Graz University of Technology

Institute for Computer Graphics and Vision

Master's Thesis

Robust Multi-View Reconstruction from Highly Redundant Aerial Images

---

Markus Rumpler, BSc

Graz, Austria, November 2010

Thesis supervisor

Prof. Dr. Horst Bischof

Thesis advisor

Dipl.-Ing. Arnold Irschara

Deutsche Fassung:
Beschluss der Curricula-Kommission für Bachelor-, Master- und Diplomstudien vom 10.11.2008
Genehmigung des Senates am 1.12.2008

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.

Graz, am ……. ……..
(Unterschrift)

Englische Fassung:

STATUTORY 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.

……. ……..
date (signature)

Abstract

This thesis investigates and presents robust multi-view matching methods to produce dense depth maps from highly redundant imagery. We gather theory from camera models and two-view geometry to methods for solving the correspondence problem. We investigate in several experiments the influence of different cost functions and cost aggregation schemes on the results of multi-view depth matching in a plane sweep framework. The evaluation includes local and global optimization methods.

The main contribution of this thesis is an extension of the highly efficient TV-L^1 optical flow algorithm that includes the epipolar constraint. While correspondence computation is still performed between pairs of images, we present a method for correspondence linking between nearby views. This enables the use of measurements from all neighboring views used for matching and provides wider baselines for robust and accurate triangulation.

We provide evaluation results of the proposed method and present its performance in contrast to a standard plane sweep approach. The benefits include less computation time and memory costs, continuous results instead of discrete depth estimates and comparable but in most cases even better accuracy. It requires no or just little user guidance, thus our design is capable for integration into a fully automatic reconstruction pipeline.

Keywords. Computer Vision, Stereo Correspondence Problem, Matching Costs, Global Optimization, Optical Flow, Epipolar Constraint, Multi-View Stereo, Dense Matching, Robust 3D Reconstruction

Kurzfassung

Diese Arbeit untersucht und präsentiert robuste Mehrbild-Verfahren, um aus redundanten Bilddaten über Punktkorrespondenzen dichte Tiefenkarten zu erstellen. Wir erarbeiten die Theorie beginnend bei Kameramodellen und Zweibild-Geometrie bis hin zu Methoden zur Lösung des Korrespondenzproblems. Dabei untersuchen wir in mehreren Experimenten den Einfluss verschiedener Kostenfunktionen und Modelle zur Kombination von Kosten auf die Ergebnisse eines Mehrbild-Rekonstruktionsverfahrens auf Basis eines Plane-Sweep-Ansatzes.

Der Hauptbeitrag dieser Arbeit besteht in der Erweiterung des hocheffizienten TV-L^1 optischen Fluss-Algorithmus zur Berücksichtigung der Epipolargeometrie. Während die Korrespondenzberechnung weiterhin zwischen Bildpaaren erfolgt, wird eine Methode zur Verlinkung von Korrespondenzen zwischen benachbarten Bildern präsentiert. Dies erlaubt die Einbeziehung von korrespondierenden Bildpunkten aus weiter entfernten Nachbarbildern und ermöglicht robuste und genaue Triangulation.

Wir zeigen Evaluierungsergebnisse für die vorgestellte Methode und präsentieren die Resultate im Vergleich zu einem Standard-Plane-Sweep-Ansatz. Die Vorteile liegen vor allem in einer Reduzierung von Berechnungsaufwand und Speicherbedarf, kontinuierlichen Tiefen im Gegensatz zu diskreten Werten und einer vergleichbaren und in den meisten Fällen bessern Genauigkeit. Die Methode benötigt keine beziehungsweise nur wenig Benutzerführung, weshalb unser Design zur Integration innerhalb einer vollautomatischen Rekonstruktionspipeline geeignet ist.

Schlagwörter. Computer Vision, Stereo Korrespondenzproblem, Kostenfunktionen, Globale Optimierung, Optischer Fluss, Epipolargeometrie, Mehrbild-Verfahren, Dense Matching, Robuste 3D Rekonstruktion

Acknowledgments

This thesis provides me the opportunity not only to document and present my work of the last year, but also to thank several people who have contributed to their success. First of all, I would like to thank Prof. Horst Bischof for giving me the opportunity to work on this interesting topic and for supervising my thesis. Special thanks go to Arnold Irschara for his guidance, prolific suggestions and support during the whole project and Thomas Pock for providing me helping implementational advice.

I dedicate this thesis to my parents who unremittingly supported me during my years of study. I really appreciate for supporting me in everything I did.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Contents

1.1 Outline and Motivation

Image-based 3D reconstruction is an active field of research in Photogrammetry and Computer Vision. The need for detailed 3D models for mapping and navigation, inspection, cultural heritage conservation or photorealistic image-based rendering for the entertainment industry lead to the development of several techniques to recover the shape of objects. Common systems are based on active-sensor methods or active and passive 3D vision.

Active-sensor methods usually based on *RAdio Detection And Ranging* (Radar) or *LIght Detection And Ranging* (Lidar) systems are able to provide directly 2.5D range images and 3D point clouds.

To achieve precision and high detail reconstructions, Lidar systems are successfully employed. Laser-based methods on the other hand, are very complex for large scale outdoor scenes, especially when aerial data acquisition with airborne sensors is required.

Active stereo systems are able to determine 3D coordinates under controlled conditions accurately and in real-time, but are not suited for large scale outdoor environments.

Passive image-based methods utilize multiple overlapping views to determine geometry. Those methods are more robust, low-cost and flexible for reconstructing large scenes and are capable to provide comparable accuracy to Lidar systems [25].

Figure 1.1: **A Reconstructed Scene:** The scene has been reconstructed from five Microsoft VEXCEL UltraCam Images from the Jakomini dataset. The images have a geometrical resolution of 7500x11500 pixels with approximately eight centimeter ground resolution per pixel.

This thesis is motivated by the goal of competing and replacing these methods with image based approaches using high resolution aerial imagery to achieve savings in cost, effort and time for acquiring automatic high accuracy large scale 3D reconstructions. But new problems and challenges can arise, in particular due to temporary varying events during data acquisition, e.g. vehicles, pedestrians and/or changing light conditions.

The objective of this thesis is to present fast, accurate and robust multi-view matching techniques suitable for high resolution images of large scale scenes. Starting from terrestrial or aerial imagery, we compute dense depth maps with (semi) automatic *passive image-based* methods as an intermediate step to a full Euclidean 3D reconstructed model. Reconstruction from multiple views hereby contributes to the completeness of the scene, aids in the correspondence problem and improves depth accuracy by increasing triangulation angles.

Motivated by related work on image-based modeling [11, 13, 14, 16, 19, 43, 51] we are going to take a look at two different approaches for multi-view reconstruction, keeping in

mind the surveys of Scharstein et al. [33], Seitz et al. [34] and Strecha et al. [37] who evaluated the performance of several algorithms.

The following dense matching approaches assume already known camera calibration. Intrinsic parameters and external orientations of the camera's positions are provided. Additionally, a sparse scene reconstruction obtained by Structure-from-Motion (SfM) [17, 22] exists that is used for initialization.

The first technique described is a plane-sweep approach that traverses 3D space by parallel planes. Local matching costs measure similarity between a key view and multiple neighboring images projected onto these planes in varying depths. The correct depth is assumed to be the one with the lowest cost value or the highest similarity score respectively. Correlation scores are used to fill a 3D cost volume. Additional optimization techniques are needed to extract high quality depth maps since this method will always be prone to errors.

One global approach is to define depth as a multi-labeling problem (each label corresponds to a discrete depth) that can be then solved exactly using a variational approach [30]. However such methods are in general very time and memory intensive.

To overcome the time consuming optimization step required to extract depth maps in the first approach, we propose a method that estimates pixel correspondences using TV-L^1 optical flow [50]. Disparities between pixels are estimated within a global optimization framework that seeks a solution by minimizing an appropriate energy function. This approach is extended to follow the epipolar constraint, hence restricts the correspondence search to a one-dimensional problem.

The successive subsections of this introductory chapter describe the challenges to cope with in 3D reconstruction and recall the basics of camera models and two-view geometry, followed by a brief overview of different scene representations. The chapter closes with a classification of dense stereo algorithms. Chapter 2 examines how to find pixel correspondences (i.e. matching) in pairs of images using local correlation measures and total variation based optical flow minimizing a global energy function. 3D reconstruction extended to multiple views, occlusion handling and the requirement for initialization is discussed in Chapter 3. We present details on the plane sweep and optical flow based reconstruction methods in Chapter 4, followed by experimental results and qualitative

and quantitative evaluation in Chapter 5. The thesis concludes with Chapter 6 presenting a summary of covered considerations, a discussion based on our observations and results and displays future prospects.

1.2 3D Reconstruction

Recovering scene structure from images is one major topic in the field of Computer Vision. As we use two or more images as input, we talk about *image-based* (multi-view) stereo reconstruction. Image-based 3D reconstruction usually incorporates and solves three main problems [8]:

1. **Camera calibration**,
2. **point correspondence** and
3. **reconstruction**.

Digital images are captured by a sensor within a camera measuring the radiance reflected (or emitted) by the object's surface. This strategy allows that 3D structure can be inferred passively without interfering the reconstructed object.

Extracting 3D information from 2D image representations is far from being an easy task. The object's appearance in an image can change completely with viewpoint. Formally, there is no unique solution to such an ill-posed problem [7].

The main challenges of 3D reconstruction are [36]:

Perspective Projection. As the human eye does, a camera projects all scene points along a ray from the camera center onto a single point in the 2D image plane. A single image does not provide enough information to recover depth.

Brightness Constancy. Depending on surface attributes and orientation, type and position of light sources and viewpoint, image intensity values of corresponding scene points might vary.

Occlusions. Visibility of scene points of non-convex objects or points affected by mutual occlusion changes with viewpoint, making the search for uniquely corresponding image points impossible.

Noise. Due to the physics of image formation, input data is corrupted by noise.

To overcome all these difficulties two or more images and robust methods incorporating assumptions about the physical world are necessary to achieve reliable and accurate results.

1.3 Camera Model and Two-View Geometry

1.3.1 Camera Model

A camera projects 3D scene points onto a 2D image plane. In general, a camera models central projection with its camera center being finite. The mapping of 3D world points and pixel coordinates can be described by employing homogeneous coordinates and projective geometry. This allows to represent the camera as a matrix incorporating all its geometric entities and attributes. A 3x4 matrix P, the *projection matrix*, usually has 11 degrees of freedom (DOF) to describe internal and external parameters of the camera.

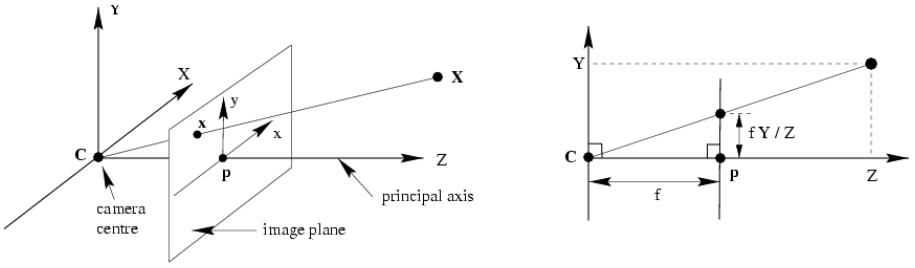

Figure 1.2: **The Pinhole Camera Model:** A scene point X projects onto the image plane to a point x on the line connecting the projection center C and X. The optical axis goes through the image plane at the principal point p. The point X maps to the point x on the image plane by similar triangles (adopted from [17]).

We start to develop our camera model with the simplest one, the pinhole camera. We assume that 3D scene points project onto a plane through the projection center lying at the origin of a Euclidean coordinate system. The *image plane* or *focal plane* is at $Z = f$ and perpendicular to the *principal* or *optical axis* with f equal to the focal length of the camera. The central projection mapping from world to image coordinates of a scene point $X = (X, Y, Z)^\top$ to the point $x = (fX/Z, fY/Z, f)^\top$ on the image plane is computed by similar triangles. The pinhole camera geometry is shown in Figure 1.2.

We can express the linear mapping in terms of a matrix multiplication by representing the coordinates of both points by homogeneous vectors:

$$\begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \longmapsto \begin{pmatrix} fX \\ fY \\ Z \end{pmatrix} = \begin{pmatrix} f & & & 0 \\ & f & & 0 \\ & & 1 & 0 \end{pmatrix} \begin{pmatrix} X \\ Y \\ Z \\ 1 \end{pmatrix} \tag{1.1}$$

Equation (1.1) can be described in a compact manner as

$$x = PX \tag{1.2}$$

The 3x4 homogeneous *camera projection matrix* P can be written as

$$P = K[I|0] \tag{1.3}$$

K is the *camera calibration matrix* and $[I|0]$ the identity matrix with an appended column vector with zeros. The camera calibration matrix is of the form K = *diag*(*f*, *f*, 1).

Taking the principal point offset into account, then K becomes:

$$K = \begin{pmatrix} f & & p_x \\ & f & p_y \\ & & 1 \end{pmatrix} \tag{1.4}$$

The camera model derived so far assumes square pixels with equal scales in both dimensions. To derive the general form of a camera calibration matrix, we multiply K on the left with $diag(m_x, m_y, 1)$. The parameters m_x and m_y represent the number of pixels per unit distance in the x and y direction respectively. An additional *skew* parameter s (which usually equals zero) modeling non-rectangular pixels gives 5 DOFs for the intrinsic camera parameters in K.

$$K = \begin{pmatrix} \alpha_x & s & x_0 \\ & \alpha_y & y_0 \\ & & 1 \end{pmatrix} \tag{1.5}$$

The camera is still located at the origin of our Euclidean coordinate system, the current *camera coordinate frame*. Scene point coordinates make use of a *world coordinate frame* which, in general, is different from our camera coordinate system. It relates to the world coordinates via a rotation and a translation (Figure 1.3).

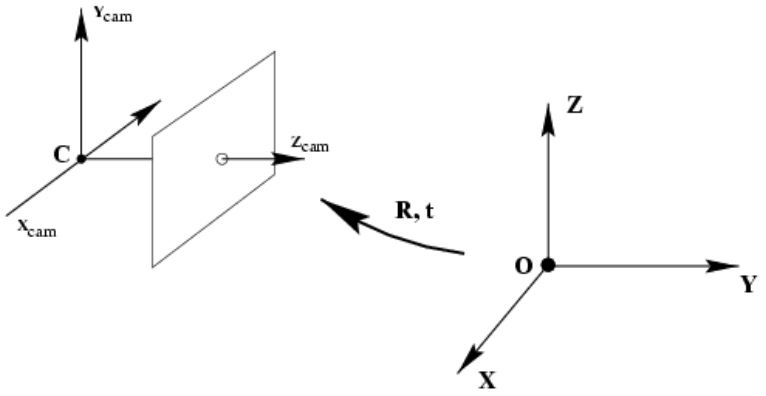

Figure 1.3: **Camera Rotation and Translation:** The *camera coordinate frame* is related to the *world coordinate system* via rotation and translation (adopted from [17]).

A point $\tilde{X}$ in world coordinates is transformed into the camera coordinate system with $\tilde{X}_{cam} = R(\tilde{X} - \tilde{C})$. The camera center in world coordinates is denoted as $\tilde{C}$ and R is a 3x3 rotation matrix representing the external orientation of the camera coordinate frame.

Together with (1.3), P results in

$$P = KR[I| - \tilde{C}] \tag{1.6}$$

and with $t = -R\tilde{C}$

$$P = K[R|t] \tag{1.7}$$

Rotation and translation add three DOFs each, together they form the set of extrinsic parameters representing the external orientation of the camera. Hence, a finite projective camera has a total of 11 DOFs, equal to the number of degrees of freedom of a 3x4 matrix, defined up to an arbitrary scale [17].

The inverse of Expression (1.2) describes the back-projection of image points to rays. It writes to:

$$X = P^{+}x \tag{1.8}$$

$$X(\lambda) = P^{+}x + \lambda C \tag{1.9}$$

with the pseudo-inverse $P^+ = P^\top(PP^\top)^{-1}$ of the matrix P for which $PP^+ = I$ is valid. Two points on the ray in Equation (1.9) are known: P^+x and the camera center C. Every point on that ray projects onto the same image coordinates [4, 17].

1.3.2 Epipolar Geometry

In this section we introduce the geometric relation of two views. Given two images with associated projection matrices P for the first view and P' for the second, capturing the same scene from different viewpoints, then there exists a geometric relationship between these views depending solely on the relative poses and internal parameters of the cameras.

A 3D scene point projects to $x = PX$ in the first view and to $x' = P'X$ in the second. The Epipolar geometry (see Figure 1.4) describes the relationship between two views constraining possible positions for the corresponding image points x and x'. This relationship is expressed through a 3x3 matrix, named the *fundamental matrix* F.

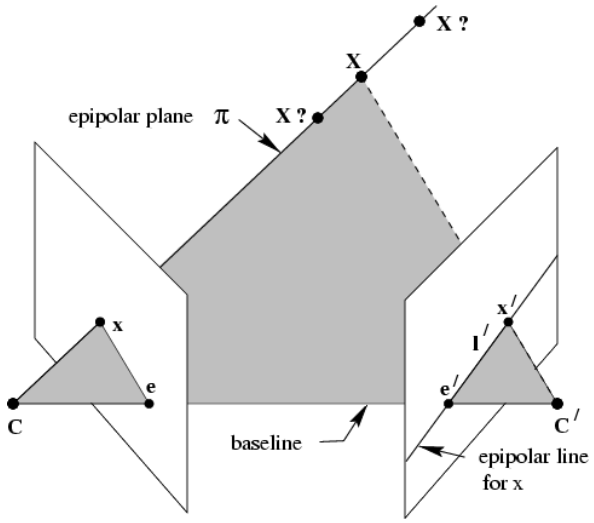

Figure 1.4: **Epipolar Geometry:** The corresponding points x and x' are projections of the same 3D point in space. Point x in the first view defines an *epipolar line* l' in the second image. The imaged point of X in the second view must lie on that line (adopted and modified from [17]).

The ray through C and x together with the *baseline* connecting the projection centers C and C' of the two cameras span the *epipolar plane* π. The *epipolar line* l' locates where the plane π meets the image plane of the second view. Equivalently, the epipolar line is the back-projected ray in 3D-space defined by the camera center C of the first view and x, projected into second image.

$$x \mapsto l' \tag{1.10}$$

$$l' = Fx \qquad l = F^\top x' \tag{1.11}$$

The imaged point x' of X in the second view must locate on the line l'. If we know only x, this yields that the search for correspondences in stereo matching algorithms can be reduced to a 1D search problem along the epipolar line, no need for covering the entire image plane.

The point where the baseline intersects the image plane is called *epipole* and corresponds to the imaged camera center of the second camera. It is the intersection point of all epipolar lines in that image.

For corresponding image points x and x' the fundamental matrix F satisfies the relation

$$x'^\top F x = 0 \tag{1.12}$$

The fundamental matrix F can be derived either from the camera projection matrices P and P' or computed via known point correspondences in the images. However, F is independent from scene structure. The following algebraic derivation of the fundamental matrix is one of various ways and uses the former approach.

Again we back-project a pixel by $X = P^+x$ and get a ray, as shown in Equation (1.9). Then, we use two points on that ray imaged by the second camera. In particular, we project the camera center C of the first camera, which gives the epipole in the second image with $e' = P'C$, and a second point on the ray (P^+x at $\lambda = 0$), that projects to $P'P^+x$. The line connecting these two points is the epipolar line

$$l' = (P'C) \times (P'P^+x) = [e']_\times (P'P^+)x = Fx \tag{1.13}$$

with the fundamental matrix

$$F = [P'C]_\times P'P^+ = [e']_\times P'P^+ \tag{1.14}$$

In (1.13) and (1.14) the notation $[\cdot]_\times$ is used to represent the cross product with the epipole as a matrix multiplication.

For a 3-vector $a = (a_1, a_2, a_3)^\top$ it is defined as

$$[a]_\times = \begin{pmatrix} 0 & -a_3 & a_2 \\ a_3 & 0 & -a_1 \\ -a_2 & a_1 & 0 \end{pmatrix} \tag{1.15}$$

To form the fundamental matrix based on the cameras extrinsic parameters, we transform both cameras in order that the camera center of first camera becomes the world origin with $R = I$, $t = 0$ and with $R' = R$, $t' = t$ for the second camera by applying $R = R'R^\top$ and $t = -R'R^\top t + t'$. The projection matrices for the two cameras now write to

$$P = K[I|0] \qquad P' = K'[R|t] \tag{1.16}$$

With

$$P^+ = \begin{pmatrix} K^{-1} \\ 0^\top \end{pmatrix} \qquad C = \begin{pmatrix} 0 \\ 1 \end{pmatrix} \tag{1.17}$$

applied to Equation (1.14), the fundamental matrix becomes

$$F = [P'C]_\times P'P^+ = [K't]_\times K'RK^{-1} = K'^{-\top}[t]_\times RK^{-1} \tag{1.18}$$

From R and t we are able to derive the epipoles $e = -KR^\top t$ and $e' = K't$ directly [4, 17]

1.3.3 Stereo Triangulation

Given corresponding image points x and x' and the calibration matrices of the cameras, we are now able to reconstruct the 3D position of the imaged scene point X to estimate its depth. We suppose the left cameras coordinate system is the world origin with the optical axis in z-direction, the right camera's optical axis is parallel to the first one. Translation of the second camera is only along the x-axis as shown in Figure 1.5.

The focal length of both cameras is f and the distance along the baseline between them is denoted as T. The depth in z-direction of the point X in this canonical configuration is now deduced by applying elementary geometry.

$$\frac{Z}{T} = \frac{Z - f}{T - x - x'} \tag{1.19}$$

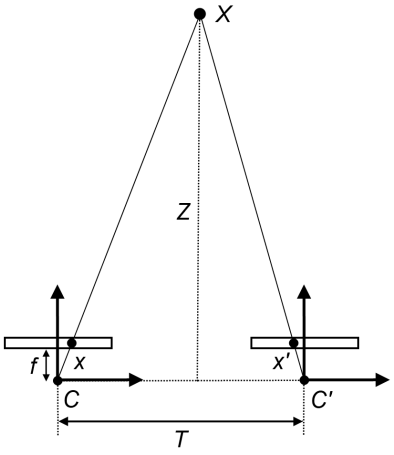

Figure 1.5: **Stereo Triangulation:** Recovering depth in a simple canonical stereo configuration. The depth calculates to $Z = f\frac{T}{x+x'}$ using similar triangles CC'X and xx'X.

$$Z = f\frac{T}{x + x'} = f\frac{T}{d} \tag{1.20}$$

In this basic stereo triangulation for a camera setup in canonical or standard configuration, the disparity $d = x+x'$ measures the difference in the position between corresponding points. The depth is inversely proportional to disparity.

