
MRF	Markov Random Field
NIR	Near Infrared
PDE	Partial Differential Equation
POI	Point of Interest
SOR	Successive Over-Relaxation
SVM	Support Vector Machine

## List of Symbols

$\nabla$	Nabla operator
$\Delta$	Laplace operator
$\partial$	Border of
$ \cdot $	Measure (length) in the Euclidean sense if no other metric is mentioned

## Bibliography

- [1] Albert, J., Bachmann, M., and Hellmeier, A. (2003). Zielgruppen und Anwendungen für Digitale Stadtmodelle und Digitale Geländemodelle. Technical report, Census of the working group "Anwendungen und Zielgruppen" of the SGI3D.
- [2] Anderson, R. J. and Setubal, J. C. (1992). On the parallel implementation of goldberg's maximum flow algorithm. In *Proceedings of the fourth annual ACM symposium on Parallel algorithms and architectures*, pages 168–177.
- [3] Aujol, J.-F., Gilboa, G., Chan, T., and Osher, S. (2006). Structure-Texture Image Decomposition - Modelling, Algorithms, and Parameter Selection. *International Journal of Computer Vision*, 67(1):111–136.
- [4] Baillard, C. (2004). Production of DSM/DTM in Urban Areas: Role and Influence of 3D Vectors. In *ISPRS Congress*, volume 35, page 112 ff., Istanbul, Turkey.
- [5] Baillard, C. and Zisserman, A. (1999). Automatic Line Matching And 3D Reconstruction Of Buildings From Multiple Views. In *ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery*, volume 32, pages 69–80.
- [6] Bauer, J., Bischof, H., Klaus, A., and Karner, K. (2004). Robust and fully automated image registration using invariant features. In *The International Archives of the Photogrammetry, Remote Sensing and Spatial Information Services*, volume 35, pages 1119–1124.
- [7] Bignone, F., Henricsson, O., Fua, P., and Stricker, M. A. (1996). Automatic Extraction of Generic House Roofs from High Resolution Aerial Imagery. In *European Conference on Computer Vision*, pages 85–96, Berlin, Germany.
- [8] Blelloch, G. E. (1990). Prefix sums and their applications. Technical report, School of Computer Science, Carnegie Mellon University.
- [9] Bleyer, M., Gelautz, M., Rother, C., and Rhemann, C. (2009). A stereo approach that handles the matting problem via image warping. *Computer Vision and Pattern Recognition*.
- [10] Boykov, Y., Veksler, O., and Zabih, R. (1999). Fast Approximate Energy Minimization Via Graph Cuts. In *International Conference on Computer Vision*, volume 1, pages 377–384, Kerkyra, Corfu.

- [11] Bredies, K., Kunisch, K., and Pock, T. (2009). Total generalized variation.
- [12] Briggs, W. L., Henson, V. E., and McCormick, S. F. (2000). *A Multigrid Tutorial*. Society of Industrial and Applied Mathematics.
- [13] Burt, P. and Adelson, E. (1983). A multiresolution spline with application to image mosaics. *ACM Transactions on Graphics*, 2:217–236.
- [14] Chambolle, A. (2004). An algorithm for total variation minimizations and applications. *Journal of Mathematical Imaging and Vision*, 20(1-2):89–97.
- [15] Chambolle, A. (2005). Total variation minimization and a class of binary mrf models. In *Conference on Computer Vision and Pattern Recognition*.
- [16] Chambolle, A., Caselles, V., Novaga, M., Cremers, D., and Pock, T. (2009). An introduction to total variation for image analysis. Summer School on "Theoretical Foundations and Numerical Methods for Sparse Recovery".
- [17] Chan, T. and Esedoglu, S. (2004). Aspects of total variation regularized  $L^1$  function approximation. *SIAM J. Appl. Math.*, 65(5):1817–1837.
- [18] Chen, L.-C., Teo, T.-A., Shaoa, Y.-C., Lai, Y.-C., and Rau, J.-Y. (2004). Fusion of Lidar Data and Optical Imagery for Building Modeling. In *International Archives of Photogrammetry and Remote Sensing*, volume 35(B4), pages 732–737.
- [19] Chen, T., Yin, W., Zhou, X., Comaniciu, D., and Huang, T. (2006). Total Variation Models for Variable Lighting Face Recognition. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 28(9):1519–1524.
- [20] Collins, R.-T. (1995). A space-sweep approach to true multi-image matching. In *Conference on COmputer Vision and Pattern Recognition*, pages 358–363.
- [21] Curless, B. and Levoy, M. (1996). A Volumetric Method for Building Complex Models from Range Images. In *Proceedings of SIGGRAPH '96*, pages 303–312.
- [22] Esser, E., Zhang, X., and Chan, T. (2009). A general framework for a class of first order primal-dual algorithms for tv minimization. Technical report, UCLA, Center for Applied Mathematics.

