Direct Disparity Space: Robust and Real-time Visual Odometry

Hatem Alismail and Brett Browning
halismai@cs.cmu.edu brettb@cs.cmu.edu

CMU-TR-RI-14-20

October, 2014

Robotics Institute
Carnegie Mellon University
Pittsburgh, PA 15213

This publication was made possible by NPRP grant #09-980-2-380 from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

Keywords: visual odometry, stereo, disparity space, pose estimation

Abstract

We present a direct visual odometry formulation using a warping function in disparity space. In disparity space measurement noise is well-modeled by a Gaussian distribution, in contrast to the heteroscedastic noise in 3D space. In addition, the Jacobian of the warp separates the rotation and translation terms, enabling motion to be estimated from all image points even those located at infinity. Furthermore, we show that direct camera tracking can obtain accurate and robust performance using only a fraction of the image pixels through a simple and efficient pixel selection strategy. Our approach allows faster than real-time computation on a single CPU core with unoptimized code.

As our approach does not rely on feature extraction, the selected pixels over successive frames are often unique. Hence, triangulating the selected pixels to the world frame produces an accurate and dense 3D reconstruction with minimal computational cost making it appealing to robotics and embedded applications.

We evaluate the performance of our approach against state-of-the-art methods on a range of urban and indoor datasets. We show that our algorithm produces competitive performance, requires no specialized tuning, and continues to produce competitive results even when run with low resolution images where other techniques fail to operate.

Contents

1 Introduction

Visual Odometry (VO) is the problem of estimating the relative pose between two cameras sharing a common field of view. The problem is fundamental in robotics and computer vision with many solutions in the literature [Scaramuzza and Fraundorfer, 2011]. Pose estimation based on the structure-from-motion (SFM) pipeline [Torr and Zisserman, 2000] is perhaps the most common stereo VO framework for robotic applications [Nister et al., 2004]. In the SFM pipeline, feature points are extracted from consecutive images at sub-pixel positions [Valkenburg et al., 1994] and matched using their associated descriptors. Estimates of pose can be obtained using 2D-3D pose estimation algorithms [Haralick et al., 1994] embedded in a robust estimation framework [Fischler and Bolles, 1981]. Due to the local nature of VO, drift in pose is unbounded (typically super-linear [Olson et al., 2001]). Direct can be reduced via nonlinear optimization over a batch of frames by minimizing the reprojection error [Triggs et al., 2000].

An alternative to feature-based algorithms is the direct method [Irani and Anandan, 2000; Baker and Matthews, 2004; Horn and Schunck, 1981]. In direct methods, pixel intensities are used *directly* in the estimation without resorting to features as an intermediate representation of the image. This is accomplished by estimating the parameters of the motion between consecutive frames such that a function of pixel intensity dissimilarity is minimized.

At the core of direct methods are two assumption: (i) the *brightness constancy* assumption – stating that the brightness/intensity of a pixel remains constant as the corresponding surface in space, or camera, moves – and (ii) the displacements of pixels are infinitesimally small. Brightness constancy is commonly relaxed by modeling appearance variations; *e.g.* using a linear model [Gennert and Negahdaripour, 1987]. The small pixel displacements assumption remains fundamental as direct methods rely on gradients. This requirement is commonly mitigated by implementing the algorithm in scale space [Lindeberg, 1994], or via providing an appropriate initialization. An alternative solution is using a high camera frame-rate such that the expected 3D motion of the cameras results in small pixel displacements in the image plane [Handa et al., 2012].

Recently, with the introduction of the Kinect, many researchers have exploited the high frame-rate of intensity and depth images for designing direct methods for RGB-D pose estimation [Klose et al., 2013; Kerl et al., 2013; Steinbrucker et al., 2011; Henry et al., 2012; Whelan et al., 2013], as well as VO and VSLAM systems from monocular and stereo data [Forster et al., 2014; Engel et al., 2014; Comport et al., 2010]. The main advantages of a direct method in RGB-D systems are robustness, speed and the ability to obtain a dense 3D model of the en-

vironment in real-time [Steinbrucker et al., 2013; Audras et al., 2011; Sturm et al., 2013; Fang and Scherer, 2014].