Recovering depth for non-canonical setups with arbitrary rotations between cameras is done by finding the intersection point of two rays in 3D space. The rays are given through back-projecting the image measurements x and x'. For corresponding image points, the epipolar constraint $x'^{\top}Fx = 0$ is fulfilled and both rays lie in an epipolar plane and so intersect in the point X.

But in general, the epipolar constraint is not satisfied and naive triangulation will fail. Due to errors in the measured image points x and x', their back-projected rays will not intersect, as shown in Figure 1.6. It is necessary to estimate an algebraically best solution related to the reprojection error for the point of X in 3D space using linear triangulation or Maximum Likelihood Estimates (MLE) for the true image point correspondences. Here, only linear triangulation is discussed because it provides acceptable results in most real world problems and easily generalizes to more than two views.

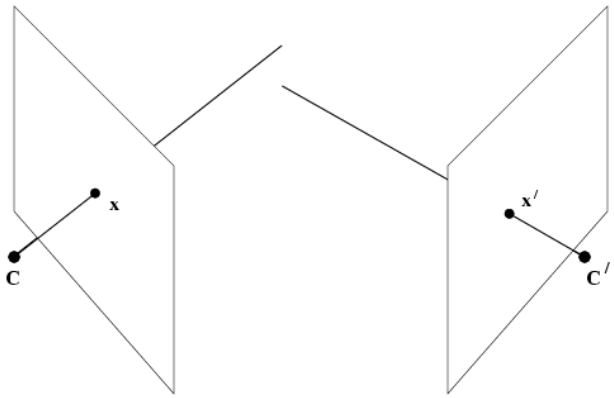

Figure 1.6: **Ray Back-Projection:** Back-projected rays from imperfectly measured image points x and x' that do not satisfy the epipolar constraint will not intersect.

The measurements $x = PX$ and $x' = P'X$ can be combined into an equation of the form $AX = 0$ linear in X. For each image point, we get three equations of which two are linearly independent by a cross product eliminating the homogeneous scale factor. For the first image $x \times (PX) = 0$, this is gives

$$\begin{aligned} x(p^{3\top}X) - (p^{1\top}X) &= 0 \\ y(p^{3\top}X) - (p^{2\top}X) &= 0 \\ x(p^{2\top}X) - y(p^{1\top}X) &= 0 \end{aligned} \tag{1.21}$$

The rows of the projection matrix P are denoted with $p^{i\top}$. Using two equations from each image results in four equations in four homogeneous unknowns. We can write

$$A = \begin{pmatrix} xp^{3\top} - p^{1\top} \\ yp^{3\top} - p^{2\top} \\ x'p'^{3\top} - p'^{1\top} \\ y'p'^{3\top} - p'^{2\top} \end{pmatrix} \tag{1.22}$$

The system of linear equations of the form $AX = 0$ can be solved either by a homogeneous or inhomogeneous method. The Homogeneous method solves by finding the smallest singular value of A and its corresponding unit singular vector.

With X in homogeneous coordinates, $AX = 0$ is reduced to a set of four inhomogeneous equations in tree unknowns. The solution of this inhomogeneous method is achieved by

using a least-squares approximation [17].

1.4 Classification of Dense Stereo Algorithms

We focus on algorithms that produce a *dense* disparity or depth map (i.e. a disparity or depth value is assigned to each pixel) and operate on images with known camera geometry. The emphasis on dense output is motivated by the requirement for depth estimates in all image regions including texture-less or occluded areas. The disparity or depth map is usually a univalued function $d(x, y)$ encoding the output of the stereo correspondence method with respect to a reference or key view.

Alternative approaches for scene geometry representation used in related work include *voxel-based* [43] surface approximations in a regularly sampled 3D grid or volume, *level-set* methods [12] for encoding the distance to the closest surface as a function and *polygonal meshes* [52] that represent a surface as a collection of vertices and (triangulated) planar polygonal faces. In contrast to these output representations, the depth map [23] representation eliminates the need for resampling the geometry in the three-dimensional domain [33, 34].

Dense stereo correspondence algorithms can be divided into two categories. The first class of dense stereo methods are *local* (window-based) algorithms, which use information from each pixel or its local neighborhood making implicit smoothness assumptions. This allows for determining a pixel's disparity or depth independently, typically by analyzing the intensity values within a finite (rectangular) window around the pixel of interest and comparing two windows by statistical means. The result of this comparison is a cost value or similarity score that measures the similarity between the two windows. It is usually derived based on correlation.

Secondly, algorithms in the category of *global* methods make explicit smoothness assumptions. Algorithms in this class typically find a solution by minimizing an energy function. The choice of an appropriate energy function, from which the most contain a data term and an additional smoothness term is an important aspect. A solution for the minimization of the energy function is in general achieved by an iterative optimization technique. Commonly used minimization procedures in global algorithms are simulated annealing, probabilistic (mean-field) diffusion or graph-cuts [33, 38].

Chapter 2

The Two-View Correspondence Problem

Contents

2.1 Local Stereo Matching

A core component of every stereo correspondence algorithm is a method to measure similarity between image locations. The similarity is expressed by a *matching cost* value, usually scaled between [0 ... 1]. At each pixel, the matching cost is computed over all disparities within a certain disparity range under consideration. Depending on the method that calculates the matching cost, we try to find the disparity which either maximizes a similarity measure or minimizes an error score [4, 20].

A general notation of a cost function comparing intensity values of the images I_1 and I_2 at an image location $x = (x, y)^\top$ is given by:

$$C = f_C(I_1(x), I_2(x + d)) \tag{2.1}$$

Equation (2.1) defines a mapping that takes intensity values from an image pair and outputs a cost value for a certain disparity vector d. For example, with rectified image pairs where epipolar lines are horizontally aligned, the search for corresponding image

locations can be restricted to a path along the x-direction (with the y-coordinate fixed) and d reduces to $d = (d_x, 0)^\top$.

Cost functions examined here are defined on intensity (luminance) values, but can easily be extended to color. Therefore, costs for all color channels are separately computed and then combined accordingly.

We can distinguish between *pixel-based* and *window-based* cost functions. Whereas simple pixel-based cost functions rely only on values I_1 and I_2 from a single image location x and assume constant intensities (*brightness constancy assumption*), window-based matching costs take more intensity values within a finite neighborhood around the pixel of interest into account.

Typically, the neighborhood is defined by a rectangular $k \times k$ window w, with the window size k as an odd integer number. A rectangular window around a pixel of interest can be defined by a single parameter r, denoting the radius of the window. The radius is linked with the window size k through the relation $k = 2r + 1$.

$$C(I_1, I_2, x, y, d_x, d_y) = \sum_{u=-r}^{r} \sum_{v=-r}^{r} f_C(I_1(x+u, y+v), I_2(x+d_x+u, y+d_y+v)) \quad (2.2)$$

A shortened notation for Equation (2.2) for all pixels w in a certain neighborhood around a pixel $x = (x, y)^\top$ (for the disparity we again use the notation $d = (d_x, d_y)^\top$) is given as

$$C(I_1, I_2, x, d) = \sum_{w} f_C(I_1(x), I_2(x+d)) \quad (2.3)$$

For better readability we omit pixel coordinates and the disparity in Equation (2.3) and write

$$C(I_1, I_2) = \sum_{w} f_C(I_1, I_2) \quad (2.4)$$

While rectangular windows are commonly used, arbitrary shapes of windows aggregating support are possible.

Due to varying illumination conditions and changing appearance of image points according to viewpoint changes, in general, a similarity measures with normalization is required.

More robust (normalized) matching costs are able to compensate for noise and certain radiometric differences (e.g. additive or multiplicative brightness variations) [4, 18, 20].

In general, any stereo correspondence algorithm makes implicit or explicit assumptions, necessary to model the physical world and the image formation process. *Matching assumptions* include, for example, that image points are projections of the same scene point. Often object surfaces are assumed to hold Lambertian properties, i.e. that their appearance does not change with viewpoint (diffuse reflection, no specular highlights). Other methods try to embed assumptions about radiometric differences appearing as image intensity changes in gain and bias or model certain forms of noise of the camera sensor.

The correspondence problem would be ill-posed and underconstrained without additional assumptions about visual appearance of scene objects or the world and scene geometry, e.g. that the physical world consists of piecewise smooth surface patches. These assumption can be summarized under the term *smoothness assumptions* [33].

Matching costs evaluated pixel-based on pixel intensities include *absolute differences* (AD), *squared differences* (SD) or the sampling insensitive dissimilarity measure (BT) proposed by Birchfield and Tomasi [6].

As window-based matching costs, the *sum of absolute differences* (SAD) or *squared differences* (SSD) and cross correlation (CC) are commonly used. To reduce mismatches due to radiometric differences between images, in practical implementations at least cross correlation is mainly used only in its *normalized* variant (NCC) accounting for multiplicative changes (gain), denoted by the prefix *N*. Additive differences are compensated by the *zero-mean* versions of SAD, SSD or NCC to address constant intensity offsets (bias), denoted by the prefix *Z*, resulting in ZSAD, ZSSD and ZNCC.

Insensitivity against radiometric differences can be achieved by filtering the images in a preprocessing step, for example mean filtering, computing the first derivative producing a gradient magnitude image or the Laplacian of Gaussian (LoG). These filters applied in stereo matching algorithms have the disadvantage to produce blurred depth images.

To avoid blurring high contrast texture differences that may correspond to depth discontinuities, Ansar et al. proposed *background subtraction by bilateral filtering* (BilSub) in [1]. Bilateral filtering works by summing neighboring pixel values weighted depending on their spatial (proximity) and radiometric (color similarity) distance to the center pixel. This technique is able to effectively remove a local offset by smoothing without blurring

high contrast texture.

Cost functions mentioned so far can be classified under the term *parametric* matching cost functions [20, 33].

Other measures insensitive to differences in gain or bias belong to the group of *non-parametric* matching cost functions. Non-parametric measures like *Rank* and *Census*, introduced for being robust against outliers at object boundaries, were proposed in [48]. Since they rely solely on the relative ordering of intensities and not on the intensity values itself, they are also invariant to radiometric variations that preserve the original order of the pixels [20, 33].

The disadvantage of the methods mentioned so far is their problematic behavior when matching images over wide baselines, due to perspective distortions, occluded areas and, in general, regions with uniform texture. As an alternative, *feature-based* methods attempt to overcome these problems by computing reliable descriptors. Local region descriptors have been designed to be robust to perspective distortions and changes in illumination. Traditionally, descriptors like SIFT or GLOH are computationally demanding and therefore used only for sparse matching.

A local region descriptor named DAISY was introduced by Tola et al. able to be computed quickly at every pixel, however, retaining the robustness of SIFT and GLOH. DAISY descriptors may then be matched by calculating the Euclidean distance between the two feature vectors [39].

2.1.1 Parametric Cost Functions

2.1.1.1 Absolute Differences

A simple but commonly used error measure is the *absolute difference* (AD). It is defined as the absolute difference of intensity values of two pixels:

$$C_{AD}(I_1, I_2) = |I_1 - I_2| \tag{2.5}$$

The window-based version is called *sum of absolute differences* (SAD) summing up all differences within the neighborhood w:

$$C_{SAD}(I_1, I_2) = \sum_{w} |I_1 - I_2| \tag{2.6}$$

Window-based SAD allows to compensate for radiometric differences in bias and gain through a normalization strategy. For additive and multiplicative intensity differences, there exists an approach to achieve a certain degree of invariance to variations in pixel brightness. Both approaches can be combined.

An additive offset (bias) is compensated by *zero-mean* normalization. We therefore assume that the pixel intensities vary in a constant additive value t:

$$I_1 + t = I_2 \tag{2.7}$$

To compensate for additive intensity variations, we need to reduce the mean intensity value within a window w to zero. This is achieved by subtracting it from each original pixel intensity within the window:

$$C_{ZSAD}(I_1, I_2) = \sum_w |(I_1 - \bar{I}_1) - (I_2 - \bar{I}_2)| \tag{2.8}$$

with

$$\bar{I} = \frac{1}{N_w} \sum_w I \tag{2.9}$$

For the normalized images $I_1' = I_1 - \bar{I}_1 = I_2 - \bar{I}_2 = I_2'$ holds true:

$$I_2' = I_2 - \bar{I}_2 = I_2 - \frac{1}{N_w} \sum_w I_2 = I_1 + t - \frac{1}{N_w} \sum_w (I_1 + t) = I_1 - \frac{1}{N_w} \sum_w I_1 = I_1 - \bar{I}_1 = I_1' \tag{2.10}$$

To compensate for multiplicative changes (gain), we assume that pixels of both images I_1 and I_2 differ only in a constant factor r. For all pixels applies:

$$r \cdot I_1 = I_2 \tag{2.11}$$

To normalize the image data, we need to scale the original pixel intensity values with its Frobeniusnorm (2.12) within the observed window.

$$\|I\|_F = \sqrt{\sum_w I^2} \tag{2.12}$$

Both images normalized for multiplicative changes calculate to:

$$I_1'' = \frac{I_1}{\|I_1\|_F} \qquad I_2'' = \frac{I_2}{\|I_2\|_F} \tag{2.13}$$

It is easy to prove that $I_1'' = I_2''$ using Equation (2.11):

$$I_2'' = \frac{I_2}{\|I_2\|_F} = \frac{I_2}{\sqrt{\sum_w I_2^2}} = \frac{rI_1}{\sqrt{\sum_w (rI_1)^2}} = \frac{I_1}{\sqrt{\sum_w I_1^2}} = \frac{I_1}{\|I_1\|_F} = I_1'' \tag{2.14}$$

The resulting cost value is called *normalized* and is denoted by the prefix *N*.

$$C_{NSAD}(I_1, I_2) = \sum_w \left| \frac{I_1}{\|I_1\|_F} - \frac{I_2}{\|I_2\|_F} \right| = \sum_w \left| \frac{I_1}{\sqrt{\sum_w I_1^2}} - \frac{I_2}{\sqrt{\sum_w I_2^2}} \right| \tag{2.15}$$

Both normalization approaches combined result in a cost function that is insensitive to additive and multiplicative changes. The Frobeniusnorm calculates now from the zero-mean image data.

$$\|I'\|_F = \sqrt{\sum_w (I - \bar{I})^2} \tag{2.16}$$

$$C_{ZNSAD}(I_1, I_2) = \sum_w \left| \frac{I_1 - \bar{I}_1}{\|I_1'\|_F} - \frac{I_2 - \bar{I}_2}{\|I_2'\|_F} \right| = \sum_w \left| \frac{I_1 - \bar{I}_1}{\sqrt{\sum_w (I_1 - \bar{I}_1)^2}} - \frac{I_2 - \bar{I}_2}{\sqrt{\sum_w (I_2 - \bar{I}_2)^2}} \right| \tag{2.17}$$

What we get is a *zero-mean normalized* cost function value that has normal distribution with zero mean and unit variance, denoted by the prefix *ZN*.

In the same way, this normalization approach is applied to the parametric cost functions SSD and cross correlation described in the following subsections.

2.1.1.2 Squared Differences

The next classical cost function under consideration is the *squared difference* (SD). It calculates a reliable error score for like intensity values, but is more sensitive to outliers, accounting to the square of the intensity differences.

$$C_{SD}(I_1, I_2) = (I_1 - I_2)^2 \tag{2.18}$$

$$C_{SSD}(I_1, I_2) = \sum_w (I_1 - I_2)^2 \tag{2.19}$$

The *sum of squared differences* (SSD) from Equation (2.19) calculates the cost value over a window w which is again suitable for normalization.

For the normalized variants the same steps from above apply to SSD analogously, resulting in

$$C_{ZSSD}(I_1, I_2) = \sum_w ((I_1 - \bar{I}_1) - (I_2 - \bar{I}_2))^2, \tag{2.20}$$

$$C_{NSSD}(I_1, I_2) = \sum_w \left(\frac{I_1}{\|I_1\|_F} - \frac{I_2}{\|I_2\|_F} \right)^2 = \sum_w \left(\frac{I_1}{\sqrt{\sum_w I_1^2}} - \frac{I_2}{\sqrt{\sum_w I_2^2}} \right)^2 \tag{2.21}$$

and finally

$$C_{ZNSSD}(I_1, I_2) = \sum_w \left(\frac{I_1 - \bar{I}_1}{\|I_1'\|_F} - \frac{I_2 - \bar{I}_2}{\|I_2'\|_F} \right)^2 = \sum_w \left(\frac{I_1 - \bar{I}_1}{\sqrt{\sum_w (I_1 - \bar{I}_1)^2}} - \frac{I_2 - \bar{I}_2}{\sqrt{\sum_w (I_2 - \bar{I}_2)^2}} \right)^2 . \tag{2.22}$$

2.1.1.3 Cross Correlation

Unlike both previously discussed cost functions, *cross correlation* (CC) defines a similarity measure. The more similar the intensity values are, the higher is the similarity score.

$$C_{CC}(I_1, I_2) = \sum_w I_1 \cdot I_2 \tag{2.23}$$

The expressions for the normalized variants are:

$$C_{ZCC}(I_1, I_2) = \sum_w (I_1 - \bar{I}_1) \cdot (I_2 - \bar{I}_2) \tag{2.24}$$

$$C_{NCC}(I_1, I_2) = \sum_w \frac{I_1}{\|I_1\|_F} \cdot \frac{I_2}{\|I_2\|_F} = \frac{\sum_w I_1 \cdot I_2}{\sqrt{\sum_w I_1^2 \cdot \sum_w I_2^2}} \tag{2.25}$$

$$C_{ZNCC}(I_1, I_2) = \sum_w \frac{I_1 - \bar{I}_1}{\|I_1'\|_F} \cdot \frac{I_2 - \bar{I}_2}{\|I_2'\|_F} = \frac{\sum_w (I_1 - \bar{I}_1) \cdot (I_2 - \bar{I}_2)}{\sqrt{\sum_w (I_1 - \bar{I}_1)^2 \cdot \sum_w (I_2 - \bar{I}_2)^2}} \tag{2.26}$$

In contrast to the previous cost functions, we notice that the Frobeniusnorm is not part of the summation here, which is a benefit when taking runtime under consideration.

When dealing with Gaussian noise, normalized cross correlation is statistically the optimal correlation method. On the contrary, due to high errors in presence of outliers it tends to blur depth discontinuities more than other cost functions [20].

2.1.2 Non-Parametric Cost Functions

A different approach for solving the correspondence problem was introduced in [48] by Zabih and Woodfill by applying *non-parametric* local transforms to the images before matching.

Non-parametric local transforms rely solely on the local ordering of intensity values rather than on the intensities themselves. The transforms only depend on the sign of the comparison between the center pixel and the pixel intensities in its neighborhood and are therefore invariant under radiometric distortions that preserve this ordering. Those measures can reduce sensitivity to outliers arising from radiometric gain and bias or noise significantly [8]. The limited error-proneness and enhanced tolerance against outliers can improve the resulting performance near depth discontinuities at the boundaries of objects.

Some of these cost functions can be implemented as filters that change the input images individually. Then, matching is performed using correlation.

Pixels within a local region near object boundaries picture distinct parts of the scene and represent scene elements from two different intensity populations. The intensity distribution within such a local region is in general multimodal and poses a severe problem for many statistical correlation measures.

Correspondence measures based on statistical methods such as normalized cross correlation are suited best for unimodal intensity distributions. This issue referred as *faction-*

alism arising in many computer vision tasks has been addressed with methods like robust statistics, Markov Random Fields and regularization [48].

We are now going to take a look at two non-parametric measures: the *rank transform* and the *census transform*.

2.1.2.1 Rank

The *rank transform* measures intensities within a local region w and is defined as the number of pixels whose intensity is less than the intensity of the center pixel x. It replaces the pixel's intensity with its rank among all neighboring pixels p in that local region [8, 20, 48].

$$I_{Rank}(x) = \sum_{w} T(I(p) < I(x)) \tag{2.27}$$

The function $T(\cdot)$ returns 1 if its argument evaluates true, and 0 otherwise. An example is given in Figure 2.1.

$$\begin{matrix} 95 & 97 & 105 \\ 90 & 100 & 107 \\ 97 & 105 & 110 \end{matrix} \quad \Longrightarrow \quad \begin{matrix} 1 & 1 & 0 \\ 1 & & 0 \\ 1 & 0 & 0 \end{matrix} \quad \Longrightarrow \quad 4$$

Figure 2.1: **The Rank Transform:** The rank transform replaces the intensity of a pixel by the number of pixels within a local region for which the intensity values are less than that of the center pixel.

Noise can be a problem for rank filtering in textureless image regions. To address this shortcoming, *soft rank*, a variation of the rank transform defines a linear, soft transition zone between 0 and 1 which produces clearly less noisy results:

$$I_{SoftRank}(x) = \sum_{w} min\left(1, max\left(0, \frac{I(x) - I(p)}{2t} + \frac{1}{2}\right)\right), \tag{2.28}$$

with t acting as a threshold.

Matching of the rank filtered images is then performed using the L^1 norm (i.e. with the absolute difference (AD)).

The rank transform compresses the magnitudes of the intensities into one single value, which is the reason for the reduced sensitivity to outliers and improved tolerance for factionalism. The influence of a minority of neighboring pixels with very different intensity values is limited. Such pixels only contribute proportional to their number, not proportional to their intensity, as this would be the case with parametric measures.

While this is an advantage in presence of radiometric variations in the image data, it also reduces the discriminatory power as the ordering information of the pixels is lost during the transform.

2.1.2.2 Census Transform

Census stores not only the intensity ordering of the surrounding pixels, but preserves also the spatial structure of the intensity values within the local neighborhood by encoding it in a bitstring. Each bit set to 1 corresponds to a certain pixel p whose intensity is lower than that of the center pixel x [8, 20, 48].

$$I_{Census}(x) = BITSTRING_w[\quad T(I(p) < I(x))\quad] \tag{2.29}$$

with $T(\cdot)$ returning 1 if its argument evaluates true, and 0 otherwise.

Through this transform, the dimensionality of the resulting census filtered image is increased by a factor of the size of the local neighborhood used for deriving the bitstring.

$$\begin{array}{ccc} 95 & 97 & 105 \\ 90 & 100 & 107 \\ 97 & 105 & 110 \end{array} \quad \Longrightarrow \quad \begin{array}{ccc} 1 & 1 & 0 \\ 1 & & 0 \\ 1 & 0 & 0 \end{array} \quad \Longrightarrow \quad [11010100]$$

Figure 2.2: **The Census Transform:** The census transform defines a bitstring in some canonical ordering for a pixel of interest, where each bit corresponds to a certain neighboring pixel. A bit for a corresponding pixel is set, if its intensity is lower than that of the center pixel.

Similar to the rank transform, the influence of a minority of pixels with very different intensity values within the neighborhood is restricted. The effects of a minority of pixels with very different intensity values is limited by the size of the minority.