- [23] Fischer, A., Kolbe, T., and Lang, F. (1999). Integration of 2D and 3D Reasoning for Building Reconstruction using a Generic Hierarchical Model. In *Workshop on Semantic Modeling for the Acquisition of Topographic Information*, pages 101–119, Munich, Germany.
- [24] Haala, N. and Anders, K.-H. (1996). Fusion of 2D-GIS and Image Data for 3D Building Reconstruction. In *International Archives of Photogrammetry and Remote Sensing*, volume 31, pages 289–290.
- [25] Haala, N. and Brenner, C. (1997). Generation of 3D City Models from Airborne Laser Scanning Data. In *3rd EARSEL Workshop on Lidar Remote Sensing on Land and Sea*, pages 105–112, Tallinn, Estonia.
- [26] Hirschmüller, H. (2005). Accurate and efficient stereo processing by semi-global matching and mutual information. In *Conference on Computer Vision and Pattern Recognition*, pages 328–341.
- [27] Hirschmüller, H. (2006). Stereo vision in structured environments by consistent semi-global matching. In *Conference on Computer Vision and Pattern Recognition*, pages 328–341.
- [28] Huang, C., Davis, L. S., and Townshend, J. R. G. (2002). An assessment of support vector machines for land cover classification. *International Journal of Remote Sensing*, 23:725–749.
- [29] Huber, P. (1981). *Robust Statistics*. Wiley, New York.
- [30] Hui, L. Y., John Trinder, and Kurt Kubik (2003). Automatic Building Extraction for 3D Terrain Reconstruction using Interpretation Techniques. In *ISPRS Workshop on High Resolution Mapping from Space*, page 9, Hannover, Germany.
- [31] Illingworth, J. and Kittler, J. (1988). A Survey of the Hough Transform. *Computer Vision, Graphics and Image Processing*, 44(1).
- [32] Keeling, S. L. (2003). Total variation based convex filters for medical imaging. *Applied Mathematics and Computation*, 139:101 – 119.
- [33] Klaus, A., Sormann, M., and Karner, K. (2006). Segment-Based Stereo Matching Using Belief Propagation and a Self-Adapting Dissimilarity Measure. In *Proceedings*