Algorithms for RGB-D VO estimate the parameters of a rigid-body transformation such that the intensity values of warped 3D points from the previous frame match those of the current. The problem is usually posed as a compositional image alignment [Baker and Matthews, 2004] and solved by linearizing the cost function in an iteratively re-weighted least squares (IRLS) optimization [Black and Anandan, 1993; Wolke and Schwetlick, 1988]. Two assumptions must be satisfied for the linearization to be valid. First, the magnitude of the estimated parameter vector must be small. Second, and critical to the accuracy of the approximation, is that the measurement noise model is correctly accounted for. In most RGB-D VO systems, a Guassian noise model is assumed. This is justified in practice as the usable depth values are restricted to close range. Hence, approximating close-range triangulation errors with a Gaussian distribution is reasonable.

For stereo VO in unstructured environments, the Gaussian noise assumption does not hold, especially for distal observations [Matthies and Shafer, 1987]. Comport et al. [2010] propose a direct VO method that does not require depth from stereo. This is accomplished by using a decomposition of the quadrifocal tensor to eliminate depth entirely and instead to estimate the pose of the stereo rig from the four views. The algorithm is initialized with a specially tuned stereo algorithm for urban scene. The authors do not discuss handling of points at infinity, which are common in most outdoor scenes. Tykkälä et al. [2011] describe a direct VO approach from stereo images incorporating photometric and geometric terms, but it was demonstrated on scenes with constrained, close-range depth and relies on high quality stereo [Hirschmuller, 2005].

In this work, we develop an algorithm for direct VO from stereo data using a warping function in *disparity space* [Demirdjian and Darrell, 2001]. The algorithm is suitable for a variety of environments without restrictive assumptions. In contrast to Comport et al. [2010], our algorithm is simple to implement and only requires two temporal images and a disparity map. Furthermore, we require only the most basic of stereo algorithms. Our algorithm has the following properties:

- Automated parameter tuning by design. Parameter settings are robust to different environments and stereo rigs.
- Deterministic performance (as opposed to sampling based methods Fischler and Bolles [1981]), which is desirable in critical systems.

- Applicable to different sensors (stereo, RGB-D, *etc.*).

- Efficient and amenable to parallelization (GPU, FPGA, *etc.*). Our unoptimized implementation runs in real-time on a CPU.

- Exploits points at infinity with no specialization; All pixels with non vanishing gradient may be used to estimate the rotation of the camera.

- Robust and capable of accurate estimation of pose even from thumbnail-sized stereo images. This is particularly useful in embedded systems and space robots Howard [2008].

Traditionally, direct methods have been *dense* in the sense that they use all pixel information. In this work, we show that direct methods can achieve the same level of accuracy and robustness by using only a fraction of the image pixels. We implement a simple, and efficient pixel selection strategy that greatly improves computational efficiency without degrading accuracy. Indeed, for high frame-rate data our algorithm achieves $100+$ Hz, excluding stereo computation, on VGA sized images on a single CPU core.

The nature of the pixel selection process also means that we can produce visually pleasing 3D point cloud reconstructions that are sufficiently dense for various robotic perception tasks without additional computation. As pixels are selected independently for each frame pair, simply triangulating these pixels and transforming them to the world frame accumulates a dense point cloud quickly (see section 4.5).

We will describe the algorithm in section 3. In the next section, we briefly review the disparity space formulation and dense image alignment.

2 Preliminaries & Notation

2.1 Disparity space

Consider a rectified stereo image with baseline B and an upper triangular camera intrinsic matrix composed of the camera focal length f and the principle point $\mathbf{c} = (c_u, c_v)^\top$. Without loss of generality, let the left image be the origin of the coordinate system. A point $\mathbf{p} = (x, y, d, 1)^\top$ is an element of disparity space, where $x = u - c_u$, $y = v - c_v$ and $d = u - u_r$ is the disparity; the difference between the u-coordinate in the left image and its corresponding coordinate in the right image. Given this rectified stereo, the depth of an image point can be obtained with $Z = Bf/d$.