Furthermore, mean and median variants of the census transform try to reduce the influence of the center pixel's intensity value. As the name suggests, the *mean census* variant uses the mean intensity within the local neighborhood in the comparison step. The *median census transform* calculates the, in terms of computation time more costly median to generate the bitstring.

For correspondence computation, the two bitstrings are matched using the *Hamming distance* (i.e. the number of bits that differ) between them. The Hamming distance is then minimized after applying the census transform. Comparing for similarity using the Hamming distance can be performed very efficiently and confirms a trend of moving from Euclidean to Hamming distance for matching purposes [9].

Table 2.1 summarizes all local correspondence methods and matching metrics with their formulas discussed so far.

2.1.3 Cost Window Aggregation

Stereo algorithms typically incorporate four steps: matching cost computation, cost aggregation, disparity/depth estimation and refinement/optimization [33]. So far we have discussed the cost computation step, now we focus on aggregating costs on a *variable support*.

Most recent research on advances in computational stereo concentrated on robust matching in the presence of radiometric distortions and noise, occlusion detection and real-time methods. Besides, several interesting and effective approaches for cost aggregation on local methods have been developed during the last couple of years. These methods provide results that promise to yield comparable accuracy to that of many global algorithms [41].

Different cost aggregation schemes were implemented using fixed squared windows symmetrically around a center pixel of *varying window size* [41], *shiftable windows* [23] anchored at variable positions within the window or *(adaptive) weights* [2, 47], where every single pixel's contribution to the support region is defined by (adaptive) weights allowing any unconstrained, arbitrary shape [33].

Local Correspondence Method		Definition
Absolute Difference	AD	$C_{AD}(I_1, I_2) = \lvert I_1 - I_2 \rvert$
	SAD	$C_{SAD}(I_1, I_2) = \sum_w \lvert I_1 - I_2 \rvert$
	ZSAD	$C_{ZSAD}(I_1, I_2) = \sum_w \lvert (I_1 - \bar{I}_1) - (I_2 - \bar{I}_2) \rvert$
	NSAD	$C_{NSAD}(I_1, I_2) = \sum_w \left\lvert \frac{I_1}{\sqrt{\sum_w I_1^2}} - \frac{I_2}{\sqrt{\sum_w I_2^2}} \right\rvert$
	ZNSAD	$C_{ZNSAD}(I_1, I_2) = \sum_w \left\lvert \frac{I_1 - \bar{I}_1}{\sqrt{\sum_w (I_1 - \bar{I}_1)^2}} - \frac{I_2 - \bar{I}_2}{\sqrt{\sum_w (I_2 - \bar{I}_2)^2}} \right\rvert$
Squared Difference	SD	$C_{SD}(I_1, I_2) = (I_1 - I_2)^2$
	SSD	$C_{SSD}(I_1, I_2) = \sum_w (I_1 - I_2)^2$
	ZSSD	$C_{ZSSD}(I_1, I_2) = \sum_w ((I_1 - \bar{I}_1) - (I_2 - \bar{I}_2))^2$
	NSSD	$C_{NSSD}(I_1, I_2) = \sum_w \left(\frac{I_1}{\sqrt{\sum_w I_1^2}} - \frac{I_2}{\sqrt{\sum_w I_2^2}} \right)^2$
	ZNSSD	$C_{ZNSSD}(I_1, I_2) = \sum_w \left(\frac{I_1 - \bar{I}_1}{\sqrt{\sum_w (I_1 - \bar{I}_1)^2}} - \frac{I_2 - \bar{I}_2}{\sqrt{\sum_w (I_2 - \bar{I}_2)^2}} \right)^2$
Cross Correlation	CC	$C_{CC}(I_1, I_2) = \sum_w I_1 \cdot I_2$
	ZCC	$C_{ZCC}(I_1, I_2) = \sum_w (I_1 - \bar{I}_1) \cdot (I_2 - \bar{I}_2)$
	NCC	$C_{NCC}(I_1, I_2) = \frac{\sum_w I_1 \cdot I_2}{\sqrt{\sum_w I_1^2 \cdot \sum_w I_2^2}}$
	ZNCC	$C_{ZNCC}(I_1, I_2) = \frac{\sum_w (I_1 - \bar{I}_1) \cdot (I_2 - \bar{I}_2)}{\sqrt{\sum_w (I_1 - \bar{I}_1)^2 \cdot \sum_w (I_2 - \bar{I}_2)^2}}$
Rank Transform		$I_{Rank}(x) = \sum_w T(I(p) < I(x))$
		Matching is performed using the absolute difference.
Census		$I_{Census}(x) = BITSTRING_w[\quad T(I(p) < I(x)) \quad]$
		Matching is performed using the Hamming distance.
Feature Matching		Distinctive image features are matched rather than intensities.

Table 2.1: **Local Matching Metrics:** A Summary of local matching metrics and their definitions.

Using variable support for cost aggregation aims at higher accuracy at depth discontinuities and an overall lower matching ambiguity in textureless image regions. The concept behind is to find the best set of pixels (i.e. the *support*) for computing the matching cost at a potential correspondence. In contrast to *fixed static support* like squared windows or a single pixel, these methods vary and adapt itself depending on each correspondence's local characteristics.

It is important to state, that the criterion on determining the support window size, shift offset and weights at each correspondence depends on both images, due to the fact that it is typically obtained from the cost function itself. And that is by nature based on values of both images. The best support between a set of windows of different size and/or displacement is chosen based on minimization of e.g. the variance of the cost function serving as a local confidence (or reliability) value [41].

2.1.3.1 Rectangular Windows for Cost Aggregation

Choosing an appropriate window size for cost aggregation that fits multiple scenarios is a difficult problem. While small windows allow good localization of the minimum cost along the search path and better handling near depth discontinuities and object borders, they do not capture enough information in low textured image areas yielding to a high matching ambiguity. Large windows instead lead to boundary overreach at depth discontinuities, when pixels belonging to different depths are aggregated within the support window. This causes blurred edged in the results and comprises the possibility of missing fine image details.

See Figure 2.3 for typical correlation results for different window sizes. We assume the correct disparity or depth to locate at the overall minimum score along the search path. This is a typical *winner-takes-all* strategy used in real-time applications and for evaluation. Using the example of an SAD score, we can see that different sizes of correlation windows lead to different results.

The first method for cost aggregation with variable support is performed using rectangular windows of *varying window size*. The size is adapted and follows the local characteristics of each pixel correspondence under evaluation. Local constraints for selecting a support region include that for example large windows are favored in low-textured image regions whereas small windows are used near depth edges [2].

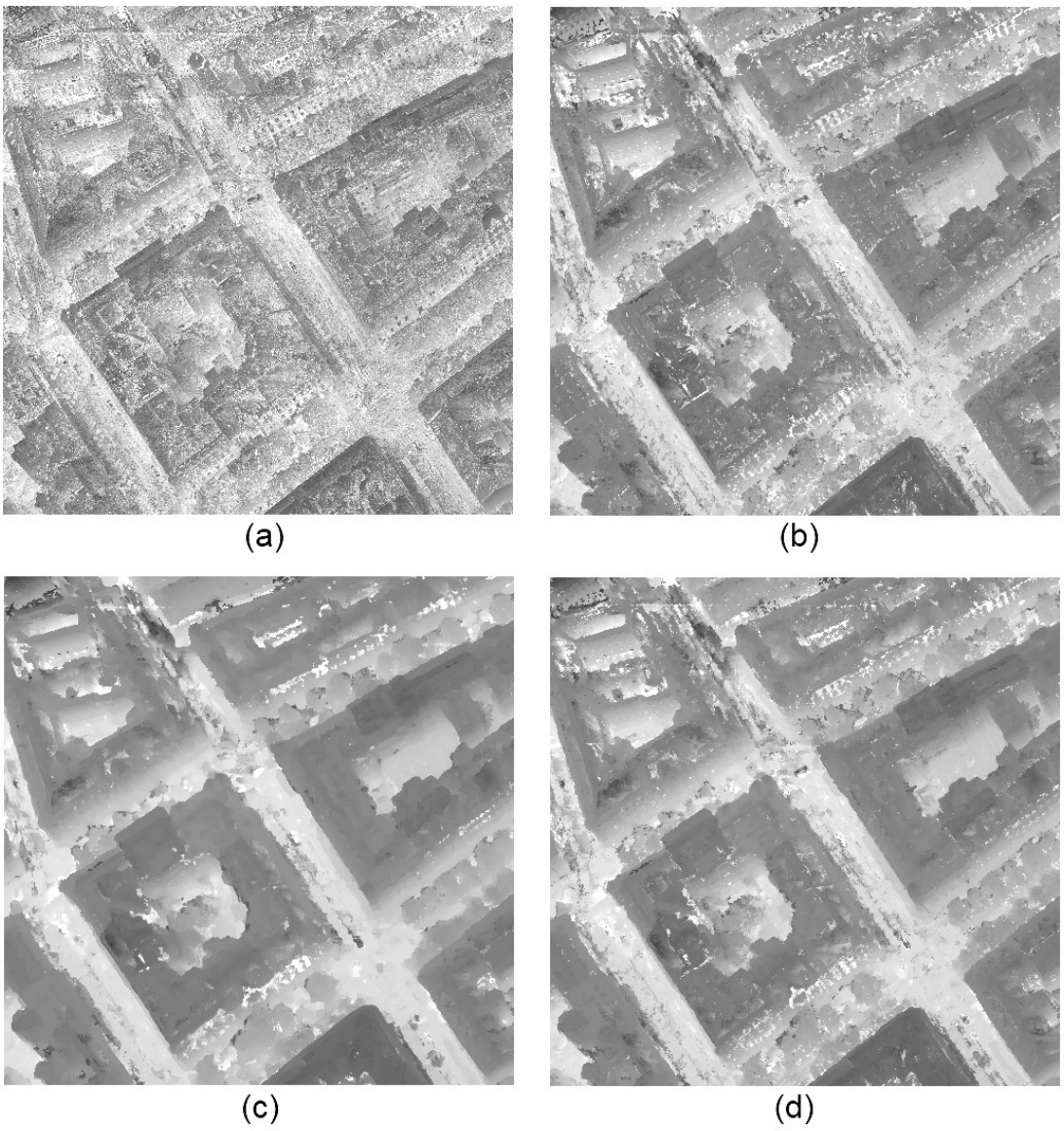

Figure 2.3: **The Influence of Window Size:** The four figures above showcase typical results for different correlation window sizes on the Jakomini aerial image sequence.
(a) Matching with the pixel-wise absolute difference lead to noisy results with lots of outliers, since it does not gather enough information in low-textured image regions. Increasing the window radius to r=1 (b) and r=3 (c) reduces outliers but tends to smooth edges. Large windows reduce the possibility of a mismatch, but yield reduced accuracy near depth discontinuities. (d) The image illustrates the effect of a multi-resolution kernel combining three correlation windows. This allows increased robustness to outliers through a large support region and though good localization (i.e. less smoothing).

The resulting matching cost is then computed from the obtained support region with fixed weights assigned to each pixel within the region. Besides just minimizing the cost, thus improves robustness, accuracy and provides better results than support windows of fixed size.

When aggregating support near object boundaries, the matching reliability suffers from erroneous cost computation introduced by image points within the window belonging to multiple objects with different depths. Instead of using just one window symmetrically centered around the pixel we are trying to match, several spatially shifted windows that include the pixel of interest are investigated. *Shiftable windows* try to find an appropriate window in order to aggregate image points that lie on the same depth plane. The support window is not necessarily anchored around the pixel anymore, for which we search a corresponding match. This approach can improve matching results near object boundaries and depth discontinuities, generally speaking, image regions that can suffer from occlusion effects [23, 41].

Moreover, hierarchical solutions try to select and combine multiple windows representing the best support instead of a single window [41]. Yang et al. [46] describes a *multi-resolution* approach for cost aggregation. In Figure 2.3 we have already seen the influence of window size on matching scores. Figure 2.4 shows an example of a multi-resolution kernel for cost aggregation. Cost values of each level are either just summed/averaged or only the cost value of the window yielding the minimum error is used. This approach is well suited for computation on graphics hardware.

Figure 2.4: **A Multi-Resolution Cost Aggregation Kernel:** The shape of the multi-resolution kernel [46] puts larger weights to pixels at the center allowing good localization while increasing robustness through a large support region.

Whereas the multi-resolution approach uses a fixed number of overlapping windows, a variable support strategy deploys the selection and union of just a subset of all tested (partially) overlapping support regions, referred to as *best supporting windows*. Any combination of varying window size and shiftable windows is possible.

The approach of combining the global characteristics of large support regions and well-localized small windows corresponds to the use of a large window and assigning stronger weights to the center pixels. The shape of the resulting support region is then no longer constrained to a rectangular window. This also leads to a cost aggregation strategy that allows explicit assignments of different *weights* for the points of the support region [41, 46].

Since all these approaches of cost aggregation rely on rectangular windows, a considerable speedup in efficiency can be achieved by exploiting incremental computation schemes [41].

2.1.3.2 Support Regions of Unconstrained Shapes

Generalizations of using a set of rectangular windows allow support regions of *unconstrained shapes* and the use of *adaptive weights* to better adapt to local image characteristics. Image points within a certain neighborhood around a pixel of interest are classified as associated or not-associated to the support region as a result of a photometric relation or segmentation process beforehand. The best supporting region for cost aggregation is then selected as the largest set of connected image points within that region. The approach is based on the idea that connected components of related intensities belong to the same object, thus having similar depths. Segmentation information relies either on only one image or is obtained symmetrically from both [15, 41].

Different and variable weights can be assigned to the image points within the support region, e.g. according to the variance of the error function. Weights based on spatial proximity and radiometric distance in color space and weighted by means of a Gaussian function are also possible. There is now a continuous transition if, and how strong image points contribute to the support region.

A disadvantage of cost aggregation based on unconstrained shapes and adaptive weights is that they do not always leads to computationally efficient algorithms [41].

2.2 Global Stereo Methods

2.2.1 Global Optimization

Stereo algorithms in this class exploit global support to search for correspondence in order to increase robustness and accuracy, where local methods would otherwise result in incorrect matches and fail due to a lack of texture or occlusions.

While local methods emphasis on the cost computation and aggregation steps and

e.g. extract final disparities by simply choosing the depth associated with the lowest cost value (winner-takes-all) at each pixel, global approaches perform all of their work during disparity computation.

Global methods rely on minimizing a global energy function using an iterative optimization scheme. Typically, such an energy function consists of two terms, a *data term* and a *smoothness term* and a parameter λ that weights between them and determines the degree of smoothness. The influence on the results of TV-L^1 stereo for different values of λ is examined in Figure 2.5. Large values for λ lead to stronger smoothing of fine details.

$$E_{global}(d(x,y)) = E_{data}(d(x,y)) + \lambda E_{smooth}(d(x,y)) \tag{2.30}$$

The expression $d(x,y)$ represents the disparity field. The objective and desired solution is chosen as the value of the disparity function d that minimizes the global energy E_{global}.

How well the disparity function d matches with the input images is measured with the data term E_{data}.

$$E_{data}(d(x,y)) = \sum_{(x,y)} C(x,y,d(x,y)), \tag{2.31}$$

where C can be just the aggregated or initial unaggregated matching cost.

When using a smoothness term, spatial aggregation of the cost values, i.e. using a window based method is usually not necessary. Hence, the cost function often reduces to a pixel-based measure.

The smoothness term E_{smooth} integrates the smoothness assumptions made by the algorithm and measures the piecewise smoothness in the disparity field. Often, a restriction to only measure the disparity differences between neighboring pixels allows easier, manageable computation.

$$E_{smooth}(d(x,y)) = \sum_{(x,y)} s^h_{x,y}\rho(d(x,y) - d(x+1,y)) + s^v_{x,y}\rho(d(x,y) - d(x,y+1)) \tag{2.32}$$

The smoothness function or potential $\rho(\cdot)$ is some monotonically increasing function. Typically, ρ is a quadratic, a truncated quadratic or a delta function in regularization-based vision. Simple quadratic functions smooth d everywhere and may result in erroneous

matches at depth discontinuities and object borders. Other more robust functions of the disparity differences do not have this problem, which are then referred to as *discontinuity preserving* energy functions.

The parameters $s^h_{x,y}$ and $s^v_{x,y}$ weight the smoothness strengths and can vary spatially.

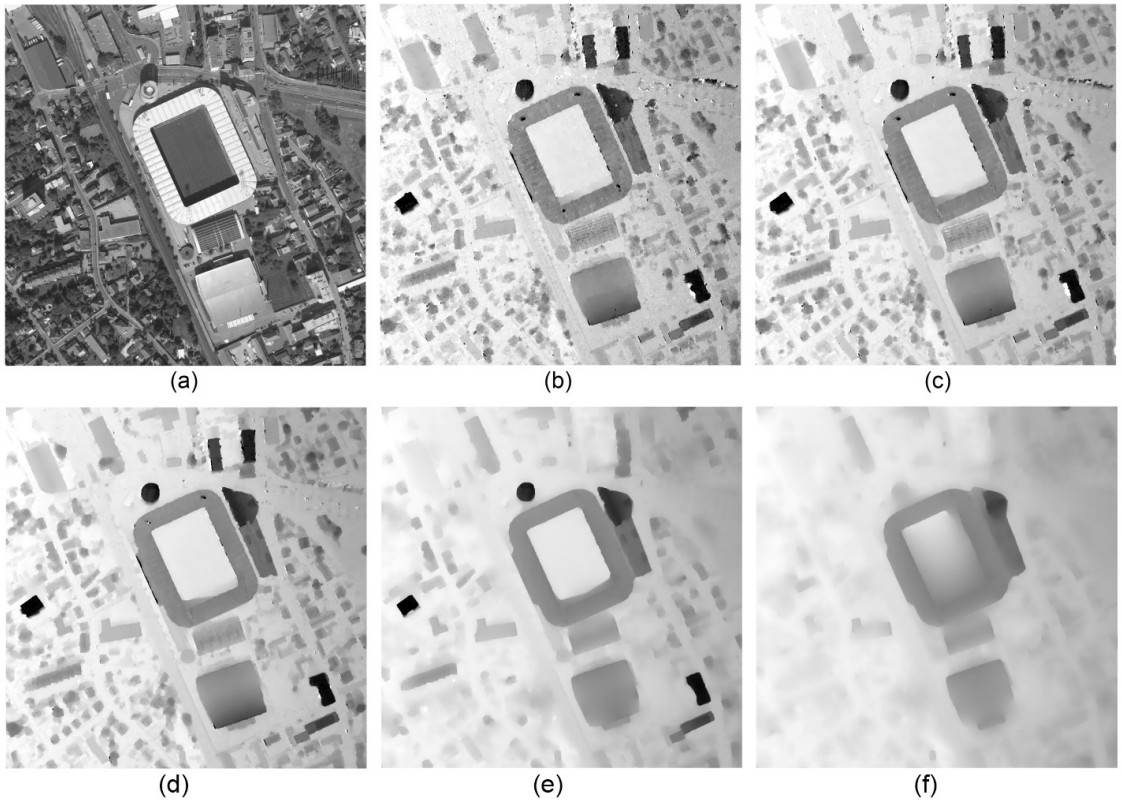

Figure 2.5: **Lambda and the Degree of Smoothness:** The parameter λ determines the degree of smoothness. (a) Key view; TV-L^1 stereo results for (b) $\lambda = 0.01$ (c) $\lambda = 0.02$ (d) $\lambda = 0.04$ (e) $\lambda = 0.16$ (f) $\lambda = 0.32$

Additionally, the terms in E_{smooth} can also depend on intensity differences to link disparity discontinuities with intensity changes, e.g. at intensity edges related to object boundaries.

This accounts furthermore for some of the good performance of global algorithms. In general, global methods are able to produce results that are superior to those of local methods.

The main difference between global optimization approaches arises in the method that is used to find the minimum, once the global energy function is defined. Commonly used

approaches associated with regularization and Markov Random Fields (MRF) include simulated annealing, mean-field methods and graph cuts [23, 33].

Solving the two-dimensional optimization problem from Equation (2.30) using common classes of smoothness function is an NP-hard problem. An approach based on *dynamic programming* reduces the computational complexity by decomposing the optimization task into smaller sub-problems. Each one-dimensional energy function can then be solved independently along each scanline in polynomial time [8, 23, 33, 38].

2.2.2 Pixel Correspondences through Optical Flow

2.2.2.1 Determining Optical Flow

Optical flow seeks to determine displacement fields between two images estimating the motion of pixels. Hence, optical flow is equivalent to the search for correspondences is stereo vision.

Horn and Schunck [21] formulate the problem as a differential equation that relates the change of image brightness at a point to motion of the brightness pattern. Therefore, it is assumed that the intensity value of a particular point is constant between two views. This constraint is called the *brightness constancy assumption*:

$$\frac{dI}{dt} = 0 \tag{2.33}$$

We define $I(x, y, t)$ as the image brightness at a point (x, y) and at time t. Here the temporal parameter t is understood as the sequence between left and right stereo image.

Applying the chain rule for differentiation yields

$$\frac{\partial I}{\partial x}\frac{dx}{dt} + \frac{\partial I}{\partial y}\frac{dy}{dt} + \frac{\partial I}{\partial t} = 0 \tag{2.34}$$

With

$$u_1 = \frac{dx}{dt} \quad \text{and} \quad u_2 = \frac{dy}{dt}, \tag{2.35}$$

we get a single linear equation in two unknowns u_1 and u_2:

$$I_x u_1 + I_y u_2 + I_t = 0 \tag{2.36}$$

The abbreviations I_x, I_y and I_t represent the partial derivatives of image intensity in x, y and t.

Recovering u_1 and u_2, depicting the displacement or flow respectively solely based on intensity constraints will in general result in an under-determined system of equations. Determining optical flow is a highly ill-posed inverse problem, i.e. there is no unique solution. This is known as the *aperture problem*.

Some kind of regularization (a prior) is needed to avoid the aperture problem and to get physically meaningful results. We assume that objects of finite size undergo rigid motion, this means that neighboring points have similar displacements and the flow field varies smoothly. Therefore, Horn and Schunck use a quadratic smoothness term to penalize deviations of the displacement field.

$$\left(\frac{\partial u_1}{\partial x}\right)^2 + \left(\frac{\partial u_1}{\partial y}\right)^2 \quad \text{and} \quad \left(\frac{\partial u_2}{\partial x}\right)^2 + \left(\frac{\partial u_2}{\partial y}\right)^2 \tag{2.37}$$

Equation (2.37) expresses the additional smoothness constraint as the square of the magnitude of the flow gradient [8, 21, 50].

Horn and Schunck formulate the optical flow as a variational problem.

$$\min_u \left\{ \int_\Omega |\nabla u_1|^2 + |\nabla u_2|^2 d\Omega + \lambda \int_\Omega \left(I_1(x + u(x)) - I_0(x)\right)^2 d\Omega \right\} \tag{2.38}$$

Variational methods are motivated by statistical inference methods and aim to minimize an energy *functional*. A functional maps an input function to an output value. Variational methods are able to successfully solve a number of different computer vision problems.