- of the 18th International Conference on Pattern Recognition, volume 3, pages 15–18, Washington, DC, USA. IEEE Computer Society.
- [34] Kolbe, T. H., Gröger, G., and Plümer, L. (2005). Citygml - interoperable access to 3d city models. In *First International Symposium on Geo-Information for Disaster Management*.
- [35] Kolmogorov, V. and Rother, C. (2007). Minimizing nonsubmodular functions with graph cuts - a review. *Pattern Analysis and Machine Intelligence*, 29:1274–1279.
- [36] Kolmogorov, V. and Zabih, R. (2001). Computing visual correspondence with occlusions using graph cuts.
- [37] Kolmogorov, V. and Zabih, R. (2002). What Energy Functions Can Be Minimized Via Graph Cuts? In *European Conference on Computer Vision*, volume 3, pages 65–81, Copenhagen, Denmark.
- [38] Leberl, F., Irschara, A., Pock, T., Meixner, P., Gruber, M., Scholz, S., and Wiechert, A. (2009). Point clouds from laser scanning versus 3d vision. *ASPRS-PE*, X:X.
- [39] Leonardis, A., Gupta, A., and Bajcsy, R. (1995). Segmentation of Range Images as the Search for Geometric Parametric Models. *International Journal of Computer Vision*, 14(3):253–277.
- [40] Lowe, D. G. (1999). Object recognition from local scale-invariant features. *International Conference on Computer Vision*, pages 1150–1157.
- [41] Maas, H.-G. and Vosselman, G. (1999). Two Algorithms for Extracting Building Models from Raw Laser Altimetry Data. In *ISPRS Journal of Photogrammetry and Remote Sensing*, volume 54, pages 153–163.
- [42] Mikolajczyk, K. and Schmid, C. (2003). A performance evaluation of local descriptors. In *Conference on Computer Vision and Pattern Recognition*.
- [43] Mumford, D. and Shah, J. (1989). Optimal approximations by piecewise smooth functions and associated variational problems. *Comm. Pure Appl. Math.*, 42:577–685.
- [44] Nemirovski, A. (2005). Prox-method with rate of convergence  $o(1/t)$  for variational inequalities with lipschitz continuous monotone operators and smooth convex-concave saddle point problems. *SIAM Journal on Optimization*, 15:229–251.

- [45] Nikolova, M. (2004). A variational approach to remove outliers and impulse noise. *Journal of Mathematical Imaging and Vision*, 20(1-2):99–120.
- [46] Nikolova, M., Esedoglu, S., and Chan, T. (2006). Algorithms for finding global minimizers of image segmentation and denoising models. *SIAM Journal of Applied Mathematics*, 66(5):1632–1648.
- [47] Nistér, D. (2003). An efficient solution to the five-point relative pose problem. In *Conference on Computer Vision and Pattern Recognition*, pages 195–202.
- [48] Oda, K., Lu, W., Uchida, O., and Doihara, T. (2004). Triangle-based visibility analysis and true ortho generation. *International Archives of Photogrammetry, Remote Sensing and Spatial Information Sciences*, 35:623–629.
- [49] Olsson, C., Byröd, M., Overgaard, N., and Kahl, F. (2009). Extending continuous cuts: Anisotropic metrics and expansion moves. In *International Conference on Computer Vision*.
- [50] Perez, P., Gangnet, M., and Blake, A. (2003). Poisson image editing. In *SIGGraph*.
- [51] Pock, T., Cremers, D., Bischof, H., and Chambolle, A. (2009). An algorithm for minimizing the mumford-shah functional. *Proceedings of the International Conference on Computer Vision*.
- [52] Pock, T., Schoenemann, T., Graber, G., Bischof, H., and Cremers, D. (2008). Convex formulation of continuous multi-label problems. In *European Conference on Computer Vision*.
- [53] Pock, T., Zach, C., and Bischof, H. (2007). Mumford-shah meets stereo: Integration of weak depth hypotheses. In *Proceedings of IEEE conference on Computer Vision and Pattern Recognition*, pages 1–8.
- [54] Pottmann, H., Leopoldseder, S., and Hofer, M. (2002). Simultaneous Registration of Multiple Views of a 3D Object. In *Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, volume 34, Part 3A.
- [55] Pottmann, H., Leopoldseder, S., and Hofer, M. (2004). Registration without ICP. volume 95, pages 54–71.
- [56] Rockafellar, R. T. (1997). *Convex Analysis*. Princeton Landmarks in Mathematics.