Consider two stereo pairs related via a rigid body transformation $\mathbf{T}(\boldsymbol{\theta}) \in SE(3)$ parameterized by $\boldsymbol{\theta} \in \mathbb{R}^p$, where p is typically 6, such that a 3D point $\mathbf{X} = (X, Y, Z, 1)^\top$ is transformed into $\mathbf{X}' = \mathbf{T}(\boldsymbol{\theta})\mathbf{X}$. This rigid-body motion relationship may be expressed in disparity space as

$$\mathbf{p}' \equiv \boldsymbol{\Gamma}\mathbf{T}(\boldsymbol{\theta})\boldsymbol{\Gamma}^{-1}\mathbf{p}, \tag{1}$$

where $\equiv$ denotes projective equality up-to-scale, and $\boldsymbol{\Gamma}$ is a 4×4 matrix that depends on the known stereo calibration and is given by:

$$\boldsymbol{\Gamma} = \begin{pmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 0 & fB \\ 0 & 0 & 1 & 0 \end{pmatrix}. \tag{2}$$

Demirdjian and Darrell [2001] analyze the disparity space and show that it is a *projective space* $\subset \mathbb{P}^3$ with the important property that the measurement noise of $\mathbf{p}$ is well-approximated with a Gaussian distribution.

2.2 Direct Visual Odometry

Let the intensity of a point $\mathbf{p}$ at the *reference* frame be given with $\mathbf{I}(\tilde{\mathbf{p}}) \in \mathbb{R}$, where $\tilde{\mathbf{p}} = (x + c_u = u, y + c_v = v)^\top$. With an abuse of notation, we well use $\mathbf{I}(\mathbf{p}) \coloneqq \mathbf{I}(\tilde{\mathbf{p}})$.

After a rigid-body motion with $\mathbf{T}(\boldsymbol{\theta})$, we obtain the *input* image $\mathbf{I}'(\mathbf{p}')$. Given an initialization $\boldsymbol{\theta}$, we seek to estimate a $\Delta\boldsymbol{\theta}$ — a small increment of pose parameters relating the two cameras — such that we minimize the sum of squared intensity error, or the *photometric error* given by:

$$\Delta\boldsymbol{\theta}^* = \underset{\Delta\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{\mathbf{p}\in\boldsymbol{\Omega}} \left\| \mathbf{I}'\left(\mathbf{w}(\mathbf{p}; \boldsymbol{\theta} + \Delta\boldsymbol{\theta})\right) - \mathbf{I}\left(\mathbf{p}\right) \right\|^2, \tag{3}$$

where Ω is a subset of pixel coordinates of interest in the reference frame, and $\mathbf{w}\left(\cdot\right)$ is a *warping* function that depends on the parameter vector we seek to estimate. After every iteration, the current estimate of parameters is updated via an additive rule (*i.e.* $\boldsymbol{\theta} \leftarrow \boldsymbol{\theta} + \Delta\boldsymbol{\theta}$). This process repeats until convergence, or some termination criteria have been met.

This formulation is the standard Lucas-Kanade (forward additive) algorithm [Lucas and Kanade, 1981]. An efficient variation on the Lucas-Kanade algorithm is Baker & Matthews' Inverse Compositional (IC) algorithm. The IC algorithm makes two modifications to the error function that significantly improve efficiency. The first is to interchange the roles of $\mathbf{I}$ (the

reference/template image) with $\mathbf{I}'$ (the input/current image). The second, is to compound incremental estimates using a compositional update rule instead of an additive one. Under the IC formulation we seek an update of the parameters $\Delta\boldsymbol{\theta}$ such that:

$$\Delta\boldsymbol{\theta}^* = \underset{\Delta\boldsymbol{\theta}}{\operatorname{argmin}} \sum_{\mathbf{p}\in\mathbf{I}} \left\|\mathbf{I}\left(\mathbf{w}(\mathbf{p};\Delta\boldsymbol{\theta})\right) - \mathbf{I}'\left(\mathbf{w}(\mathbf{p};\boldsymbol{\theta})\right)\right\|^2. \tag{4}$$