One decisive advantage of variational methods and the reason for its increasing popularity during the last years is its behavior at locations where no image information is available. In those regions, the flow functions u_1 and u_2 have hardly any influence on the contribution of the data term. As a consequence and to fulfill at least the smoothness constraint, the flow functions adapt to the local solutions and neighborhood information is propagated to image locations where such information is missing. This so-called *filling-in effect* is the reason why variational methods can compute a solution everywhere and always obtain dense results.

In Expression (2.38) denote I_0 and I_1 the image pair, the two-dimensional displacement field is represented by the vector $u = (u_1(x), u_2(x))^\top$. The first term is the regularization (or smoothness) term that penalizes deviations in the flow field u. Secondly, the data term penalizes variations from the brightness constancy constraint, that assumes that pixel intensities of $I_0(x)$ and $I_1(x + u(x))$ do not change.

A major drawback of the Horn and Schunck optical flow method is, that they use a quadratic measure to penalize deviations of the flow field. This yields strong smoothing along depth borders and does not allow for discontinuities. Another limitation is the use of a data term that does not handle outliers robustly.

Approaches with more robust error norms and higher order data terms have been proposed to overcome its limitations. Commonly used techniques use a first order Taylor approximation to linearize the nonlinear intensity profile of $I_1(x + u(x))$ locally. To allow for determining large displacements, *coarse-to-fine* strategies (i.e. *scale-space* approaches or *image pyramids*) are used, since the approximation is only valid for small disparities [29, 35, 50].

2.2.2.2 TV-L^1 Optical Flow

A highly efficient algorithm for optical flow computation was introduced in [29] and [50]. It is based on *total variation* (TV) regularization and uses a robust L^1 data fidelity term. TV-L^1 optical flow provides robustness to illumination changes, occlusions and noise and preserves discontinuities in the flow field. Furthermore, they present an efficient numerical scheme to minimize the model employing a dual formulation of the TV energy. Fast and efficient implementations accelerated on modern graphics processing units (GPUs) exploiting the huge computational power and parallel processing capabilities enables for real-time performance of their method [50]. An extension to the approach was proposed in [45] which further improves robustness to illumination changes.

The objective of this algorithm is, given two images I_0 and I_1, to map all image points from the first image to their new location in the second one, i.e. to find die disparity field u. This is achieved through minimizing an image-based error criterion (plain intensity differences are used for measuring similarity between pixels) and a regularization term.

$$\int_\Omega \{\lambda\phi(I_0(x) - I_1(x + u(x))) + \psi(u, \nabla u, \dots)\}\, dx \tag{2.39}$$

The first term $\phi(\cdot)$ represents the data term, known as the optical flow constraint, the second term $\psi(\cdot)$ is the regularization term (shape prior) which penalizes high variations in the flow field, The parameter λ weights between them and determines the degree of smoothness.

If we choose $\phi(x) = x^2$ and $\psi(\nabla u) = |\nabla u|^2$, we again get the Horn and Schunck model from (2.38). But with the L^1 norm for penalizing the data term, $\phi(x) = |x|$ and the total variation regularization in the smoothness term $\psi(\nabla u) = |\nabla u|$, (2.39) becomes:

$$E = \int_\Omega \{\lambda|(I_0(x) - I_1(x + u(x))| + |\nabla u|\}\, dx \tag{2.40}$$

The first order Taylor approximation to linearize the image I_1 near $x + u_0$ with respect to a fixed given disparity map u_0 is $I_1(x + u_0) = I_1(x + u_0) + \langle(\nabla I_1, (u - u_0)\rangle$.

$$E = \int_\Omega \left\{ \lambda|I_1(x + u_0) + \langle(\nabla I_1, (u - u_0)\rangle - I_0(x)| + \sum_d |\nabla u_d| \right\} dx \tag{2.41}$$

Accounting to linearization, a *multi-level* coarse-to-fine iterative warping technique is employed in order to determine large scale displacements between images and to avoid to get stuck in local minima.

The expression $I_1(x + u_0) + \langle(\nabla I_1, (u - u_0)\rangle - I_0(x)$ we call now residual $\rho(u, u_0, x)$ and introduce a new auxiliary variable v that is an approximation of u to minimize the following convex approximation of the functional:

$$E_\theta = \int_\Omega \left\{ \sum_d |\nabla u_d| + \sum_d \frac{1}{2\theta}(u_d - v_d)^2 + \lambda|\rho(v)| \right\} dx, \tag{2.42}$$

where θ is a small constant to assure, that v_d is a close approximation of u_d. Minimizing the energy is performed by alternating optimization steps, where either u or v is fixed in every iteration.

1. For v_d fixed, solve for every d:

$$\min_{u_d} \int_\Omega \left\{ |\nabla u_d| + \frac{1}{2\theta}(u_d - v_d)^2 \right\} dx \tag{2.43}$$

Equation (2.43) is the total variation based image denoising model of Rudin, Osher and Fatemi (ROF) [32], that provides modeling true statistics of natural images, as

well as allowing to compute an exact solution [29].

Chambolle in [10] proposed an efficient and globally convergent numerical scheme for solving the ROF energy, which uses a dual formulation of Equation (2.43).

The dual variables are given as

$$u_d = v_d + \theta \mathbf{div} p_d, \tag{2.44}$$

where p fulfills $\nabla(\theta \mathbf{div} p - v) = |\nabla(\theta \mathbf{div} p - v)|p$. The solution is given in (2.45) with $p^0 = 0$ and the time step $\tau \leq 1/8$:

$$p^{k+1} = \frac{p^k + \tau \nabla(\mathbf{div} p^k - v/\theta)}{1 + \tau |\nabla(\mathbf{div} p^k - v/\theta)|} \tag{2.45}$$

2. Now, for fixed u, solve:

$$\min_v \sum_d \frac{1}{2\theta}(u_d - v_d)^2 + \lambda |\rho(v)| \tag{2.46}$$

The optimization problem of (2.46) can be reduced to an efficient point-wise thresholding step.

$$v = u + \begin{cases} \lambda\theta\nabla I_1 & if \quad \rho(u) < -\lambda\theta|\nabla I_1|^2 \\ -\lambda\theta\nabla I_1 & if \quad \rho(u) > \lambda\theta|\nabla I_1|^2 \\ -\rho(u)\nabla I_1/|\nabla I_1|^2 & if \quad |\rho(u)| \leq \lambda\theta|\nabla I_1|^2 \end{cases} \tag{2.47}$$

If the required step between u and v is sufficiently small, $\rho(v)$ is allowed to vanish.

Since the data fidelity term $\phi(I_0(x) - I_1(x + u(x)))$ assumes constant brightness, it is necessary to model intensity value changes. In [45], a structure-texture decomposition approach [3] using the total variation based image denoising model from [32] is proposed for this purpose. The authors made the observation, that the computation of optical flow using the textural part of the image is not affected by shading reflection and shadow artifacts. The structural part for an image I(x) is given as the solution of

$$\min_{I_s} \int_\Omega \left\{ |\nabla I_s| + \frac{1}{2\theta}(I_s - I)^2 \right\} dx \tag{2.48}$$

The textural part is then $I_T(x) = I(x) - I_s(x)$.

Chapter 3

Reconstruction from Multiple Views

Contents

3.1 Multi-View Matching

Multi-view scene reconstruction provides additional information assisting the correspondence problem. By capturing a scene from different viewpoints, multi-view reconstruction can overcome some of the shortcomings of traditional stereo. Two-View reconstruction over wide baselines or large slant provides sufficient triangulation angles for better depth accuracy, but is often not able to find correct correspondences because the visual appearance of scene points can vary significantly with viewpoint [14, 28].

Firstly, multiple views contribute to scene completeness by increased scene coverage capturing otherwise occluded regions. Moreover, multi-view reconstruction is able to improve depth accuracy by increasing the triangulation angles.

Another important aspect is, that instead of merging a set of independently determined binocular stereo depth maps, misregistration is less a problem here. Otherwise, combining individual depth maps may lead to different coordinates of the same world point due to errors in camera calibration [14, 49].

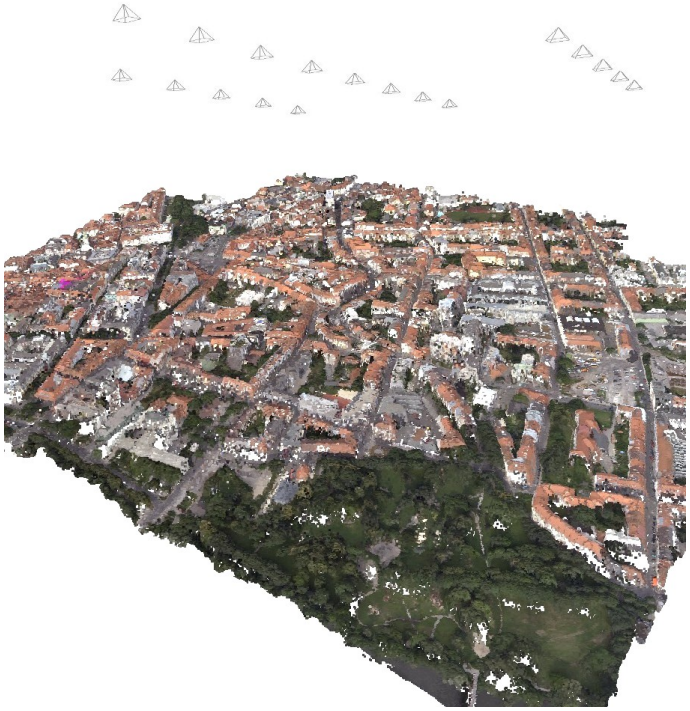

Figure 3.1: **Reconstruction from Aerial Images:** Illustration of a city model from the Jakomini dataset including the reconstructed camera positions.

In the following, we investigate and categorize multi-view stereo techniques that reconstruct dense scene models from calibrated views [34]. Methods producing sparse reconstructions from a set of feature points and structure-from-motion methods are left out in this examination. This class of multi-view algorithms starts with extracting and matching a sparse set of feature points and then fitting a surface to the reconstructed points.

There are significant differences between existing algorithms, but a first rough categorization can be made when looking at attributes including the underlying scene representation, initialization requirements and the reconstruction algorithm itself.

A lot of multi-view stereo methods studied during the last couple of years focused on the reconstruction of small objects under controlled conditions. Some of the top performing methods are able to produce results near laser-based reconstructions, but are not suited do adapt for large scale scenes.

Algorithms relying on visual hulls, that have proven to be useful for indoor scene reconstruction are among this category. The visual hull serves either as an initial guess for

further optimization, as a soft or hard constraint to be fulfilled by the determined shape [19]. As algorithms depending on visual hulls can not be applied on outdoor scenes, these methods can be discarded for our purpose.

Typical representations of the geometry of an object include volumetric approaches due to their simplicity and ability to approximate any surface, i.e. regularly sampled voxels on a discrete 3D grid or as level sets encoding the distance to the closest surface [34].

One class of multi-view algorithms operating on a 3D grid perform by sweeping through the volume, computing cost values and then extracting a surface from it [51]. The voxel coloring algorithm is an example for this kind of technique. Most methods in this group differ in the way of cost computation and surface extraction / optimization.

Polygon meshes consist of a set of connected, planar faces and are also commonly used, as they allow efficient storage, output rendering and furthermore, they are suited for visibility computation.

Methods working on polygon meshes or voxels, including space carving, volumetric graph cuts and level sets often iteratively evolve a surface to minimize a cost function. External and internal forces are applied to polygon meshes to evolve. Level sets try to minimize a set of partial differential equations (PDEs) defined on a volume, similar to space carving methods that shrink (or expand, if necessary) an initial volume [34].

Especially methods based on volumetric representations are less suited for large scale scenes due to their computation and memory costs that raise quickly when the size of the domain increases.

The most promising methods for large-scale multi-view stereo that have proven to be more applicable to e.g. architectural outdoor scenes are image-space methods representing geometry by a set of depth maps. Using multiple depth maps offers the advantage of avoiding to resample the geometry on a 3D domain. Consistency between the set of depth maps is either enforced through constraints to ensure a consistent 3D scene representation or the depth maps are merged into a 3D model in a post process step.

While some algorithms follow one single representation, others use different representation for every step of the reconstruction process.

Another property of multi-view algorithms is the way they measure photo-consistency between views. We can divide between *scene space* and *image space* methods. The former try to match a point, patch or voxel of the geometry with the input views by projecting it onto the images.

The latter warp an image from one viewpoint to a predicted image using an estimate of the scene geometry. The comparison between the measured image and the predicted one yields a measure known as the *prediction error*.

In addition to photo-consistency measures, shape priors are used to induce constraints to ensure that the results have desired characteristics, e.g. the amount of (local) smoothness.

Detailed information and an evaluation of multi-view stereo reconstruction algorithms can be found in Seiz et al. [34] and [19, 37].

3.2 Visibility and Occlusion Handling

Visibility of scene points can change dramatically with viewpoint (Figure 3.2). Some points are visible to one camera but not to others due to the geometry of the scene and camera positions from which the scene is observed. Any multi-view reconstruction algorithm needs some kind of visibility model to handle occlusions in some way or another to decide which views to consider when evaluating cost measures and correspondences.

Figure 3.2: **Visibility of Scene Points:** Visibility of scene points can change dramatically with viewpoint. The facade visible on the right of the marked building in (a) is barely visible in image (b) and totally occluded in view (c).

Figure 3.3 illustrates two scenes with occlusions. The scene on the left shows a typical occlusion observed in most scenes, less common arrow occlusions occurring at narrow structures can be observed on the right. The depths of the points P_O in both configurations can not be recovered unless additional views are added in which the points are visible or assumptions about the scene geometry are introduced.

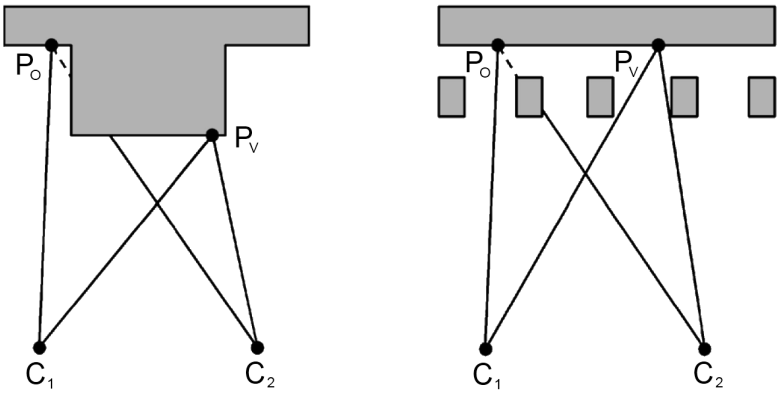

Figure 3.3: **Occlusions:** Depth discontinuities cause object to appear and disappear at different viewpoints. In each of the two scenes, two points P_O and P_V are observed from two cameras C_1 and C_2. The points P_V are visible to both cameras and their depths may be reconstructed. The points P_O are called *half-occluded* because they are visible to only one camera and not to the other, hence they may not be recovered (adopted and modified from [8]).

One approach is to either detect and handle occlusions before or after matching. For dense results, these regions are then interpolated from neighboring pixels in the results. On the contrary, if handled afterwards by detecting discontinuities in the depth maps, median filtering [8] or TV denoising [32] can be used to eliminate outliers caused by occluded regions.

In more sophisticated solutions, occlusions can be handled either implicitly or explicitly. The first technique avoids handling occlusions explicitly from geometric reasoning and treats them rather as outliers. Supporting views are selected using simple outlier rejection. This can be applied especially in cases where scene points are visible more often than they are occluded [34].

Truncated sum: The final score for the current depth hypothesis d is formed by accumulating all previously calculated matching costs between a reference image (i.e. the key view) and all N neighboring views (Equation 3.1). High individual cost values C_i of an

image point at location (x,y) above a threshold t are assumed as occluded. The matching scores are truncated and then summed up to limit the contribution of those image pairs to the accumulated total cost and instead favor good depth hypotheses supported by other views.

$$C(x, y, d) = \frac{1}{N} \sum_{i=1}^{N} \min \left(C_i(x, y, d), t \right) \tag{3.1}$$

Best half selection: Instead of using all cost scores, we split the set of images into two halves with respect to the matching cost. Only the scores of the better half (best 50% of all images available) contribute to the final aggregated cost for the current depth hypothesis.

If we assume that the images were captured in a sequence by a camera moving along a continuous path, then objects occluded along the path in one direction may be visible the reverse way. Half-occluded regions will appear either in the left (preceding) or right half-sequence (succeeding frames) with respect to a key view [23] [51].

Quasi-geometric approaches use for minimizing the effect of occlusions only a rough estimate of the shape (e.g. the visual hull) or just an approximation of the scene geometry to limit the amount of computations by clustering neighboring cameras and discard diverging views.

Explicit visibility modeling to determine which scene structures are visible in which images is commonly used in approaches that evolve a surface. These methods are called *geometric* techniques. They use a current estimate of scene geometry to predict visibility of every point on the surface.

Simplifications can be made to the visibility computation by constraining the distribution of camera viewpoints. The occlusion ordering of points can be fixed for all cameras if the scene lies outside the convex hull of the camera centers, which leads to more efficient algorithms [34].

In global optimization frameworks, pixels with erroneous matches will still have some disparity assigned. In spite that such pixels can be handled as outliers in a post processing step due to the fact that they often correspond to a high matching error, it would make sense to include visibility information into the global energy, that is being optimized [23].

3.3 Initialization Requirements

All multi-view algorithms require more input information than just a set of images and their associated camera calibration parameters. Some information about the geometric dimensions of the reconstructed scene needs to be provided to eliminate trivial shapes.

There are manifold ways of how this information is given to the algorithm [34]. Some algorithms are based on visual hulls that serve as an initial estimate for scene geometry. This implies silhouette detection, e.g. through foreground-background segmentation for each image.

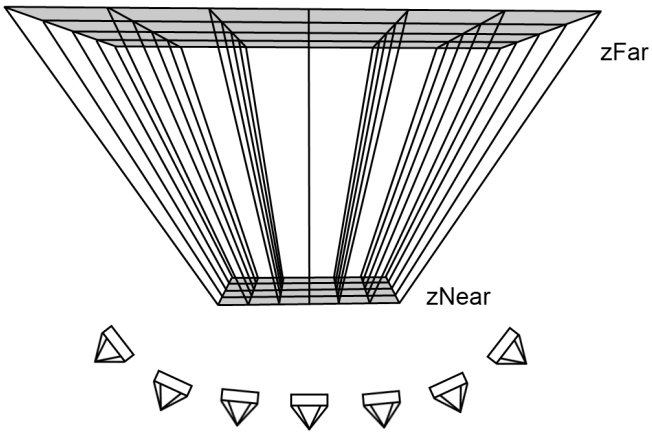

Figure 3.4: **Bounding Volume:** A near and far plane parallel to the image plane of the reference camera define a bounding volume.

For space carving methods and level sets, only a roughly estimated bounding box or volume is necessary. Approaches based on image-space algorithms usually constrain the maximum disparity range or interval, in which possible depth values can occur. The extent of scene geometry is hereby determined to lie between a near and far plane from the camera center of each view (Figure 3.4).

zNear and zFar can either be estimated from the sparse scene reconstruction (Structure-from-Motion output) or explicitly set to some global value if prior knowledge about the minimal/maximal scene depth is available (e.g. aerial mapping).

Optical flow stereo needs at least a reference point from which to start seeking for a corresponding image point. The displacement is determined with respect to that reference point. If, in addition, a rough depth estimate is available, the disparity range and hence the number of pyramid levels in a coarse-to-fine framework can be reduced.

Chapter 4

Robust Multi-View Methods

Contents

4.1 Reconstruction Pipeline Overview

The presented robust multi-view reconstruction method consists of several steps. In the following, will have an in-depth look at each of the core steps regarding dense matching and reconstruction. First, we will discuss the plane sweep method, followed by a detailed examination of the TV-L^1 optical flow approach for robust multi-view reconstruction. The implemented approaches are based on theory gathered in the preceding chapters. An illustration of the whole reconstruction pipeline is shown in Figure 4.1.

Both methods take a set of images corrected for geometrical distortions and their corresponding internal and external camera calibration parameters, together with a sparse reconstruction of extracted SIFT feature points from the Structure-from-Motion step and camera calibration stage (Figure 4.2).

The number of shared sparse feature points determines the amount of overlap and thus which views are considered as neighbors. Then, we compute a depth map for one of the input images (i.e. the reference or key view) and its neighbors (the sensor images, respectively). Every image serves as a reference view only once.

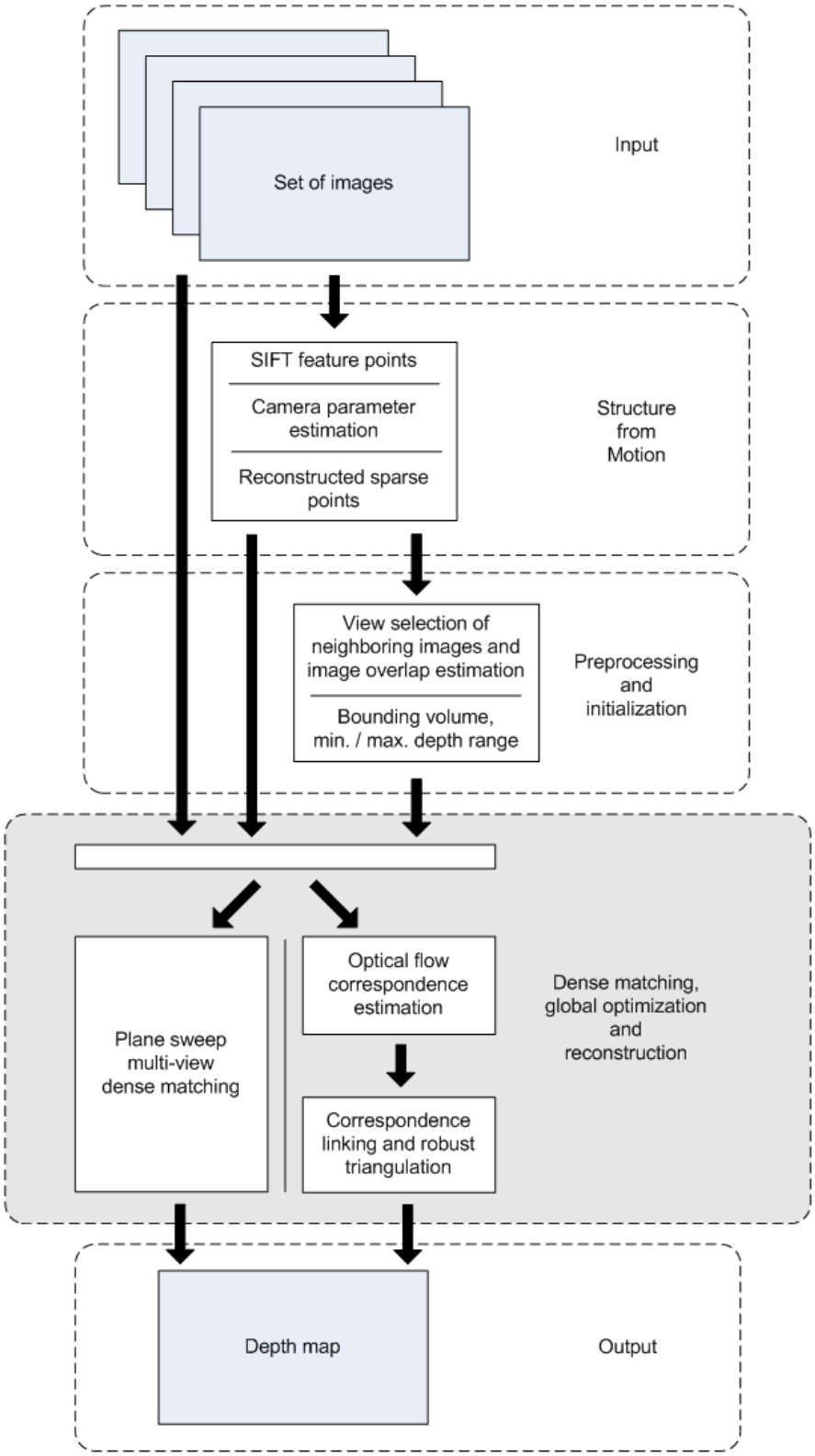

Figure 4.1: **The Reconstruction Pipeline:** Both implemented dense matching methods take a sequence of calibrated images and information about the scene extent as input and output a depth map with respect to a key view. The optical flow approach, in addition, may be initialized with rough depth estimates from the sparse reconstruction.