- [57] Rudin, L., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Physica D*, 60:259–268.
- [58] Scharstein, D. and Szeliski, R. (2002). A Taxonomy and Evaluation of Dense Two-Frame Stereo Correspondence Algorithms. In *International Journal of Computer Vision*, volume 47, pages 7–42.
- [59] Schenk, T. (1997). Towards automatic aerial triangulation. *ISPRS Journal of Photogrammetry and Remote Sensing*, 52:110–121.
- [60] Schmid, C. and Zisserman, A. (1997). Automatic Line Matching Across Views. In *IEEE Conference on Computer Vision and Pattern Recognition*, pages 666–671.
- [61] Sinha, S. N., Steedly, D., and Szeliski, R. (2009). Piecewise planar stereo for image-based rendering. In *International Conference of Computer Vision*.
- [62] Snavely, N., Seitz, S. M., and Szeliski, R. (2007). Modeling the world from internet photo collections. *International Journal of Computer Vision*, 80:189 – 210.
- [63] Sohn, G. and Dowman, I. (2007). Data Fusion of High-Resolution Satellite Imagery and LIDAR Data for Automatic Building Extraction. *ISPRS Journal of Photogrammetry and Remote Sensing*, 62(1):43–63.
- [64] Strang, G. (1983). Maximal flow through a domain. *Mathematical Programming*, 26:123–143.
- [65] Suveg, I. and Vosselman, G. (2004). Reconstruction of 3D Building Models from Aerial Images and Maps. *ISPRS Journal of Photogrammetry and Remote Sensing*, 58(3-4):202–224.
- [66] Szeliski, R., Uyttendaele, M., , and Steedly, D. (2008). Fast poisson blending using multi-splines. Technical report, Microsoft Research.
- [67] Taillardier, F. and Deriche, R. (2004). Automatic Buildings Reconstruction from Aerial Images: a Generic Bayesian Framework. In *Proceedings of the XXth ISPRS Congress, Istanbul, Turkey*.
- [68] Tell, D. and Carlsson, S. (2002). Combining appearance and topology for wide baseline matching. In *7th European Conference on Computer Vision*, pages 68–81.



- [69] Tikhonov, A. N. (1943). On the stability of inverse problems. *Dokl. Akad. Nauk SSSR*, 5:195–198.
- [70] Unger, M., Pock, T., Grabner, M., Klaus, A., and Bischof, H. (2009). A variational approach to semiautomatic generation of digital terrain models. In *5th Int. Symp. on Visual Computing*.
- [71] Vineet, V. and Narayanan, P. (2008). Cudacuts: Fast graph cuts on the gpu. In *Computer Vision and Pattern Recognition Workshops*.
- [72] Vogel, C. R. and Oman, M. E. (1996). Iterative methods for total variation denoising. *SIAM J. Sci. Comput.*, 17:227–238.
- [73] Vosselman, G. (1999). Building Reconstruction Using Planar Faces in Very High Density Height Data. In *ISPRS Conference on Automatic Extraction of GIS Objects from Digital Imagery*, volume 32, pages 87–92, Munich.
- [74] Wang, L., Liao, M., Gong, M., Yang, R., and Nister, D. (2006). High-quality real-time stereo using adaptive cost aggregation and dynamic programming. *3D Data Processing, Visualization, and Transmission*, pages 798 – 805.
- [75] Werlberger, M., Trobin, W., Pock, T., Wedel, A., Cremers, D., and Bischof, H. (2009). Anisotropic huber-l1 optical flow. In *British Machine Vision Conference*.
- [76] Zach, C., Pock, T., and Bischof, H. (2007). A globally optimal algorithm for robust TV- $L^1$  range image integration. In *Proceedings of the 11th International Conference Computer Vision*, pages 1–8, Rio de Janeiro, Brazil.
- [77] Zebedin, L., Klaus, A., Gruber-Geymayer, B., and Karner, K. (2006). Towards 3D Map Generation from Digital Aerial Images. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60(6):413–427.
- [78] Zhang, L. and Gruen, A. (2006). Multi-image matching for dsm generation from ikonos imagery. *ISPRS Journal of Photogrammetry and Remote Sensing*, 60:195–211.