The optimization problem in eq. (4) is nonlinear irrespective of the form of the warping function or the parameters. To obtain a solution, we perform a first-order Taylor expansion and arrive at the following closed form (normal equations):

$$\Delta\boldsymbol{\theta} = \left(\mathbf{J}^\top\mathbf{J}\right)^{-1}\mathbf{J}^\top\mathbf{e}, \tag{5}$$

where $\mathbf{J} = \left(\mathbf{g}(\mathbf{p}_1)^\top, \ldots, \mathbf{g}(\mathbf{p}_m)^\top\right) \in \mathbb{R}^{m\times p}$. Here, m is the number of pixels and $p = |\boldsymbol{\theta}|$ is the number of parameters. Each $\mathbf{g}$ is $\in \mathbb{R}^{1\times p}$ and is given by:

$$\mathbf{g}(\mathbf{p}) = \nabla\mathbf{I}(\mathbf{p})\frac{\partial\mathbf{w}}{\partial\boldsymbol{\theta}}, \tag{6}$$

where $\nabla\mathbf{I} = (I_u, I_v) \in \mathbb{R}^{1\times 2}$ is the image gradient along the u- and v-directions. Finally,

$$\mathbf{e}(\mathbf{p}) = \mathbf{I}'(\mathbf{w}(\mathbf{p};\Delta\boldsymbol{\theta})) - \mathbf{I}(\mathbf{w}(\mathbf{p};\boldsymbol{\theta})) \tag{7}$$

is the *error image*. At the next iteration of the optimization algorithm, parameters of the motion model are updated via the IC rule given by:

$$\mathbf{w}\left(\mathbf{p},\boldsymbol{\theta}\right) \leftarrow \mathbf{w}\left(\mathbf{p},\boldsymbol{\theta}\right) \circ \mathbf{w}\left(\mathbf{p},\Delta\boldsymbol{\theta}\right)^{-1}. \tag{8}$$

We refer the reader to the excellent series by Baker & Matthews [Baker and Matthews, 2004; Baker et al., 2003] for a detailed treatment.

3 Algorithm

Given a reference image $\mathbf{I}$ with an associated disparity map and an input image after camera motion $\mathbf{I}'$, we seek to estimate the parameters of motion such that the expression in eq. (4) is minimized. The warping function is given by:

$$\mathbf{w} : \left(\mathbb{P}^3 \times \mathbb{R}^6\right) \rightarrow \mathbb{R}^2 \tag{9}$$

$$\mathbf{w}(\mathbf{p},\boldsymbol{\theta}) = \pi\left(\boldsymbol{\Gamma}\mathbf{T}(\boldsymbol{\theta})\boldsymbol{\Gamma}^{-1}\mathbf{p}\right) + \mathbf{c}, \tag{10}$$

where $\pi\left(\cdot\right)$ the projection function serving two purposes. First, it performs homogenous division to bring back the point to Euclidean space. Second, it select the first two elements of the normalized vector corresponding tot he x- and y-coordinates. Finally, to obtain 2D pixel coordinates in the image plane we add back the principle point $\mathbf{c}$.

3.1 Jacobian of the warp

In order to use a direct approach we need to compute an analytic expression of the Jacobian with respect to the parameters around the identity $\boldsymbol{\theta} = \mathbf{0}$. Using the Lie algebra parameterization of rigid transformations, *i.e. twist*, $\boldsymbol{\theta} = (\omega_x, \omega_y, \omega_z, \nu_x, \nu_y, \nu_z)^\top \in \mathbb{R}^6$, we obtain the Jacobian of the warping function in eq. (10) per point $\mathbf{p}$ as:

$$\nabla \mathbf{I} \frac{\partial \mathbf{w}}{\partial \boldsymbol{\theta}}\bigg|_{\boldsymbol{\theta}=\mathbf{0}} = \mathbf{g}(\mathbf{p}) = \begin{pmatrix} -fI_y + \alpha y/f \\ fI_x - \alpha x/f \\ yI_x - xI_y \\ I_x\beta \\ I_y\beta \\ \alpha\beta/f \end{pmatrix}^\top \in \mathbb{R}^{1\times 6}, \tag{11}$$

where $\nabla \mathbf{I} = (I_x, I_y)$ is the image gradient, $x = c_u - u$, $y = c_v - v$, $d = u - u_r$, with

$$\alpha = xI_x + yI_y, \quad \text{and} \tag{12}$$

$$\beta = d/B. \tag{13}$$

For m pixels, we stack the values of eq. (11) into an $m \times 6$ matrix and obtain an update of parameters $\Delta\boldsymbol{\theta}$ by solving the normal equations in eq. (5). After every iteration the estimate of pose is updated using the inverse compositional update rule given by:

$$\mathbf{w}(\mathbf{p};\boldsymbol{\theta}) \leftarrow \mathbf{w}(\mathbf{p};\boldsymbol{\theta}) \circ \mathbf{w}(\mathbf{p};\Delta\boldsymbol{\theta})^{-1} \tag{14}$$