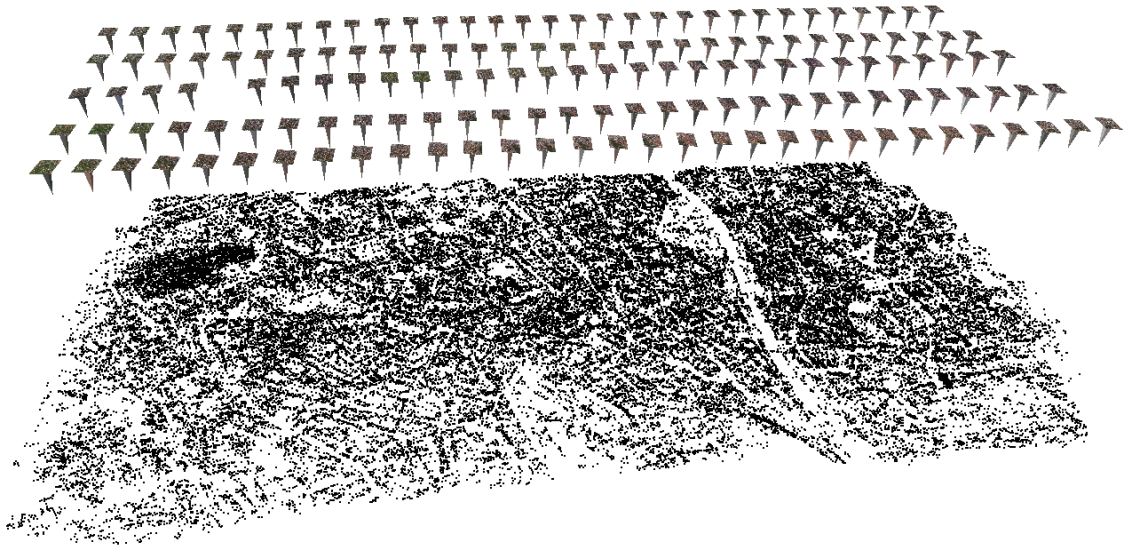

Figure 4.2: **Point Cloud from Structure-from-Motion:** The point cloud is a reconstruction from 154 aerial images of the city of Graz.

4.2 Depth Estimation using Plane Sweep

4.2.1 The Plane Sweep Principle

Our first implemented method for multi-view matching is based on a plane sweep technique and enables a simple and elegant way for image based multi-view reconstruction. It allows to reconstruct depth maps from arbitrary collections of images and an implicit aggregation of multiple view's matching costs. The plane sweep approach described here is similar to that used in the high-performance multi-view reconstruction method from Zach et al. [51].

In a plane sweep approach, 3D space is iteratively traversed by parallel planes $\Pi_d = (n^\top, d)$ aligned with the key view and positioned at an arbitrary number of discrete depths. The plane at a certain depth d from the reference view induces homographies for all sensor views. The sensor views are then mapped onto this plane [11, 51]. The principle is illustrated in Figure 4.3.

The plane sweep technique is based on the idea that if the plane at a certain depth passes exactly through the object's surface that we want to reconstruct, then, under constant brightness conditions, the appearance of the image points (i.e. the color or intensity values) in the key view should match with those of the projected sensor image

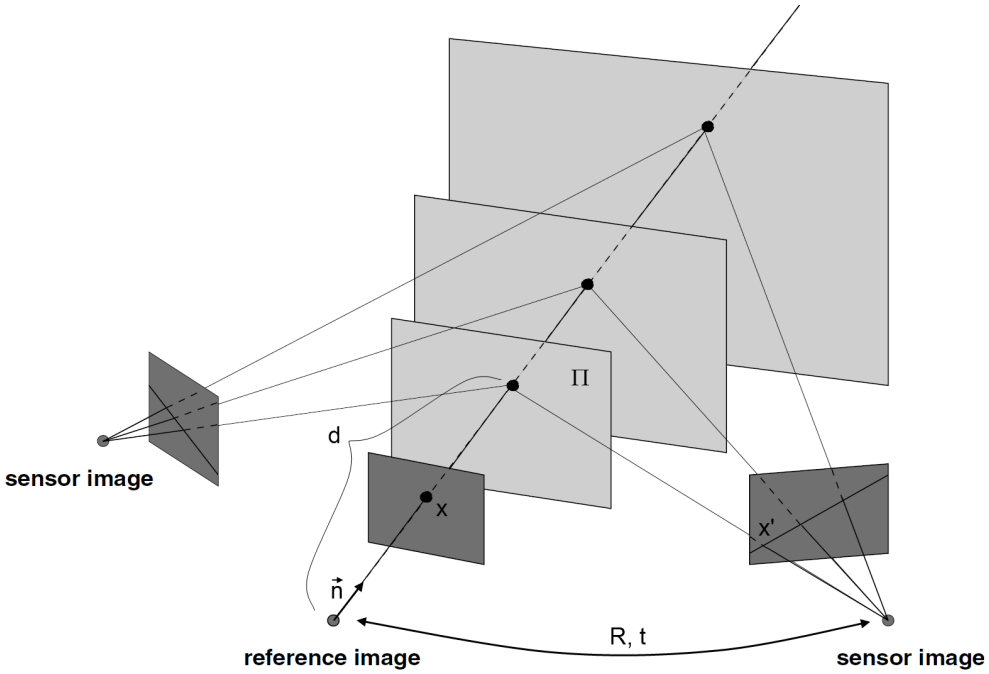

Figure 4.3: **The Plane Sweep Principle:** The 3D space is iteratively traversed by parallel planes. Different depths yield to varying homographies between reference and sensor views. When projecting the sensor image onto these planes, the image is transform according to the epipolar geometry (adopted and modified from [11]).

at the appropriate positions.

By sweeping through 3D space with a plane at varying depths parallel to the key view, a cost volume can be filled with the combined similarity scores from all sensor images.

The number of depth steps in the volume is chosen with respect to a desired pixel accuracy (e.g. sub-pixel accuracy is required) over the whole set of images. The vertices of the bounding volume are backprojected into all views. The value of the depth step is then based on the intersection covariance of the reconstructed vertices of the bounding box. The intersection covariance (uncertainty ellipsoid) $C^{(e)}$ is analytically computed as proposed in [5]. Using a singular value decomposition,

$$C^{(e)} = U \begin{pmatrix} \sigma_1^2 & 0 & 0 \\ 0 & \sigma_2^2 & 0 \\ 0 & 0 & \sigma_3^2 \end{pmatrix} V^\top \tag{4.1}$$

$\sigma_1 \dots \sigma_3$ is determined and the depth step Δd is adjusted according the length of the

semi-major axis of the uncertainty elipsoide $\Delta d = 2 \times \sigma_1$.

The best combined matching scores obtained through some local image correlation measure are assumed to correspond to certain depths which are then assigned to the pixels of the key view. This means that the final depth for each pixel can be determined by a simple winner-takes-all (WTA) strategy along the optical ray through that pixel in order to achieve high performance and low memory requirements for depth estimation. On the contrary, WTA depth extraction suffers from the lack of smoothness and unreliable results in low texture regions.

If depth discontinuity or any other constraint is required, depth extraction from the cost volume can be performed using global optimization methods to avoid unreliable values with low image correlation scores. This allows for smoothness while preserving depth discontinuities to obtain high quality depth maps. Here we rely on the convex formulation of continuous multi-label problems proposed in Pock et al. [30].

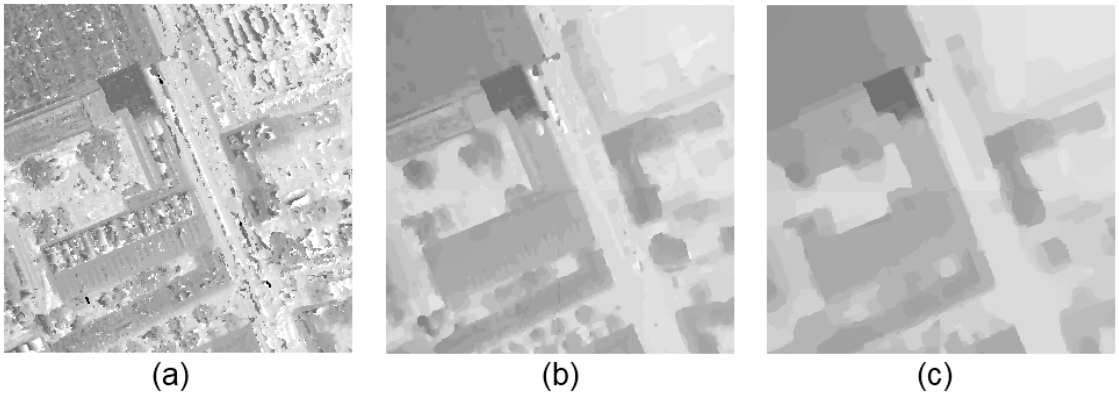

Figure 4.4: **Winner-Takes-All vs. Global Optimization:** (a) Winner-takes-all depth extraction versus global optimization with (b) $\lambda = 100$ and (c) $\lambda = 20$. The threshold for SAD truncated sum cost accumulation is $t = 0.17$.

The mapping of the sensor image into the current 3D key plane Π_d at depth d is given by a homography $H(d)$. The appropriate homography can be computed directly from the camera matrices and the value of d. The plane is defined by the depth d (i.e. the distance from the camera center) and $n^\top$ represents the vector in viewing direction (i.e. the plane normal).

$$\Pi_d = (n^\top, d) \tag{4.2}$$

We assume a canonical setup with the camera center of the key view located at the coordinate frame origin. Then, the camera projection matrices of the key view P and the sensor view P' yield to

$$P = K[I|0] \quad \text{and} \quad P' = K'[R|t]. \tag{4.3}$$

In Equation (4.3), K denotes the intrinsic parameters of the cameras. The relative pose of the sensor view with respect to the key view is given by the rotation matrix R and the translation vector t. Hence, the appropriate homography calculates to

$$H(d) = K' \left(R - \frac{tn^\top}{d} \right) K. \tag{4.4}$$

When warping the sensor image onto the plane, the image is transformed according to the epipolar geometry:

$$x' = Hx. \tag{4.5}$$

Using geometrically corrected, undistorted images has the advantage, that these are the equivalent images of ideal pinhole cameras [11].

Since the sensor images are transformed by the appropriate homography, there is no need for a rectification procedure as required in many traditional stereo matching algorithms. In case of a two frame setup with a rectified image pair, the plane sweep technique is equivalent to traditional stereo methods for disparity estimation. The homography between the plane parallel to the key view and the sensor image is then reduced to a pure translation along the X-axis between the views.

As the depth (and accordingly disparity) corresponds to a plane in 3D space, the cost function can be described as a function of the homography H used to map the sensor image onto that plane. Since the homography itself is a function of the depth d, we can now write the initial unaggregated matching cost function as:

$$C_i(x, y, d) = f_c(I_k(x, y), H(d)I_i(x, y)) \tag{4.6}$$

In Equation (4.6), $I_k(x, y)$ is the intensity of the key image at position (x, y) and $H(d)I_i(x, y)$ represents the intensity of the warped sensor view. The function f_c can be any similarity or error score to measure the intensity or color difference between key view and sensor image [23].

Any possible matching function can be used, but as we cannot expect constant brightness, robust window based measures should be preferred over pure pixel intensity differences.

All steps in this high performance multi-view reconstruction algorithm are suited to be performed on modern programmable graphics hardware, utilizing powerful computational capabilities [51].

4.2.2 Cost Functions and Aggregation Schemes

After mapping the sensor image onto the plane of the current depth hypothesis via the homography, we need to find corresponding points between key and warped sensor view by means of an error measure [11].

Many reconstruction approaches running on GPUs measure similarity between the images using the sum of absolute differences (SAD), mainly for performance reasons [51]. Besides SAD and cross correlation (CC), our implementation provides a large set of different dissimilarity measures, most of them for both CPU and GPU.

The following listing presents a summary of cost functions and methods to reduce the influence of radiometric differences. We provide a description and its main properties for each technique and its advantages and disadvantages examined with respect to our approach. Since we cannot presume constant brightness for long image sequences, the use of either normalized cost functions or an optional prenormalization step is advised. We implemented Bilateral filtering for background subtraction (BilSub), but other methods for prenormalization like subtracting the box-filtered image or mean filtering respectively are possible [51]. The investigation of the influence of different cost functions that are insensitive to radiometric differences was mainly motivated by [20].

Absolute Difference: The absolute difference is available as a pixel-based as well as its window based variant. Besides the original sum of absolute differences (SAD),

two normalized variants ZSAD and ZNSAD are able to compensate for radiometric differences up to a certain degree.

Squared Difference: The sum of squared differences (SSD) and two normalized variations compensating for gain and bias (ZSSD and ZNSSD) are very similar to the absolute difference, but tend to be more sensitive to outliers due to the squaring.

Cross Correlation: In contrast to the predecessors, this commonly used cost function is a similarity measure. It is available in its normalized (NCC) and zero-mean normalized (ZNCC) versions. Statistically, cross correlation is the best measure to deal with Gaussian noise, but it tends to blur discontinuities more than many other matching costs, as outliers lead to high error scores. Face to Face with ZNSAD and ZNSSD, it produces comparable results but its able to be calculated faster. Further acceleration is possible when using sum tables for an efficient implementation.

Bilateral Background Subtraction: It is not a similarity measure but a filter. Bilateral filtering for background subtraction (BilSub) [1, 20] effectively removes local offsets in pixel brightness. It allows for smoothness without blurring high contrast texture. Matching can be performed using the absolute difference or, as originally proposed by calculating the distance in CIELab color space. Due to its moderate effect on the results, it is not considered any further in our examination.

The Rank Transform: As all non-parametric measures have in common, Rank does not match pixel intensities itself. The rank transform substitutes a pixels intensity with its rank among its neighbors. While this measure is typically robust and insensitive to illumination changes and tolerates a small number of outliers within its neighborhood as long as the local ordering of the intensities is preserved, it is known to be susceptible to noise. Due to the loss of ordering information during the transform, the discriminatory power of this similarity measure is reduced, which leads to mismatches. Rank and its variant SoftRank are implemented as a filter. Matching is then performed using the absolute difference.

The Census Transform: The census transform is an extension of the rank filter and preserves the spatial ordering of the intensities in a bitstring. The similarity between two bitstrings is computed by calculating the Hamming distance between them, that is the number of bits that differ. Matching with census showed overall good performance under changing illumination conditions.

DAISY Descriptor: Instead of local correlation metrics, DAISY [39] is a local descriptor, fast enough to be used for dense matching. The authors provide an implementation of the descriptor on their project website, which was integrated to be tested for matching in the plane sweep approach. The descriptor provides reliable results over wide baselines but it's computation and matching of the long feature vectors with the Euclidean distance is comparatively slow on the CPU.

SAD and ZNCC are still considered as the matter of choice over all other measures, since they compute fast and enable decent results. Especially the normalized variants are always a good option, when radiometric differences can be expected in the image data.

We focus cost computation based on luminance (intensity) rather than on color when matching pixels. If color matching is needed anyhow, Cornelis et al. in [11] propose an error measure based on the distance in RGB-space:

$$C = (r_k - r_s)^2 + (g_k - g_s)^2 + (b_k - b_s)^2. \tag{4.7}$$

Another option is to use the distance defined in an alternative color space, e.g. in CIELab space for matching as proposed in [40] or extending intensity-based matching scores to colors by computing costs for each channel individually and afterwards combining them in a (weighted) sum over all channels [20].

The image correlation measures AD, SD, CC, Rank and Census take a parameter r to set the radius of the support window. For performance reasons, all cost functions are restricted to a fixed window size as usual in high performance dense matching. Adaptive or shiftable windows are not considered in our implementation, nevertheless it would easily be possible to integrate these approaches.

Anyway, to increase performance at depth discontinuities and object boundaries, the multi-resolution cost aggregation scheme described in Chapter 2 is available for several cost functions.

4.2.3 Accumulating Similarity Scores and Implicit Occlusion Handling

Now we have calculated the matching scores for all pixels at the current depth between all projected sensor views and the key view, the costs need to be combined in one way

or another to fill the cost volume and to account for occlusions. In real-time and high-performance applications, occlusion handling is usually performed implicitly. The simple blending method is sufficient, if no implicit occlusion handling is desired. Obviously, a winner-takes-all approach that assigns the minimum error from all sensor costs and the depth to the pixels of the current plane hypothesis is applicable.

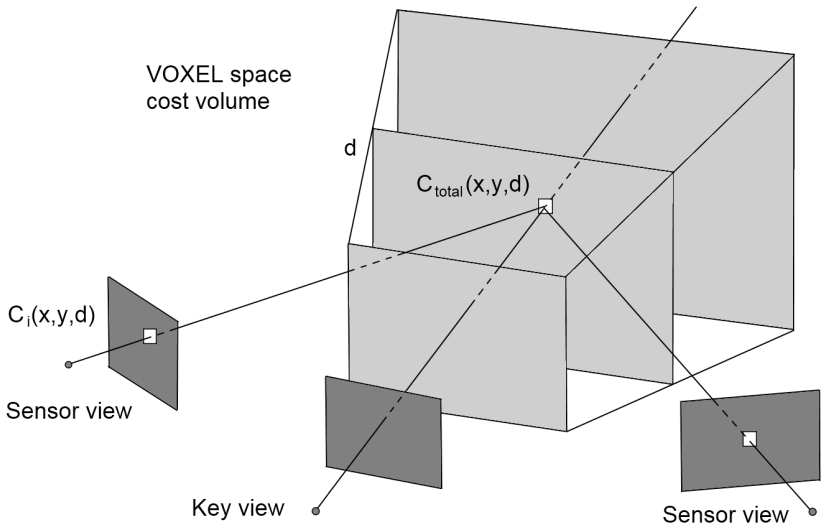

Figure 4.5: **Voxel Space Cost Accumulation:** The cost values $C_i(x, y, d)$ of each pixel from all individual sensor views are combined to a total cost score of the current plane hypothesis d (adopted and modified from [11]).

A variant of the blending allows to account for occlusions with a thresholding step before summation. The threshold t is an arbitrary value indicating the maximum cost for classifying good matching cost from bad values. Values below the threshold are approved as good matches, values above are considered as occluded.

This strategy referred to as the truncated sum limits the effect of occlusions on the total error score to favor good depth hypothesis by other image pairs. This is the standard method for occlusion handling in our plane sweep implementation.

If we assume a logical sequence of views with a total ordering in the set of images, then we can perform a best half selection. The image set is split into two half-sequences with respect to the matching cost. Since half-occluded regions may yield to high matching costs, only the best 50% of the cost values are accumulated and contribute to the final cost for the current depth hypothesis.

These occlusion handling policies limit the impact of occlusions and enhance the quality of the obtained depth maps [11, 51]. It is also possible to integrate a thresholding step into the latter approach, which then yields a truncated best half cost accumulation scheme.

In the case, we have to deal with larger baselines between views, occlusions should be handled explicitly. Another shortcoming presents the fact that we assume fronto-parallel surfaces for the correlation windows, which also reduces the reliability of the obtained results [51].

4.2.4 Depth Extraction

4.2.4.1 Winner-Takes-All

Now we have filled the cost volume with accumulated cost scores for all depth hypothesis, we can extract the final values by selecting the depth where the accumulated cost has its minimum along the optical ray. A simple winner-takes-all strategy is often employed in high performance applications. The main advantage is that there is no need to store the whole cost volume. Hence, the depth value of the current depth hypothesis is assigned to the pixel, if the cost is lower than that of the previous depth hypothesis. WTA only requires 2*n*m memory for a $n \times m$ image and can be computed in O(n*m*d) time, where d is the number of discrete depth steps.

4.2.4.2 Robust Median Depth

Instead of accumulating the costs for every depth hypothesis from all views first, we can select a best matching depth value for the pixels in each individual sensor view through a winner-takes-all strategy along the optical ray.

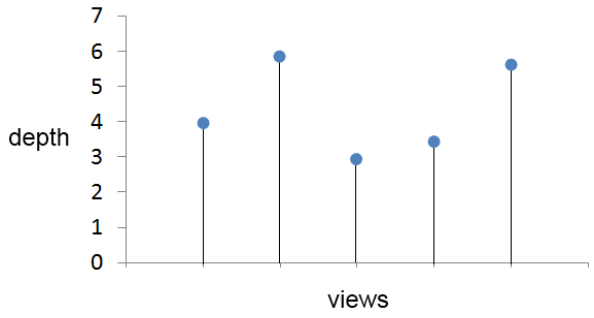

Figure 4.6: **Median Depth:** The best overall depth for a key view pixel is selected from the best matching depths in a robust manner by choosing the median depth value from all five sensor views, d = 4 in this example.

Every sensor view now has depth values assigned, which are considered the best matching depths for that specific view (Figure 4.6). Selecting the median depth value is a robust way of determining the overall final depth for a specific pixel of the key view.

4.2.4.3 Global Optimization through Multi-Label Problem

The depth maps obtained using these methods described so far do have a substantial amount of outliers since the local techniques do make errors. A global optimization approach can be applied to extract depth values from the cost volume, if depth discontinuity or any other constraint on the depth maps is needed.