$$:= \mathbf{T}(\boldsymbol{\theta}) \exp\left(-\widehat{\Delta\boldsymbol{\theta}}\right), \tag{15}$$

where $\widehat{\cdot} : \mathbb{R}^6 \to \mathfrak{se}(3)$, and $\exp : \mathfrak{se}(3) \to SE(3)$ is the matrix exponential with $\exp(\mathbf{x})^{-1} = \exp(-\mathbf{x})$ (*c.f.* [Blanco, 2010; Ma, Yi and Soatto, Stefano and Kosecka, Jana and Sastry, S. Shankar, 2003]).

A visualization of the Jacobian/steepest descent images per each of the six degrees of freedom is shown in fig. 2. The computation was carried out from the left input image and the disparity shown in fig. 1

3.2 Robustness

It is well known that the least squares optimization (eq. (5)) is sensitive to outliers. In order to obtain a robust estimate we replace the squared error with a *robust* error function [Huber, 1974]. Choice of the robust function is rather arbitrary and can only be determined experimentally [Zhang, 1997].

(a) Left image

(b) Disparity

Figure 1: Input left image and computed disparity using SGM [Hirschmuller, 2005]. We use SGM due to its denser output for better visualization. For pose results, we use block matching stereo. Figure 2 shows the steepest descent images for this input pair.

We experimented with several cost functions and found Tukey's bi-weight [Beaton and Tukey, 1974] to perform the best. This is possibly due to suppressing high residuals instead of only reducing their influence. The bi-weight function for a residual $r_i \in \mathbb{R}$ and parameter/cutoff threshold $\tau \in \mathbb{R}$ is given by:

$$\rho(r_i;\tau) = \begin{cases} \left(1-(r_i/\tau)^2\right)^2 & \text{if } |r_i| \leq \tau; \\ 0 & \text{otherwise.} \end{cases} \tag{16}$$

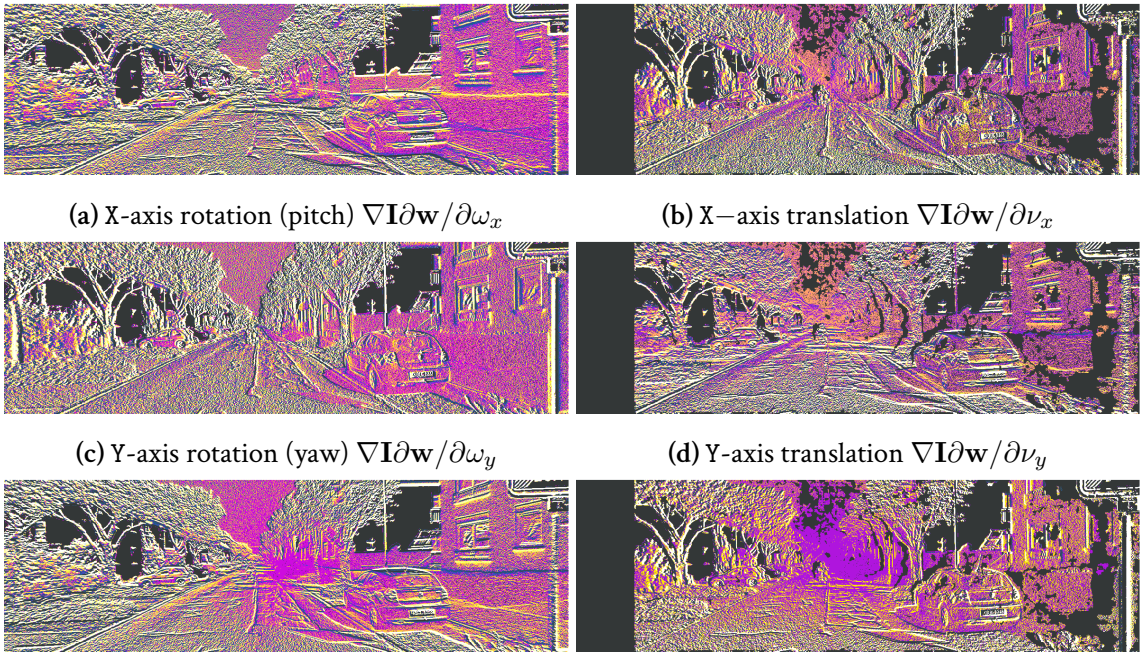
The cutoff threshold τ is set to 4.6851 to obtain a 95% asymptotic efficiency of the normal distribution. The threshold requires normalized residuals with unit deviations. For this purpose, we use a robust estimator of standard deviation. For m observations and p parameters, the robust standard deviation is given by:

$$\hat{\sigma} = 1.4826\,[1+5/(m-p)]\,\underset{i}{\text{median}}|r_i|. \tag{17}$$

The constant 1.4826 is used to obtain the same efficiency of least squares under Gaussian noise, while $[1+5/(m-p)]$ is used to compensate for small data [Zhang, 1997]. In practice, $m \gg p$ and the small data constant vanishes.

In summary, given the list of residuals $\mathbf{r} = (r_1, \ldots, r_m)^\top$, where each residual is given by:

$$r_i = \mathbf{I}'\left(\mathbf{w}(\mathbf{p}_i;\Delta\boldsymbol{\theta})\right) - \mathbf{I}\left(\mathbf{w}(\mathbf{p}_i;\boldsymbol{\theta})\right), \tag{18}$$



(a) X-axis rotation (pitch) $\nabla \mathbf{I} \partial \mathbf{w} / \partial \omega_x$ **(b)** X−axis translation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \nu_x$

(c) Y-axis rotation (yaw) $\nabla \mathbf{I} \partial \mathbf{w} / \partial \omega_y$ **(d)** Y-axis translation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \nu_y$

(e) Z-axis rotation (roll) $\nabla \mathbf{I} \partial \mathbf{w} / \partial \omega_z$ **(f)** Z-axis translation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \nu_z$

Figure 2: Visualization of the Jacobian entries per degree of freedom (DOF) from the input pair shown in fig. 1. The displayed images have been scaled for better visualization. White/yellow indicates higher absolute magnitude, while magenta indicates a lower value. Black/dark indicates invalid date points due to either (i) zero image gradient, or (ii) invalid disparity estimates. By inspecting these images, it is possible to determine which pixels will contribute the most for each DOF. For example, pixels with high gradient magnitude along the image x-axis will contribute better to camera yaw estimation, the 3D rotation about the Y-axis. Similarly, pixels closer to the focus of expansion have a limited contribution to the estimates of forward motion and camera roll. See discussion in [Dellaert et al., 1998]

we compute a robust estimate of the standard deviation $\hat{\sigma}_{\mathbf{r}}$ using eq. (17) and compute the weight per residual as $w_i = \rho\left(r_i/\hat{\sigma}_{\mathbf{r}}\right)$. By concatenating the weights into an $m \times m$ diagonal matrix $\mathbf{W}$, we may obtain an estimate of the parameters at every iteration by solving the following *weighted* normal equations:

$$\Delta\boldsymbol{\theta} = \left(\mathbf{J}^\top \mathbf{W} \mathbf{J}\right)^{-1} \mathbf{J}^\top \mathbf{W} \mathbf{e}. \tag{19}$$

3.3 Appearance variations

Pixel-wise errors cannot model typical appearance variations on real data, even when using a robust error function. There are several algorithms in the image alignment literature that address appearance variations [Bartoli, 2008; Evangelidis and Psarakis, 2008]. A common approach is to the gain + bias model [Szeliski, 2010, ch. 3], which allow affine appearance variations. Typically, affine variations are modeled per image patch. It is, however, possible to gain an increase in accuracy by modeling bias and affine gain globally for all pixels (*c.f.* [Klose et al., 2013]). That is, the reference image intensity can be written as:

$$\mathbf{I}(\mathbf{p}_i) = b + (1 + g)\, \mathbf{I}'(\mathbf{p}_i), \tag{20}$$

where b is the additive bias and g is the multiplicative gain. Under the inverse compositional update rule, we have:

$$\Delta b + (1 + \Delta g)\, \mathbf{I}(\mathbf{p}_i) = b + (1 + g)\, \mathbf{I}'(\mathbf{p}_i). \tag{21}$$

By adding appearance variations, the Jacobian of the warp becomes an 8-vector obtained by appending the derivatives w.r.t. gain and bias. Those values are the pixel intensity and a constant respectively. Finally, appearance variation parameters are updated with:

$$g \leftarrow (g - \Delta g)/\eta, \quad \text{and} \quad b \leftarrow (b - \Delta b)/\eta, \tag{22}$$

where

$$\eta = 1 + \Delta g. \tag{23}$$

Other approaches to illumination invariance include pre-filtering [Vaudrey et al., 2011], or tracking the output of high dimensional function of intensity, such as descriptors [Crivellaro and Lepetit, 2014; Sevilla-Lara and Learned-Miller, 2012].