The global depth map optimization is formulated as a multi-label problem [30] and solved via minimizing an energy functional incorporating total variation regularization. The goal is to assign a label (i.e. a depth value) to every pixel to get a label configuration that is minimal with respect to an energy function.

$$\min_u \left\{ \int_\Omega |\nabla u(x)| dx + \int_\Omega \rho(u(x), x) dx \right\} \tag{4.8}$$

A disadvantage of this approach is its complexity and that it operates on a 3D domain making it costly with respect to computation time and memory consumption. In spite of an efficient implementation for GPUs to solve the multi-label problem, the limited memory resources on graphics hardware poses the biggest drawback, since the whole cost volume needs to be stored on the graphics card.

The depth precision depends on the angles between the views and on the extent of the scene to be reconstructed. A certain number of depth hypotheses is required to guarantee sub-pixel accuracy. The large memory requirements of a voxel based approach can be handled through a partitioning approach, where the input images are split into smaller tiles of a usual size of 256x256 or 512x512 pixels with a typical number of 128 to 256 depth hypotheses.

4.3 Dense Depth Maps from TV-L^1 Stereo

4.3.1 The Matching Approach

Here we present a 3D reconstruction approach from multiple images that is based on optical flow to solve the correspondence problem. Optical flow seeks to estimate the motion of pixels from one frame (the key view) to another (a sensor view). With the

motion encoded as usual in a two-dimensional disparity field, we are able to directly extract correspondences.

The TV-L^1 stereo approach [29, 45, 50] is robust to brightness variations in the image data and preserves depth discontinuities. The performance on the results at object borders can be further increased by additional edge weighting, giving more weight to high contrast areas within the image associated with depth edges. Furthermore, it can be implemented very efficiently to exploit the huge computational power of modern graphics hardware.

Since we have to face a huge amount of data when matching high resolution aerial images in our setup, we need an approach that is able to handle this task efficiently. The TV-L^1 stereo method fits perfectly for this purpose.

The use of the first order Taylor approximation of the nonlinear image intensity profile in the data term implies an iterative warping approach, because the approximation is only valid for small disparities. To allow for large displacements, the flow estimation is embedded into a multi-level coarse-to-fine framework (Figure 4.7).

Disparity estimation then corresponds to searching for the shortest path through all pyramid levels, instead of global optimization in a cost volume as this is the case in the former approach.

Figure 4.7: **Coarse-to-Fine Pyramid Levels:** The level of detail increases from coarse structures on low scales to finer details at higher pyramid levels.

A CPU implementation is established for all experiments. We utilize an image pyramid with an adjustable number of levels, a factor determining the scaling between the pyramid levels and a smoothing parameter to realize the coarse-to-fine approach. The fully featured implementation of the scale space in our method is similar to [42] and based on [26].

4.3.2 Initialization

The two views for which a flow field needs to be obtained with standard optical flow methods should not show large displacements or heavy rotations. This almost requires that the images where captured in a close spatio-temporal sequence. The flow seeks for a corresponding pixel location in the sensor view starting from the coordinates of the reference pixel in the key view.

Because images in our datasets can show large displacements and partly rotations of 180 degrees between the views, standard optical flow algorithms will run into trouble since they do not account for the relative orientation of the views and camera geometries during disparity estimation. Epipolar geometry in fact restricts the direction of the correspondence search, though we need a reference or anchor point on the epipolar line to estimate the displacement with respect to that reference.

In order to be able to estimate disparities correctly, it is important to initialize with an adequate reference depth. Initialization is indispensable for the algorithm to work. Otherwise, no reference location is given in the image from where to start searching for correspondences.

4.3.2.1 Small Depth Variance

We define a reference depth plane somewhere within the depth range of the scenes minimum and maximum depth value. In a fronto-parallel setup (e.g. aerial images) with comparatively compact dimensions of the scene in z-direction and evenly distributed depth values, a reference plane parallel to the image plane of the key view seems to be sufficient. Alternatively, the reference plane can be chosen for example as the least squares plane with respect to the SfM sparse points that are visible in the key and sensor view.

A pixel from the key view then unprojects to a 3D point with the reference depth. This 3D point now backprojects to an image location in the sensor view to serve as a staring point for flow estimation.

4.3.2.2 Wide Depth Variance

In scenes presenting a wide depth range and hence images, where large displacements can be observed, additional assistance with an initial depth estimate is necessary.

The initialization for the disparity field can be extended by a rough depth map obtained by some other technique. The initial depth values are converted to a disparity map anchored at the appropriate reference coordinates (Figure 4.8). The better the initialization, the less pyramid levels are necessary for the algorithm and the faster the method is able to deliver satisfying results.

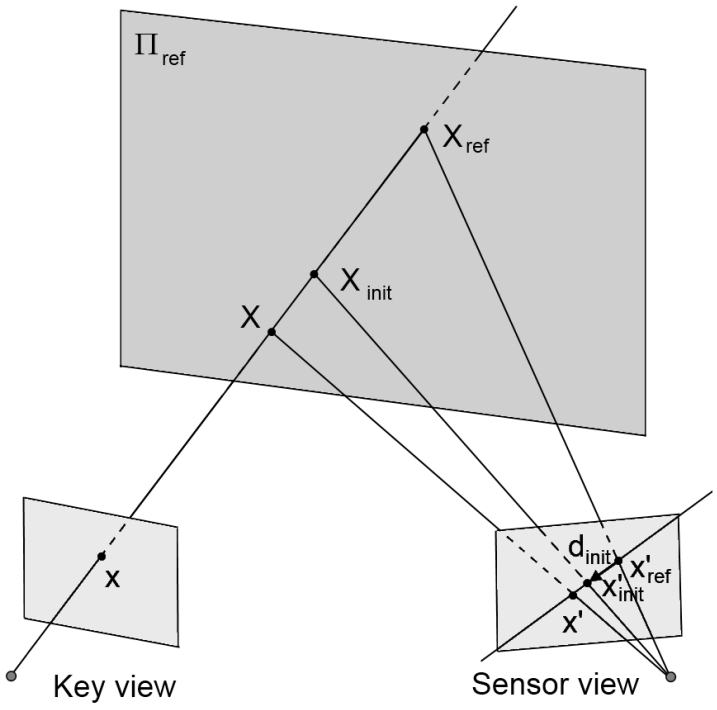

Figure 4.8: **Flow Initialization and Epipolar Geometry:** Initialization is a crucial point of estimating the flow. We define a reference plane Π_{ref} at a certain depth. Pixels x of the key view are unprojected to that depth giving a 3D point X_{ref}. This point is projected into the sensor view to a location x'_{ref} serving es the starting point for disparity estimation. If the displacement is measured in 2D, then the flow is represented by a vector $d = (u, v)^\top$. The epipolar constrained flow reduces to a scalar value defining the length of the displacement along the epipolar line, measured from the starting point. Additionally available depth information X_{init} can be used to assign an initial disparity.

We generate an initial depth map from the reconstructed sparse points by backprojecting them onto fronto-parallel patches with a fixed radius into the key view. Even a rough depth estimate to initialize at a very small pyramid level is able to improve the results (Figure 4.9 and 4.10).

Generating a polygon mesh from the sparse points is possible. A good way to roughly estimate a point's position is to assign depth values within low density regions using *radial basis functions* (RBFs) based on the distance to known points in 3D. Newcombe and Davison use this approach in their recent work on live dense reconstruction proposed in [27].

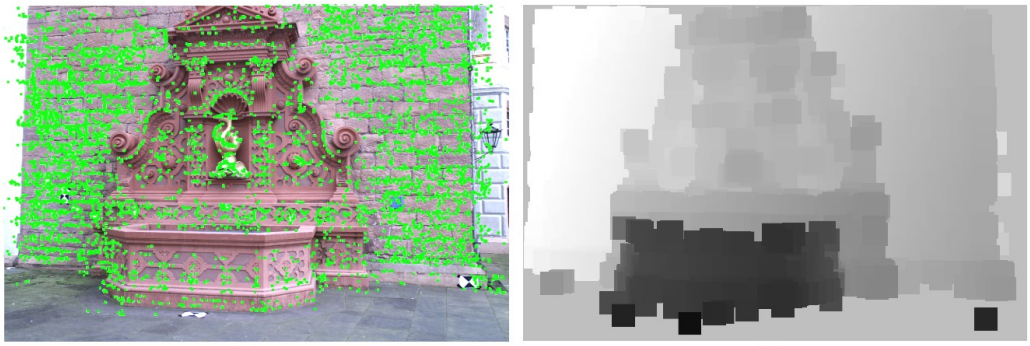

Figure 4.9: **Initialization from Sparse Points:** (a) Key view image with sparse points overlay for one image from the fountain-P11 dataset from [37]. (b) The depth map for initialization is generated from the reconstructed sparse feature points.

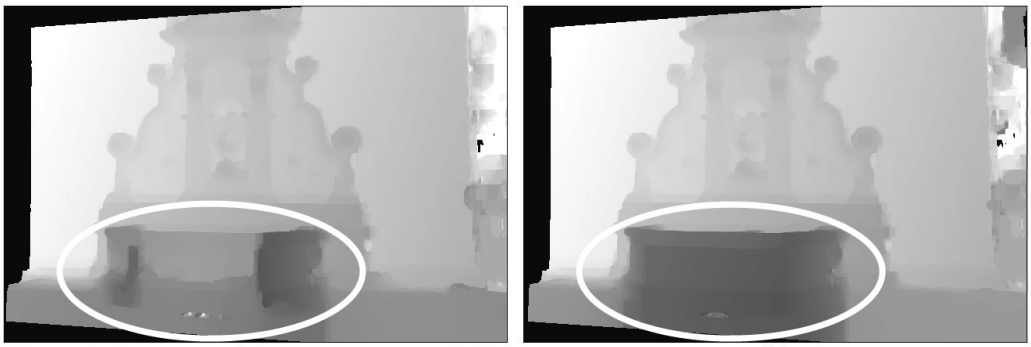

Figure 4.10: **Influence of Initialization:** Results for TV-L^1 stereo matching between adjacent images of the fountain-P11 dataset (a) without initialization and (b) with initial depths provided to the algorithm. Wrong depth values at foreground pixels in (a) result from erroneous correspondences estimation due to large displacements.

4.3.3 Disparity Estimation with Epipolar Constrained Flow

We present a modified TV-L^1 optical flow motivated by [35] and [44] and extend the algorithm to integrate the epipolar constraint. For mostly stationary scenes with a moving camera and known camera parameters, the displacement field cannot be arbitrary since the epipolar constraint must hold. Hence, the correspondence search can be restricted to one dimension. The flow then corresponds to the disparity along the epipolar line.

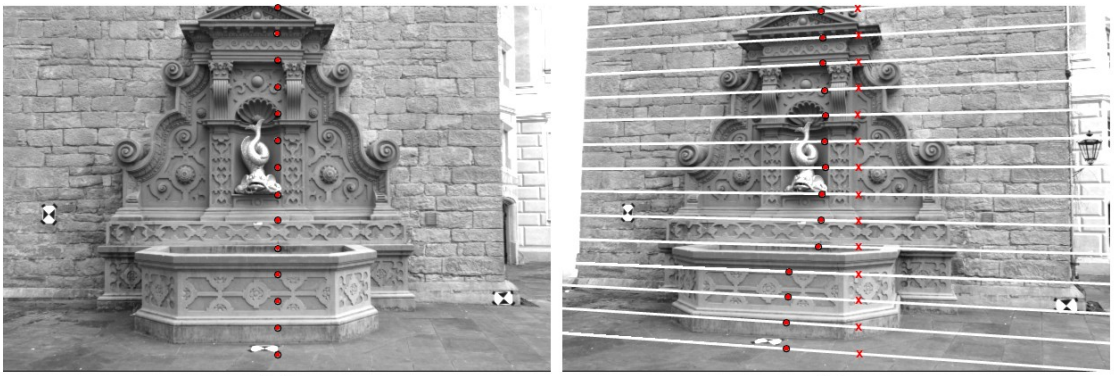

Figure 4.11: **Epipolar Constrained Flow:** (a) The image represents the key view with exemplary pixels marked with a small circle. In order to seek for correspondences, these pixels are unprojected to the reference depth and then projected to the sensor view (b) to the positions marked with a cross. The flow seeks for matches along the epipolar lines and finds correspondences, marked again by a small circle.

Wedel et al. [44] add a fundamental matrix prior as an additional data term to the total variation optical flow. A hard constrained approach as used in [35] satisfies the epipolar constraint with a suitable decomposition of the displacement field.

Integrating the epipolar constraint into the linearization step in the TV-L^1 optical flow allows to reduce the dimensionality of the flow to one dimension. The computational effort is then similar to a standard stereo case with a rectified image pair. Figure 4.8 shows the principle of our approach.

The epipolar line in the sensor view for a key view pixel x is given with $x \mapsto l'$:

$$l' = Fx \tag{4.9}$$

The direction of the epipolar line given by the unit vector l'_n together with a point on the line (i.e. the initial reference point obtained from a reference depth plane) and a given disparity u_0 yields the location of the point correspondence x':

$$x' = x_{ref} + u_0 l'_n \tag{4.10}$$

We linearize image I_1 near x'. The derivative of the image intensity with respect to the x- and y-direction respectively in the Taylor approximation of the original 2D flow (Equation (2.41)) now changes to the gradient along the epipolar line. With I_1^e denoting the derivative with respect to the epipolar direction, the energy functional then reads

$$E = \int_\Omega \left\{ \lambda |u I_1^e + I_1(x') - u_0 I_1^e - I_0| + |\nabla u| \right\} dx \tag{4.11}$$

The energy is minimized as described in Section 2.2.2.2.

4.3.4 Correspondence Linking and Robust Reconstruction

Dense correspondence computation is performed between all pairs of images. A pair always consists of the key view and one of its neighboring sensor views. What we get is a set of correspondences (i.e. measurements) for each pixel of the key view, one from every neighboring view in which the pixel is visible. We are then able to empower this redundant information from multiple views to assist in the reconstruction problem.

Reliable correspondences can be expected for direct neighbors. The quality of the obtained disparity maps decreases steadily with wider baselines and larger viewing angles due to occlusion. The possibility of a false match increases with a wider baseline. Hence, only measurements from adjacent neighbors provide sufficient confidence for the reconstruction of the 3D position and its depth.

Small baselines on the other hand, introduce inaccuracies to the reconstruction due to narrow triangulation. We suggest a method for *correspondence linking* between neighboring views to provide a large number of measurements from multiple views, motivated by [28] and [14].

For every pixel in the key view, the maximum possible number of measurement is equal to the number of sensor views in which the pixel is visible. A minimum of at least one measurement is needed in order to be able to triangulate a 3D point. Ideally, we would

like to have as much measurements as possible for more robustness. Robustness and depth accuracy profit from additional measurements and wider triangulation angles.

Since the reliability of a measurement depends on the proximity between key and sensor camera, we take a look at the baselines between the views. We made the observation that our epipolar constrained TV-L^1 stereo provides trustworthy correspondences only for small baselines (i.e. in most cases only for the direct neighbors). The decision whether the correspondence is suited for direct matching or not is based on the image overlap and hence on the distance between the views. An example is given with Figure 4.12.

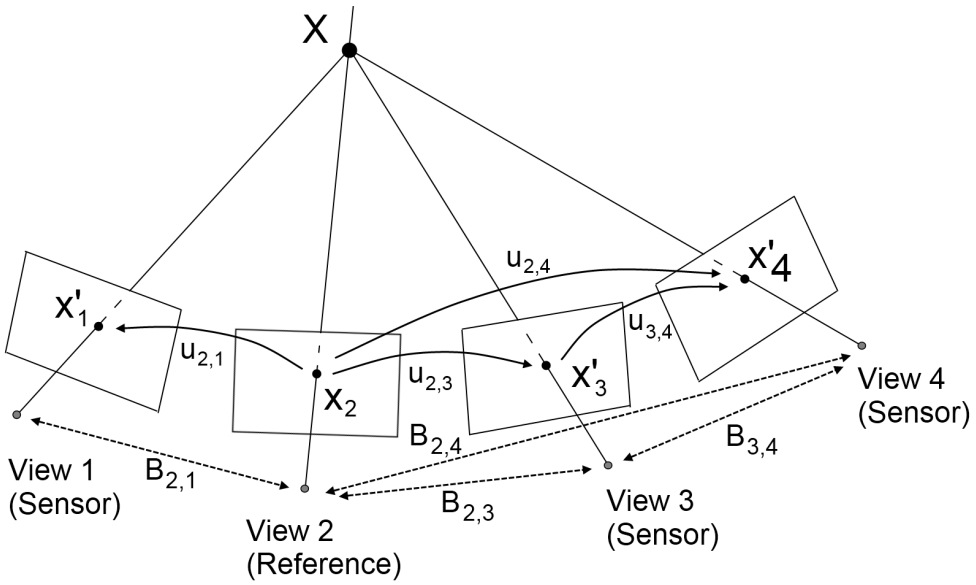

Figure 4.12: **Correspondence Linking:** The measurement of x'_4 is not considered reliable due to the wide baseline $B_{2,4}$. Robustness and depth accuracy would profit from an additional measurement and a wider triangulation angle. Because a correspondence x'_3 in view 3 is known together with a disparity estimation $u_{3,4}$ between view 3 and view 4, we can perform a lookup to add a linked measurement for x'_4.

In Figure 4.12, a scene point X at key view location x_2 is visible in all sensor views. Pairwise disparity estimations u between the key view (view 2) and all sensor views yield measurements $x'_1 = x_2 + u_{2,1}(x_2)$, $x'_3 = x_2 + u_{2,3}(x_2)$ and $x'_4 = x_2 + u_{2,4}(x_2)$.

Due to the wide baseline $B_{2,4}$ between key view and sensor view 4, the disparity estimation $u_{2,4}$ is not considered reliable. However, robustness and depth accuracy would profit from an additional measurement and a wider triangulation angle. Since a correspon-

dence x'_3 in view 3 is known and a disparity estimation $u_{3,4}$ between view 3 and view 4 is available, we can perform a lookup and are able to add a linked correspondence for view 4 over the linking view 3 through $x'_4 = x_2 + u_{2,3}(x_2) + u_{3,4}(x_2 + u_{2,3}(x_2)) = x'_3 + u_{3,4}(x'_3)$.

We start by sorting the sensor views according to their baseline to the key view in ascending order and start traversing the list of views with the one that is the closest to the reference view. At the beginning, we hold an empty list for the key view pixel x_k to store valid measurements from every sensor view. If the baseline is below a certain threshold and a measurement (i.e. a disparity estimate) is available, we add the correspondence directly. The threshold defines the maximum baseline allowed for direct correspondences. Its value is chosen to favor adjacent views with good overlap for adding direct measurements.

In case that the baseline is above the threshold, we are trying to find a link, i.e. the disparity estimate from the nearest sensor view to the current sensor that already had a correspondence added to the list. This view is now referred to as the *link* or *linking view*.

If a disparity estimate $u_{l,c}$ between the linking view l and the current sensor view c is available, we update its coordinates according to $x'_c = x_k + u_{k,l}(x_k) + u_{l,c}(x_k + u_{k,l}(x_k)) = x'_l + u_{l,c}(x'_l)$ and add it to the list of measurements. The principle of our correspondence linking approach is a lookup operation whereas it is not important for the method whether an existing correspondence was obtained directly or through linking itself.

Correspondence Linking Algorithm:

1. Sort all sensor views according to their baseline to the key view k.
2. Traverse the list of views and examine the baseline of the current sensor view c.
   (a) If the baseline is smaller than a threshold and a disparity estimate exists, then add a direct measurement x'_c.
   (b) Otherwise, if the baseline is larger than the maximum baseline allowed for direct linking:
      i. Sort the sensor views with respect to the baseline of the current sensor.
      ii. Choose the closest view that had already a measurement assigned (i.e. the link) and for that a disparity estimate to the current sensor view exists.
      iii. Load disparity estimates $u_{l,c}$ and update the measurement's pixel coordinates according to $x'_c = x'_l + u_{l,c}(x'_l)$.

Furthermore, we employ a robust triangulation strategy based on the *RANdom SAmple Consensus* (RANSAC) [17] algorithm to provide robust depth estimates in the reconstruction. The objective of the RANSAC algorithm is to robustly fit a model to a set of data points that contains outliers.

We iteratively select a random set of data points (i.e. the measurements) and generate a depth hypothesis (i.e. the model). All data points are then tested against the current hypothesis. Points supporting the current model are considered as inliers and contribute to the *consensus set*.

Robust Triangulation Algorithm:

1. Select a random number of measurements (i.e. the data points). We randomly pick a number of measurements between two and the maximum number of available measurements to establish a depth hypothesis.

2. A 3D point as an initial depth hypothesis (i.e. the model) is obtained by triangulation from the randomly selected measurements.

3. The set of measurements is tested whether it supports the current model (inliers) based on the reprojection error of the triangulated 3D point. The current 3D point is projected into all views to decide if the view's measurement supports the model. If the distance of the reprojected point to the initial measurement in that specific view is larger than the reprojection error threshold, then the current measurement depicts an outlier, an inlier otherwise. The subset of inliers from all views is the current consensus set.

4. Repeat steps 1-3 until a reasonably large consensus set is found that supports the model. In each iteration, the algorithm produces a model that is either being rejected due to a too small number of inliers or kept if the consensus set is larger than that of the last saved model. The final model with the most support, i.e. the largest consensus set with the most inliers depicts the robust fit after a certain number of iterations.

The results depend mainly on the choice of the maximum reprojection error, the number of iterations and the minimum size of the consensus set. Currently, the maximum reprojection error used to classify inliers and outliers is a constant parameter, selected by the

user. A better way for selecting a threshold value is to calculate the average reprojection error from the sparse points to offer an appropriate measure.

We set the maximum reprojection error to 0.3 pixels in our experiments and defined a number of three measurements as the minimum size for the consensus set. The number of iterations depends on the size of the dataset. If we have a number of N measurements, there exists a number of 2^N possible combinations. A subset of 20-30% of all possible configurations for $N > 10$ was usually sufficient in our experiments to obtain a robust depth estimate. We tested all combinations for smaller datasets.

The size of the consensus set for each pixel is used as a confidence value and encoded in a confidence map. The confidence map illustrates the number of correspondences that were selected as reliable for triangulation. Examples are given in Figure 4.13 and 4.14.

This approach treats occlusions and false matches as outliers instead of detecting them beforehand, since we do not handle occlusions explicitly.