In this work, we will report results without modeling appearance variations.

3.4 Pixel selection

Traditionally, direct methods are associated with the concept of dense, or semi dense, algorithms that make use of all possible pixel information. Intuitively, the use of as many as possible data points could increase robustness. However, the large number of pixels typically used in direct algorithms incur a high computational cost necessitating implementation on parallel architectures such as high-end GPUs.

In this work, we show that it is not necessary to use as all available information. By reducing the number of pixels, we are able to run at a faster frame rate (real-time) on a single CPU core.

Our pixel selection is based on the feature "bucketing" idea common to feature-based methods [Nister et al., 2004]. The image is virtually split into a grid/buckets, and a certain number of pixels with strong *cornerness* score is kept in each bucket. This was shown to improve visual odometry estimates as this feature selection strategy enforces a more uniform distribution of features across the field of view.

In our case, the influence of a pixel correlates, to an extent, with its gradient magnitude. For example, a pixel with no gradient does not contribute to the optimization as its contribution to the Jacobian in eq. (11) vanishes. Hence, we perform our pixel selection using the absolute gradient magnitude as a substitute for the cornerness map. Pixels with a local absolute gradient magnitude maxima in a neighborhood of 3×3 pixels are used for pose estimation. In contrast to feature-based methods, we do not enforce a maximum number of features per grid cell. Instead we select as many pixels as possible.

The literature on pixel selection for direct pose tracking is sparse. In the case of optical flow, the seminal work of Shi and Tomasi [1994] introduced a feature selection method based on the textureness of the patch surrounding the pixel. The textureness score is obtained by analyzing the Eigenvalues of the design matrix, which depends on gradients.

For applications with pose tracking, Dellaert and Collins [1999] propose a method that selects the pixels that constrain each degree of freedom the most. However, the method requires a known motion prior. Meilland et al. [2010] propose another method based on recursively sorting each dimension of the Jacobians and greedily keeping elements with the highest magnitude. This greedy approach reduces the number of pixels, but it is computationally expensive and was not shown to provide significant tracking improvement over simpler methods such as discarding pixels with an absolute gradient magnitude smaller than a fixed threshold [Klose et al., 2013].