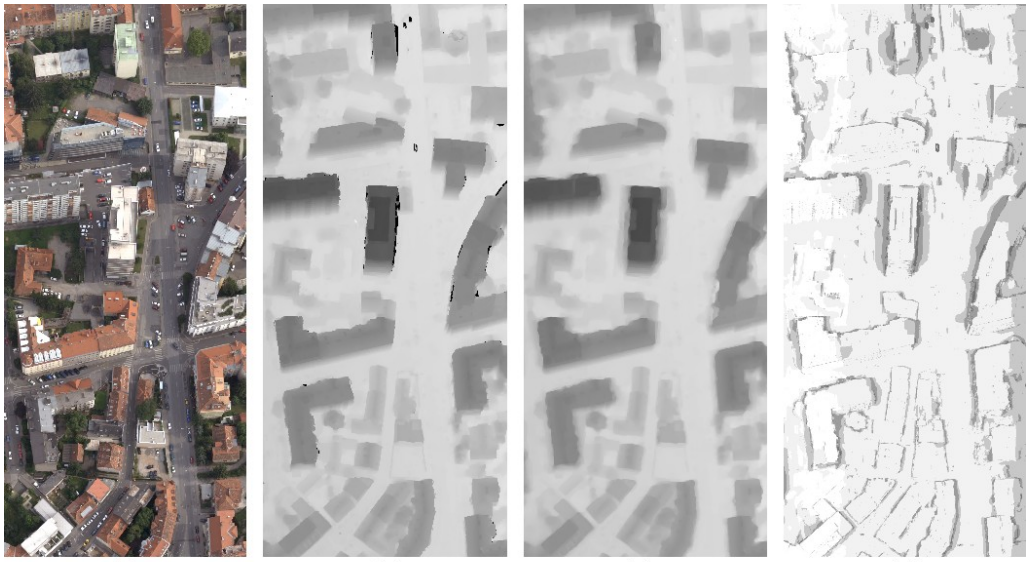

Figure 4.13: **Robust Triangulation and Confidence:** (a) A detail from the key view image from the Graz Jakomini aerial image sequence captured with Microsoft VEXCEL UltraCam. (b) Robust triangulation from linked correspondences shows crisp edges and reduced outliers. (c) Non-robust triangulation shows blurred depth discontinuities due to a missing outlier rejection scheme. (d) The confidence map shows areas of increased uncertainty with low intensity values. The lighter the areas in the confidence map, the more reliable correspondences were found.

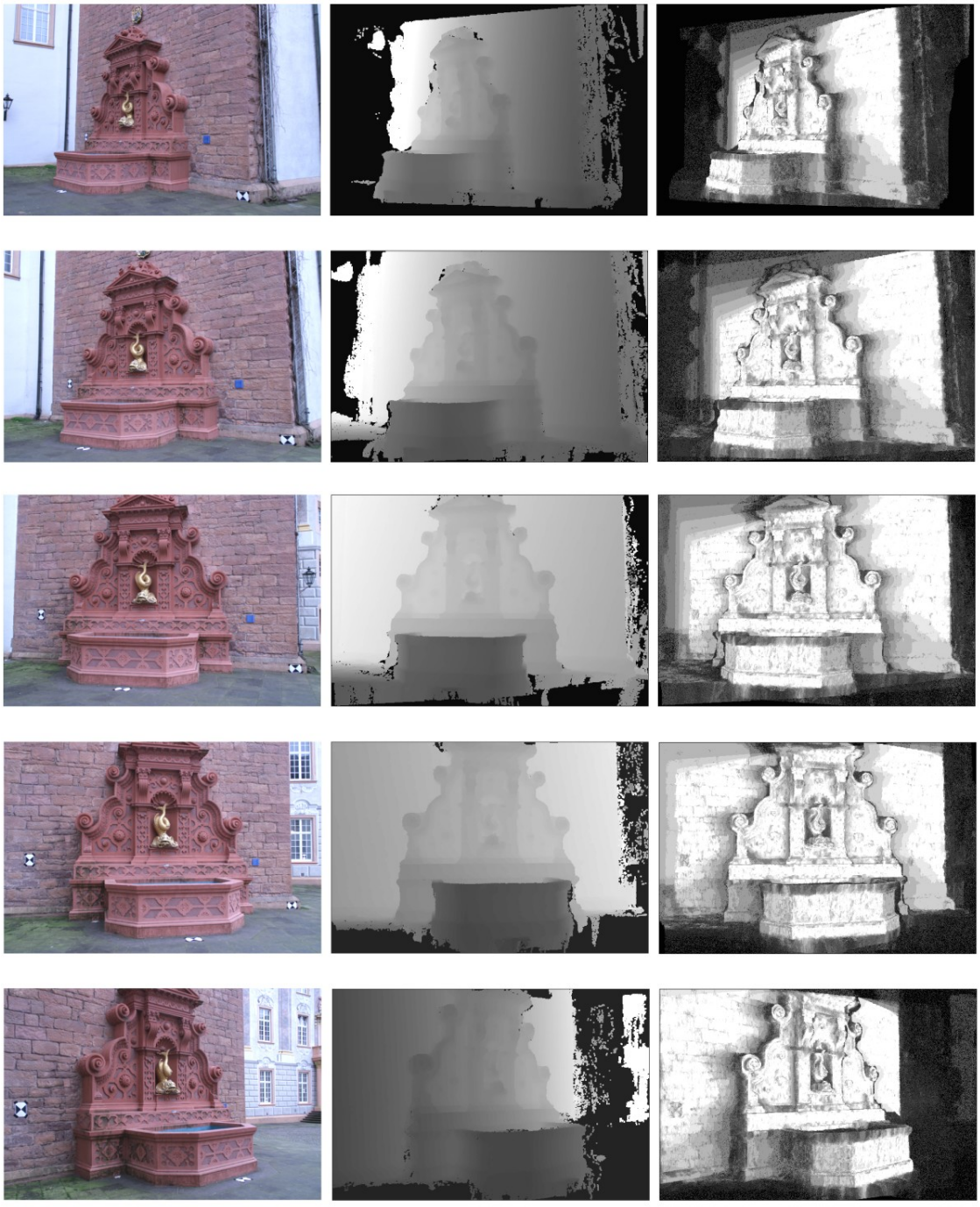

Figure 4.14: **The Fountain Dataset:** Results and confidence maps for five images of the fountain-P11 dataset from Strecha et al. [37] obtained from epipolar constrained TV-L^1 stereo with correspondence linking and robust triangulation.

Chapter 5

Experimental Results

Contents

5.1 Evaluation Methodology

We evaluate the quality of the obtained depth maps of our TV-L^1 stereo based multi-view reconstruction method compared to the plane sweep approach. In the quantitative evaluation, we provide error statistics for several images from the used datasets. Strecha et al. [37] provides multi-view datasets and a geometrical ground truth from Lidar acquisition, which allows us to compare our results on ground truth data.

We generated a set of reference depth maps (Figure 5.1) from the provided ground truth models for the fountain-P11 and the Herz-Jesu-P8 dataset for the evaluation.

We compute several error statistics for all pixels that are available in the reference depth maps from ground truth data. We focus on three quality measures in this evaluation [33].

1. The RMS (root mean square error) is measured in depth units between the ground truth d_r and the computed depth map d_c over the total number of N available pixels.

$$E = \sqrt{\frac{1}{N} \sum_{(x,y)} |d_r(x,y) - d_c(x,y)|^2} \tag{5.1}$$

Figure 5.1: **Reference Depth Maps from Lidar Ground Truth:** Reference depth maps generated from the ground truth models acquired by Lidar for one image of the fountain-P11 image sequence (left) and Herz-Jesu-P8 (right) from Strecha et al. [37].

2. The percentage of good matching pixels

$$P_{good} = \frac{1}{N} \sum_{(x,y)} (|d_r(x,y) - d_c(x,y)| < \delta_d) \tag{5.2}$$

or alternatively the percentage of bad matches, where δ_d is a depth error tolerance. We define δ_d in percent with respect to the scene's depth range. In our experiments we define a good match to lie within $+/-5\%$ of the scene's depth range around the reference depth.

3. The completeness of the scene is the percentage of estimated depths with respect to the total number of pixels available in the reference maps.

5.2 Quantitative Evaluation

5.2.1 Cost Functions under Varying Illumination Conditions

The performance of the cost functions at different window sizes is tested on an image sequence of ten images with radiometric differences. The images show a planar scene with a poster. The depth map results of our dense multi-view plane sweep reconstruction are matched with a reference plane which was retrieved by fitting a least squares plane to the reconstructed corner points of the poster.

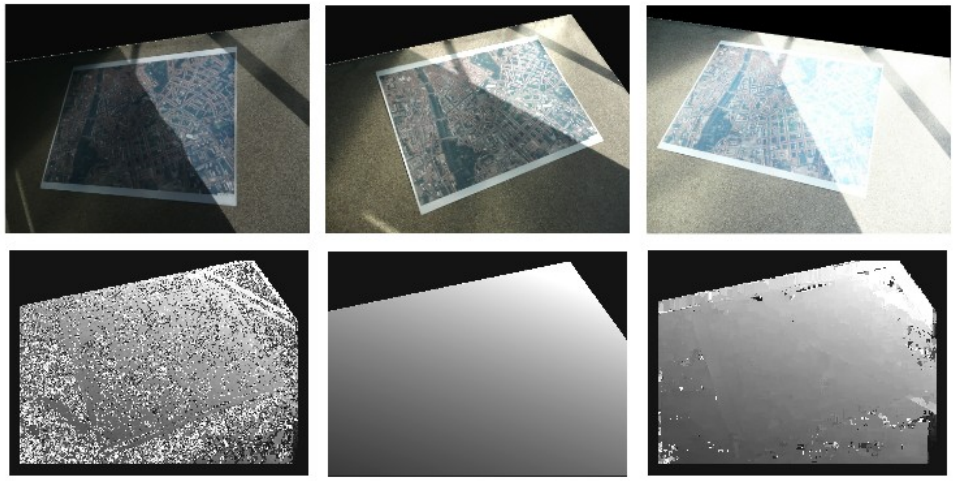

Figure 5.2: **Plane Dataset:** Top row: Three views out of ten from the plane poster dataset. The images are taken under varying illumination conditions. The middle image served as the reference view for the evaluation in Figure 5.3. Bottom row: The middle image is the reference depth map. The left image shows the winner-takes-all result for ZNCC with window radius 1 with many outliers, right ZNCC with radius 4 delivers a good reconstruction result for the plane but suffers from boundary overreach.

In Figure 5.3, we show a comparison of several cost functions for different window sizes from the plane sweep approach with winner-takes-all (WTA) depth extraction. The images in the plane dataset show heavy radiometric differences, hence the results for unnormalized cost functions (SAD and SSD) produce results that are inferior to their zero-mean normalized variants. The non-parametric census transform and the local daisy descriptor perform well for all window sizes. Throughout the best results can be obtained by ZNCC, moreover, it is able to be computed as the fastest among all normalized cost functions.

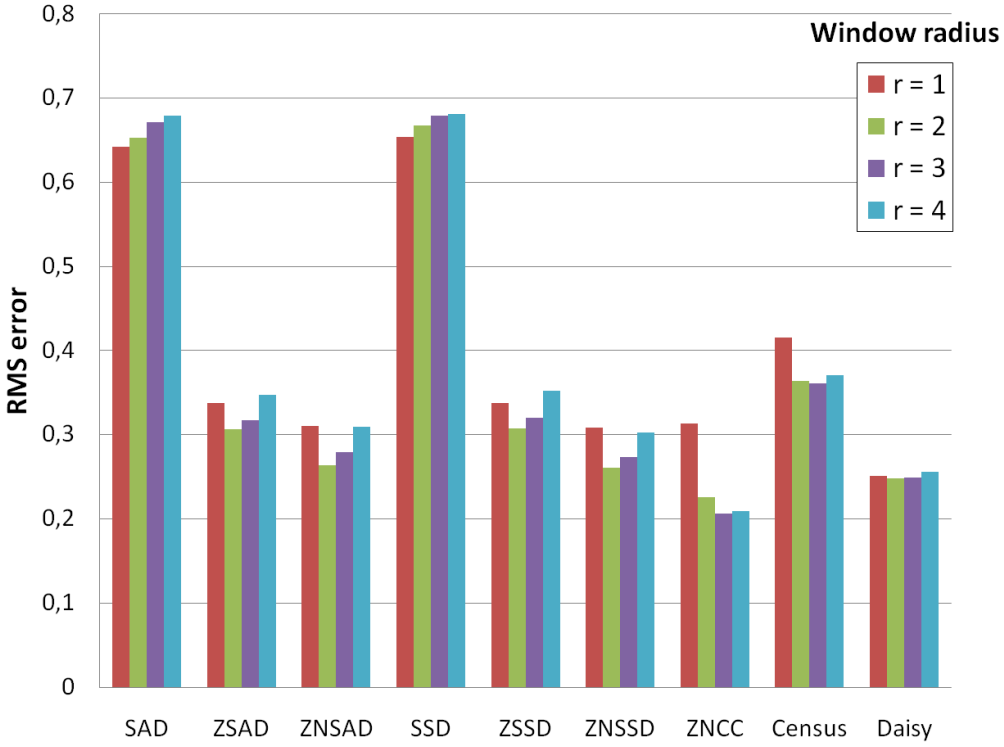

Figure 5.3: **Performance of Cost Functions:** Performance evaluation for diverse cost functions and different window sizes on the plane dataset. The images in the dataset show high radiometric differences.

5.2.2 Comparison between Local and Global Methods

Figure 5.4 provides a comparison of the results from plain local matching with either a winner-takes-all (WTA) or median depth extraction strategy and results from global optimization obtained from the plane sweep multi-label approach and the TV-L^1 stereo based method. We compare winner-takes-all and median depth cost extraction from ZNCC matched images without global optimization to ZNCC and ZNSAD matched depths after global optimization and the TV-L^1 stereo method (flow).

Global methods clearly outperform plain local matching techniques. The influence of the selected cost function is evident, nevertheless negligible after global optimization. The results are more influenced by the parameter λ, which determines the degree of smoothness in the global optimization. The same applies for the window radius, hence it is set predominantly to r=1 in the experiments. The TV-L^1 stereo based robust multi-view reconstruction provides the overall best results on the tested dataset.

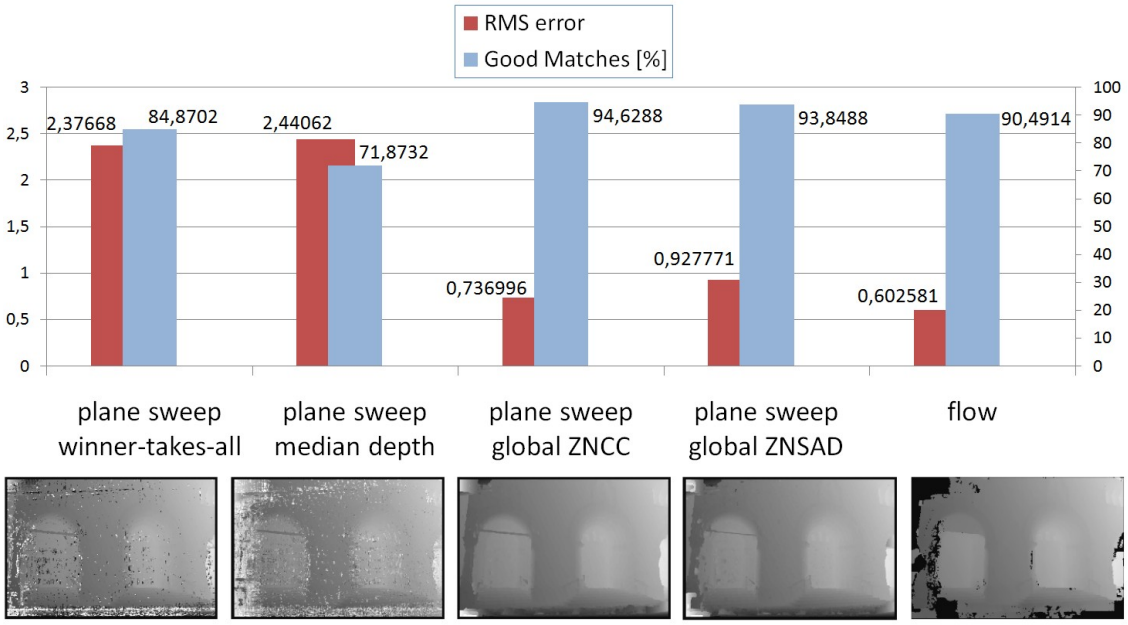

Figure 5.4: **RMS Errors:** A comparison of plane sweep matching with and without global optimization and epipolar constrained TV-L^1 stereo based multi-view reconstruction for one image of the Herz-Jesu-P8 sequence.

5.2.3 Influence of Wide Baselines on TV-L^1 matching

The baseline is a crucial point for optical flow based matching. The quality of measurements between pairs of views with large baseline decreases rapidly. We suggest to link correspondences over wider baselines and robust triangulation to obtain reliable reconstruction results. We illustrate the effect of wide baseline matching on the reconstruction and present a set of depth maps in Figure 5.5. View number five serves as the key view that is matched with all its ten neighbors.

The values for view number five in the diagram corresponds to the combined multi-view depth map. Only the adjacent correspondences from view four and six were used directly. Measurements from all other views were added view per view through correspondence linking, described in Section 4.3.4.

We define the number of supporting measurements (inliers) used in robust triangulation as the confidence value for a depth and encode it in the confidence map, shown for view number five in Figure 5.6. The higher the number of outliers, i.e. in occluded areas and image regions with insufficient overlap, the higher the uncertainty for the according pixel.

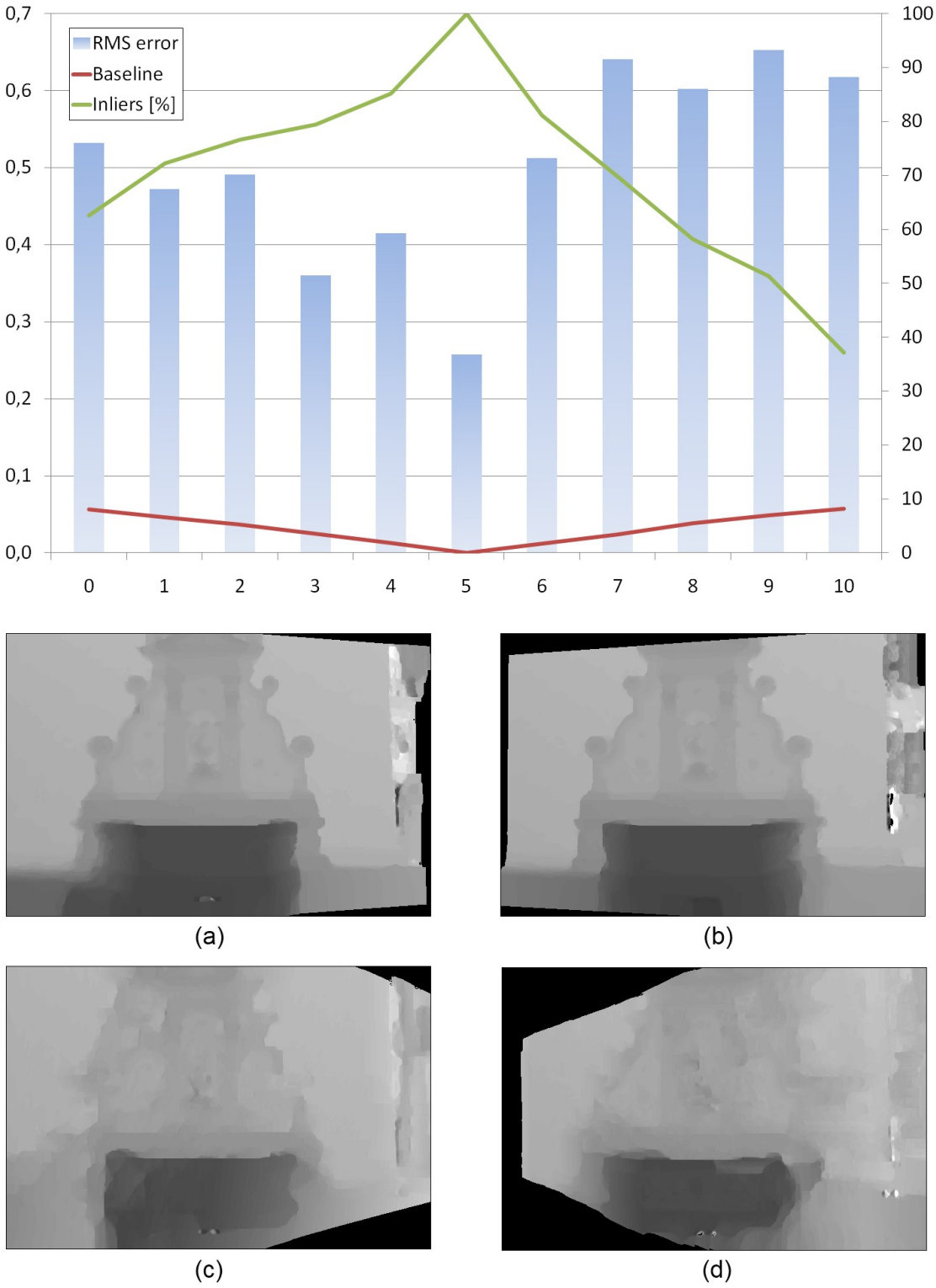

Figure 5.5: **Error per View and Baseline:** The quality of the obtained pairwise depth maps from optical flow decreases with baseline. The diagram shows for all pairs the RMS error and the number of selected inliers used for robust triangulation. View number five serves as the key view (cf. Figure 5.6). The depth maps (a) and (b) were obtained with the direct neighbors of the key view. The images (c) and (d) illustrate the poor results obtained with flow estimation between the key view and the two outermost images over wide baselines.

Figure 5.6: **Confidence and Depth Map:** The confidence map (a) shows the number of selected inliers for robust triangulation of the depth map (b). Image overlap of the input images results in a higher number of available measurements, areas with occlusions and unreliable measurements reduce the confidence.

5.2.4 Error Statistics for Plane Sweep and TV-L^1 matching

Error statistics including RMS error and completeness of the obtained depth maps are summarized in Table 5.1. We compare results from plane sweep with SAD, ZNSAD and ZNCC matching and global optimization to epipolar constrained TV-L^1 stereo based reconstruction. Evaluation is done for one image of the fountain-P11 and Herz-Jesu-P8 sequence. We used a depth map obtained from a sparse reconstruction for initialization for the flow method as it was described in Chapter 4.

		flow	plane sweep		
			SAD	ZNSAD	ZNCC
fountain-P11	RMS error	0.257	0.71454	0.540	0.421878
	completeness [%]	93.055	94.7247	94.658	94.6586
Herz-Jesu-P8	RMS error	0.602	0.95931	0.927	0.736
	completeness [%]	88.499	93.9321	93.932	93.932

Table 5.1: **Error Statistics:** All experiments and evaluations on the fountain-P11 and Herz-Jesu-P8 datasets were performed on an image resolution of 645x430 pixels. The flow reconstruction method was initialized with a depth estimate from the sparse points. Parameters for TV-L^1 matching: $\lambda = 0.15$, warps=5, iterations=100. Parameters for plane sweep: $\lambda = 100, t = 0.17$ (SAD, ZNSAD) and $\lambda = 20, t = 0.5$ (ZNCC).

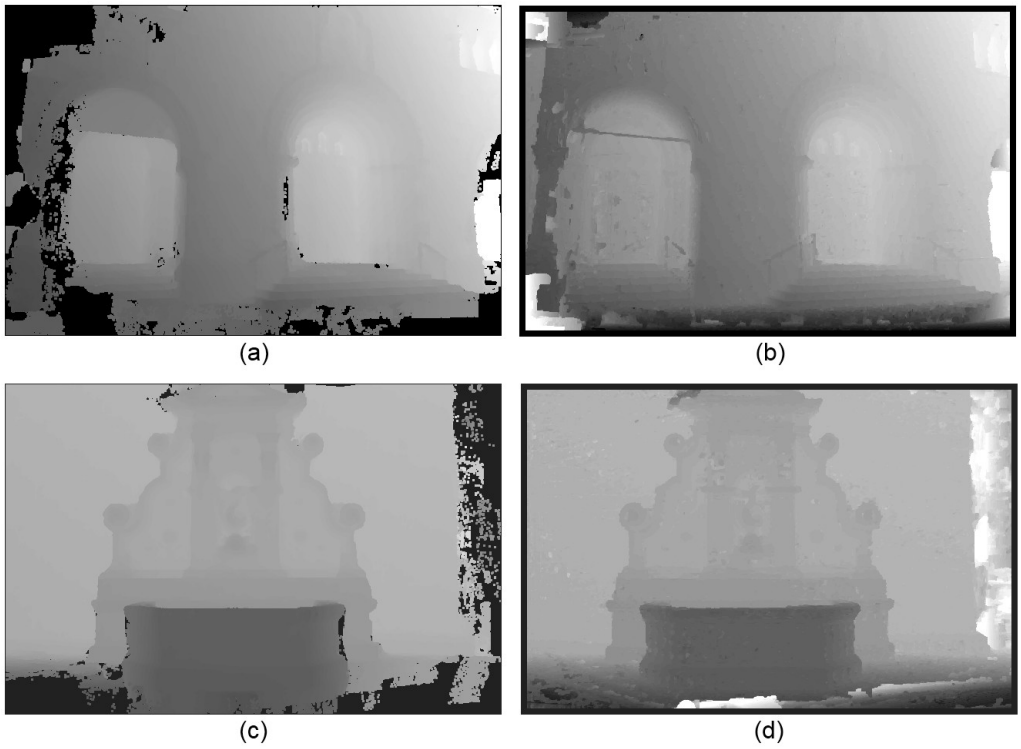

Figure 5.7: **Global Optimization Plane Sweep and Flow:** Comparison between TV-L^1 matching (a) and plane sweep with ZNSAD matching after global optimization (b) for the Herz-Jesu-P8 dataset. Images (c) and (d) illustrate the results for one image of the fountain-P11 dataset respectively.

5.3 Qualitative Comparison

Here we provide a few qualitative comparisons between the two presented approaches on aerial images from the Jakomini sequence. Since no ground truth data is available, our examination remains a visual inspection of the results. Figure 5.8 shows a comparison of TV-L^1 stereo based matching and robust triangulation in contrast to the globally optimized plane sweep depth map obtained with the SAD matching cost. The TV-L^1 stereo shows crisper edges than the plane sweep approach but does contain holes (e.g. in occluded regions) due to the outlier rejection during robust triangulation.

Figure 5.9 provides a 3D view of a single reconstructed depth map computed from two adjacent views of the Middlebury Dino Sparse Ring dataset [34].

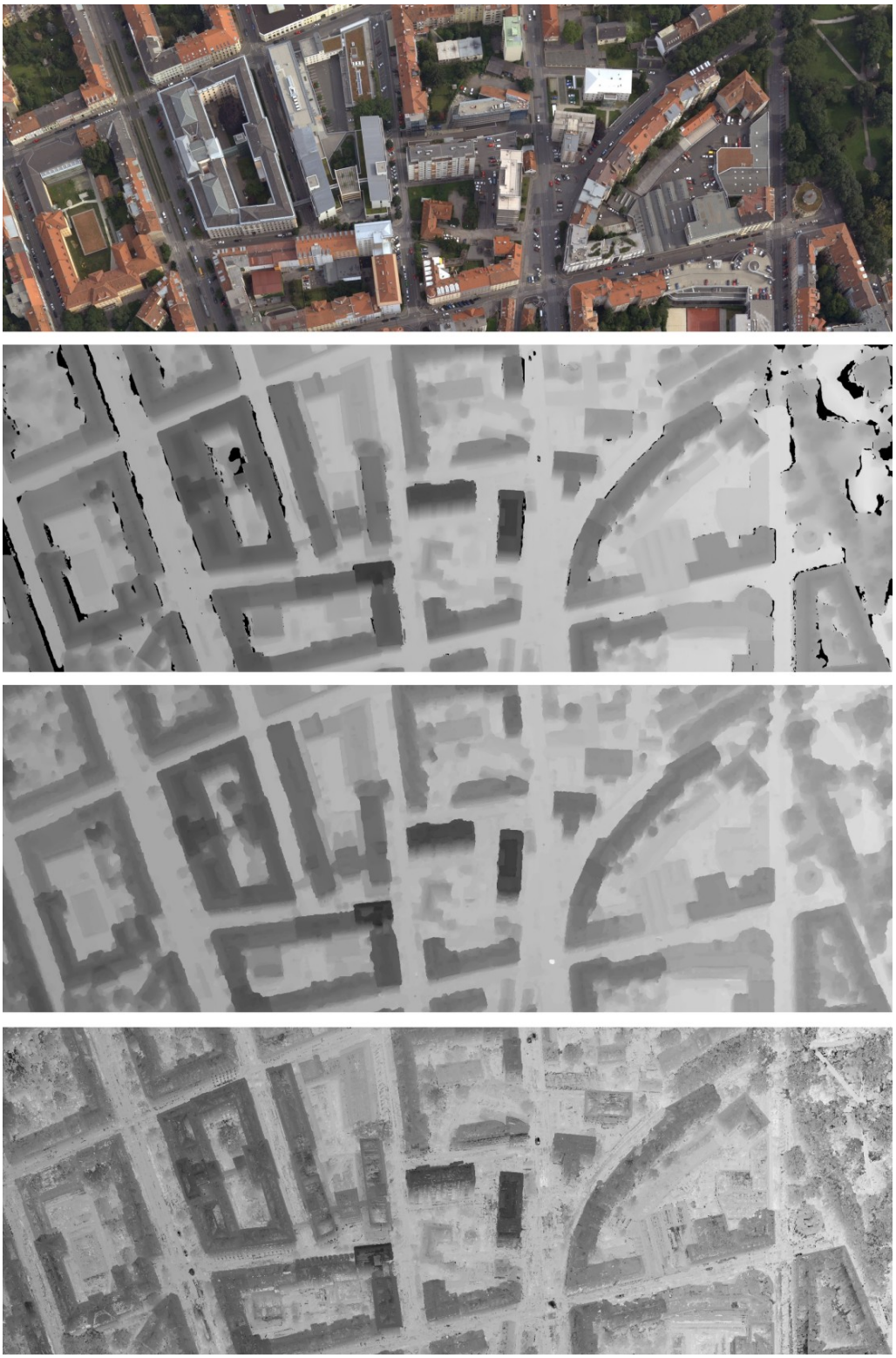

Figure 5.8: **Qualitative Comparison on Aerial Images:** From top to bottom: key view image; depth map from epipolar constrained TV-L^1 optical flow and robust triangulation; Result from plane sweep with SAD matching and global optimization; plane sweep winner-takes-all.

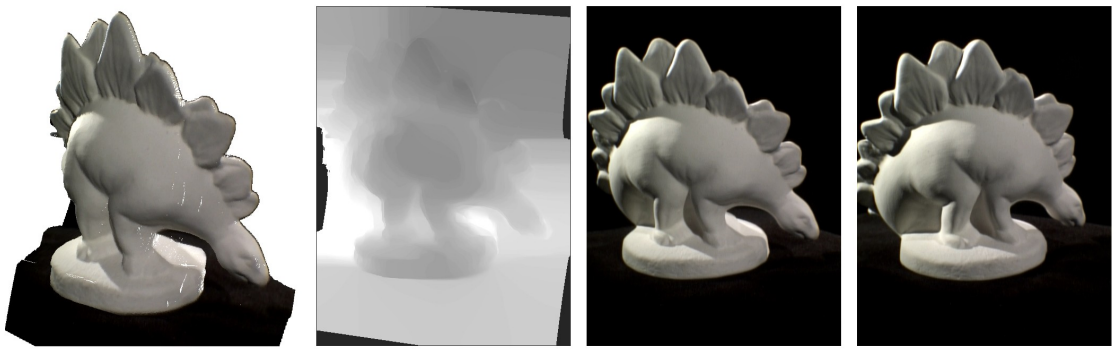

Figure 5.9: **Depth Map Visualization:** One depth map from the Dino Sparse Ring sequence reconstructed from two adjacent views.

Finally, we illustrate the results of our 3D reconstruction for the complete fountain-P11 and Herz-Jesu-P8 sequence as colored point cloud models (Figure 5.10 and 5.11) in addition to the quantitative evaluation above. The depth maps are unprojected to 3D points into a common coordinate frame and colored according to their key view pixels.

Figure 5.10: **3D Point Cloud Reconstruction fountain-P11:** 3D reconstruction as colored point clouds for the complete image sequence of the fountain-P11 datasets. The picture demonstrates the camera setup reconstructed from SfM.

Figure 5.11: **3D Point Cloud Reconstruction Herz-Jesu-P8:** 3D reconstruction as colored point clouds for the complete image sequence of the Herz-Jesu-P8 datasets.

Chapter 6

Conclusion

Contents

6.1 Summary

3D reconstruction is an active field of research in computer vision. Several methods from active stereo to laser measurements (Lidar) for recovering the shape of objects have been studied over the last decades. While the former is able to determine 3D coordinates in real time under controlled conditions, it is mainly suited for indoor environments. The latter, though suited for outdoor scenes, demands complex methods and time consuming methods for large scenes and causes high costs in particular when aerial acquisition is required.

As an alternative, image-based reconstruction techniques allow portability, flexibility and low-costs [31]. The availability of cheap digital cameras and massive computational power of programmable graphics hardware additionally boost the development of algorithms for generating 3D models.

While the geometric relations of multiple views, camera calibration and reconstruction of sparse models through Structure-from-Motion [17] is well understood, dense reconstruction yet poses a few challenges. The main difficulty is to determine reliable correspondences

between the views for all image points that are visible in the neighboring images, since the appearance of scene points may vary with viewpoint and changing illumination conditions.

Multiple views aid in the correspondence problem and contribute to scene completeness of otherwise occluded areas.

Plane sweep is one method that allows reconstruction from arbitrary collections of views. A local matching and cost aggregation step fills a cost volume from which the final depth values can be extracted using global optimization techniques. All steps of this high-performance reconstruction algorithm are suited to be performed on modern graphics hardware. We investigated the influence of different matching cost functions on the results.

The disadvantages of the plane sweep method are its time and memory consuming depth extraction step and the need for setting user-specified parameters. A new technique based on total variation based optical flow is able to overcome these limitations. We integrated the epipolar constraint into the TV-L^1 optical flow [29, 50] to reduce the search for correspondences to a one-dimensional problem.

A method for correspondence linking for wide baseline reconstruction is suggested. RANSAC based robust triangulation is used to reject outliers due to occlusions in the multi-view reconstruction step.

6.2 Discussion and Future Work

The experiments have proven that we are able to produce depth maps with similar and in most cases even better quality with our TV-L^1 stereo based multi-view reconstruction approach in contrast to the plane sweep multi-label method.

The results are even better and computation time faster (i.e. due to a reduced number of pyramid levels in the coarse-to-fine framework) if a good depth estimate is used for initialization. Relating to that, the quality of the computed depth maps depends more on the initialization than on the parameters for λ or the number of warping steps.

This approves that our design is able to facilitate the transition from a semi automatic to a fully automatic reconstruction pipeline. The perspective is to put a set of images into the pipeline and get a complete 3D model out, with no user guidance.

Very good results can be obtained from optical flow matching between views with small baselines. Wider baselines pose a severe problem for correspondence matching due to an increasing amount of occluded areas. This necessitates additional techniques for acquiring more measurements than those of the nearest neighboring views for better triangulation angles and robust reconstruction with respect to outliers.

The suggested method for correspondence linking offers the possibility to add correspondences from views with wider baseline to the actual key view. Since outliers are detected at the end in the reconstruction process during triangulation, we are unable to decide whether a direct correspondence is reliable enough for linking or not. If a bad correspondence is used, the error propagates over all links and corrupts the correct depth estimation at this pixel.

The only way of detecting bad measurements is during the triangulation and outlier rejection stage. Obviously, a flaw of this method is the fact that the depth maps may then contain holes.

Future work encompasses besides a fast GPU implementation an improved initialization scheme. A better way of initialization promises an improvement for the algorithm's performance.

Recent advances in 3D reconstruction algorithms yielded to live dense reconstruction with a single moving camera, as proposed by Newcombe et al. [27]. The authors use a rough surface fitted into sparse points from PTAM [24] and assign interpolated initial depth values according to radial basis functions (RBFs).

Future work might include an occlusion handling policy, which is able to detect them in an earlier stage of the reconstruction process rather than at the very end.

A possible extension of the optical flow method to account for different cost functions in the data term would be of further interest. The influence of window-based measures compared to the pixel-based intensity difference may be investigated.

Bibliography

[1] Ansar, A., Castano, A., and Matthies, L. (2004). Enhanced real-time stereo using bilateral filtering. In *3DPVT '04: Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium*, pages 455–462.

[2] Asmaa Hosni, M. B. and Gelautz, M. (2010). Near real-time stereo with adaptive support weight approaches. In *3DPVT*.

[3] Aujol, J.-F., Gilboa, G., Chan, T., and Osher, S. (2006). Structure-texture image decomposition–modeling, algorithms, and parameter selection. *nternational Journal of Computer Vision*, 67:111–136.

[4] Azad, P., Gockel, T., and Dillmann, R. (2007). *Computer Vision – Das Praxisbuch*. Elektor-Verlag, ISBN: 3895761656.

[5] Beder, C. and Steffen, R. (2006). Determining an initial image pair for fixing the scale of a 3d reconstruction from an image sequence. In *Annual Symposium of the German Association for Pattern Recognition (DAGM)*, pages 657–666.

[6] Birchfield, S. and Tomasi, C. (1998). A pixel dissimilarity measure that is insensitive to image sampling. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20:401–406.

[7] Bradski, G. and Kaehler, A. (2008). *Learning OpenCV*. O'Reilly Media, ISBN: 0596516134.

[8] Brown, M. Z., Burschka, D., Hager, G. D., and Member, S. (2003). Advances in computational stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

[9] Calonder, M., Lepetit, V., and Fua, P. (2010). BRIEF: Binary Robust Independent Elementary Features. In *11th European Conference on Computer Vision (ECCV)*.

[10] Chambolle, A. (2004). An algorithm for total variation minimization and applications. *Journal of Mathematical Imaging and Vision*, 20:89–97.

[11] Cornelis, N. and Van Gool, L. (2005). Real-time connectivity constrained depth map computation using programmable graphics hardware. In *Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*, pages 1099–1104.

[12] Faugeras, O. and Keriven, R. (1999). Variational principles, surface evolution, pde's, level set methods and the stereo problem. *IEEE Transactions on Image Processing*, 7(3):336–344.

[13] Furukawa, Y. and Ponce, J. (2010). Accurate, dense, and robust multiview stereopsis. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 32:1362–1376.

[14] Gallup, D., Frahm, J.-M., and Pollefeys, M. (2008). Variable baseline/resolution stereo. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[15] Gerrits, M. and Bekaert, P. (2006). Local stereo matching with segmentation-based outlier rejection. *Proc. Conf. Computer and Robot Vision*.

[16] Goesele, M., Snavely, N., Curless, B., Hoppe, H., and Seitz, S. M. (2007). Multi-view stereo for community photo collections. *Computer Vision, IEEE International Conference on*.

[17] Hartley, R. I. and Zisserman, A. (2004). *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition.

[18] Hermann, S. and Klette, R. (2009). The naked truth about cost functions. *Multimedia Imaging Report 33, The University of Auckland, New Zealand*.

[19] Hiep, V. H., Keriven, R., Labatut, P., and Pons, J.-P. (2009). Towards high-resolution large-scale multi-view stereo. In *CVPR*, pages 1430–1437. IEEE.

[20] Hirschmüller, H. and Scharstein, D. (2009). Evaluation of stereo matching costs on images with radiometric differences. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 31(9):1582–1599.

[21] Horn, B. K. and Schunck, B. G. (1981). Determining optical flow. *Artificial Intelligence*, 17:185–203.

[22] Irschara, A., Zach, C., and Bischof, H. (2007). Towards wiki-based dense city modeling. In *Proceedings of the Eleventh IEEE International Conference on Computer Vision*.

[23] Kang, S. B., Szeliski, R., and Chai, J. (2001). Handling occlusions in dense multi-view stereo. In *Proceedings of the 2001 IEEE Computer Society Conference on Computer Vision and Pattern Recognition. CVPR*, pages 103–110.

[24] Klein, G. and Murray, D. (2007). Parallel tracking and mapping for small AR workspaces. In *Proc. Sixth IEEE and ACM International Symposium on Mixed and Augmented Reality (ISMAR'07)*.

[25] Leberl, F., Irschara, A., Pock, T., Meixner, P., et al. (2010). Point clouds: Lidar versus 3d vision. *Photogrammetric Engineering and Remote Sensing, Vol. 76, No. 10*, pages 1123–1134.

[26] Lowe, D. G. (2004). Distinctive image features from scale-invariant keypoints. *International Journal of Computer Vision*, 60:91–110.

[27] Newcombe, R. A. and Davison, A. J. (2010). Live dense reconstruction with a single moving camera. In *CVPR*, pages 1498–1505.

[28] Okutomi, M. and Kanade, T. (1993). A multiple-baseline stereo. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 15:353–363.

[29] Pock, T. (2008). *Fast Total Variation for Computer Vision, PhD*. Graz University of Technology, Austria.

[30] Pock, T., Schoenemann, T., Graber, G., Bischof, H., and Cremers, D. (2008). A convex formulation of continuous multi-label problems. In *ECCV'08*, pages 792–805.

[31] Remondino, F. and El-Hakim, S. (2006). Image-based 3d modelling: A review. *The Photogrammetric Record*, 21(115):269–291.

[32] Rudin, L. I., Osher, S., and Fatemi, E. (1992). Nonlinear total variation based noise removal algorithms. *Phys. D*, 60:259–268.

[33] Scharstein, D. and Szeliski, R. (2002). A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47:7–42.

[34] Seitz, S. M., Curless, B., Diebel, J., Scharstein, D., and Szeliski, R. (2006). A comparison and evaluation of multi-view stereo reconstruction algorithms. In *Proceedings of the 2006 IEEE Computer Society Conference on Computer Vision and Pattern Recognition - Volume 1*, pages 519–528.

[35] Slesareva, N., Bruhn, A., and Weickert, J. (2005). Optic flow goes stereo: A variational method for estimating discontinuity-preserving dense disparity maps. In *Proc. 27th DAGM Symposium*, pages 33–40.

[36] Sonka, M., Hlavac, V., and Boyle, R. (1998). *Image Processing, Analysis, and Machine Vision*. Brooks/Cole Publishing, ISBN: 053495393X.

[37] Strecha, C., von Hansen, W., Van Gool, L., Fua, P., and Thoennessen, U. (2008). On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*.

[38] Szeliski, R. and Zabih, R. (1999). An experimental comparison of stereo algorithms. In *Vision Algorithms: Theory and Practice, number 1883 in LNCS*, pages 1–19. Springer-Verlag.

[39] Tola, E., Lepetit, V., and Fua, P. (2008). A fast local descriptor for dense matching. *Conference on Computer Vision and Pattern Recognition*.

[40] Tomasi, C. and Manduchi, R. (1998). Bilateral filtering for gray and color images. In *Proceedings of the Sixth International Conference on Computer Vision*, ICCV '98. IEEE Computer Society.

[41] Tombari, F., Mattoccia, S., Stefano, L. D., and Addimanda, E. (2008). Classification and evaluation of cost aggregation methods for stereo correspondence. *IEEE Conference on Computer Vision and Pattern Recognition*.

[42] Vedaldi, A. (2007). An open implementation of the SIFT detector and descriptor. Technical Report 070012, UCLA CSD.

[43] Vogiatzis, G., Hernández Esteban, C., Torr, P. H. S., and Cipolla, R. (2007). Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29:2241–2246.

[44] Wedel, A., Pock, T., Braun, J., Franke, U., and Cremers, D. (2008a). Duality tv-l1 flow with fundamental matrix prior. In *Image and Vision Computing New Zealand, 2008. IVCNZ 2008. 23rd International Conference*.

[45] Wedel, A., Pock, T., Zach, C., Bischof, H., and Cremers, D. (2008b). An improved algorithm for tv-l1 optical flow computation. In *Proceedings of the Dagstuhl Visual Motion Analysis Workshop*.

[46] Yang, R. and Pollefeys, M. (2003). Multi-resolution real-time stereo on commodity graphics hardware. In *Proceedings of IEEE Computer Society Conference on Computer Vision and Pattern Recognition*.

[47] Yoon, K.-J. and Kweon, I.-S. (2005). Locally adaptive support-weight approach for visual correspondence search. In *Computer Vision and Pattern Recognition*, pages 924–931.

[48] Zabih, R. and Woodfill, J. (1994). Non-parametric local transforms for computing visual correspondence. In *Proceedings of the third European conference on Computer Vision (Vol. II), ECCV '94*, pages 151–158.

[49] Zabulis, X. and Daniilidis, K. (2004). Multi-camera reconstruction based on surface normal estimation and best viewpoint selection. In *Proceedings of the 3D Data Processing, Visualization, and Transmission, 2nd International Symposium*, pages 733–740. IEEE Computer Society.

[50] Zach, C., Pock, T., and Bischof, H. (2007). A duality based approach for realtime tv-l1 optical flow. In *Proceedings of the 29th DAGM Symposium on Pattern Recognition*, pages 214–223.

[51] Zach, C., Sormann, M., and Karner, K. (2006). High-performance multi-view reconstruction. In *International Symposium on 3D Data Processing, Visualization and Transmission (3DPVT)*.

[52] Zhang, L. and Seitz, S. M. (2001). Image-based multiresolution shape recovery by surface deformation. In *Proceedings of SPIE: Videometrics and Optical Methods for 3D Shape Measurement*.