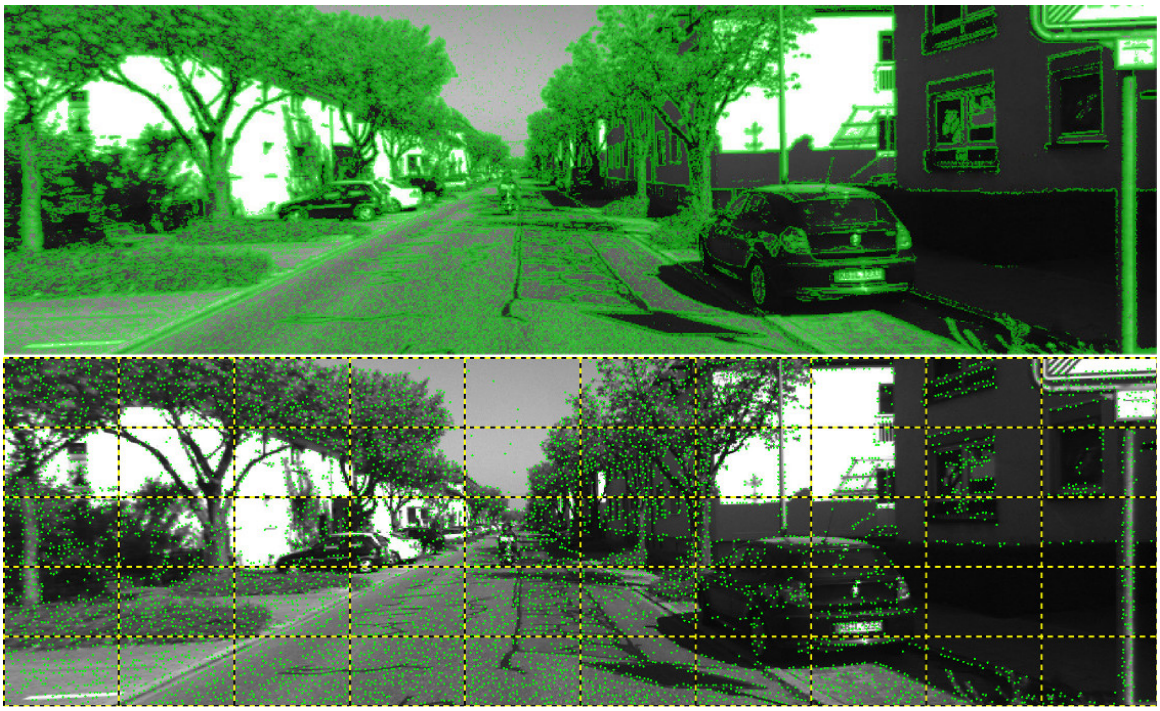


Figure 3: Illustration of pixel selection. Areas shaded with green indicate an pixels with absolute gradient magnitude greater than 5 resulting in 220422 pixels (47.3% of total pixels). Selected pixels are shown in the second row resulting in 7573 (1.62% of total pixels).

An example of selected pixels using our approach is shown in fig. 3. The figure compares the number of pixels that would be detected by absolute gradient magnitude thresholding (with threshold set to five pixels) versus our method. As we will show in section 4, this pixel selection scheme is sufficient for accurate pose tracking and real-time performance using only $1.0 - 3.0\%$ of the usable image pixels.

Pixel selection is implemented for all pyramid levels with image size at least $160 \times 120\text{px}^2$. For smaller images, we simply use all possible pixels with non-vanishing absolute gradient magnitude. In fact, at very low resolutions we have not observed an improvement in accuracy or efficiency when implementing any pixel selection strategy.

3.5 Additional implementation details

Our algorithm does not require elaborate parameter tuning or specialized heuristics. The only tunable parameters are the stereo algorithm parameters, which depend on the dataset. We use a basic block matching stereo (as implemented in OpenCV[1]). Stereo parameters include, SAD window size and disparity range. To address large motions (and speed up the convergence

[1]Using Matlab R2013b http://www.mathworks.com/help/vision/ref/disparity.html

rate) the algorithm is implemented in a scale space pyramid. We do not scale down the disparity image to avoid interpolation across occlusion boundaries. Instead, disparities for coarser levels of the pyramid are interpolated from the disparity map computed at the finest level. Each level of the pyramid is scaled using a factor of $1/2$ of the previous image size and smoothed with a Gaussian filter prior to downsampling with bilinear interpolation. All interpolation operations in this work are done with a bilinear interpolation kernel.[2] Convergence is determined if the norm of the estimated parameters is less than 1×10^{-6}, change in parameters is less than 1×10^{-8} or a maximum number of iterations is reached. The maximum number of iterations was set to 300 iterations on the finest level, and 50 iterations for all other levels.

When using a large number of points, estimating the standard deviation of errors becomes a bottleneck due to the repeated need for computing their median at every iteration. To mitigate this, several authors proposed the use of histograms to approximate the median (*c.f.* [Klose et al., 2013; Tykkälä et al., 2014; Kerl et al., 2013]). This approximation might improve efficiency, but the accuracy of this estimate depends on the resolution of the histogram. Without a sufficiently fine histogram resolution, the approximation accuracy will be compromised.

In contrast, we estimate the median with an efficient algorithm [Musser, 1997] and monitor the change in this estimate across iterations. Once the change in the estimated standard deviation falls below a threshold (1×10^{-6}) we fix the estimate and do not compute it again. The estimate of the median typically converges within the first 10 iterations.

Finally, we gain a small efficiency improvement by scaling the baseline instead of the disparity for each pyramid level. That is, instead of halving disparities per pyramid level, we double the baseline.

The efficiency enhancements mentioned above improve the performance of the algorithm noticeably when using a dense collection of pixels. When using our pixel selection scheme, however, the modifications are negligible.

3.6 Normalization

We experimented with pre-normalizing the selected points per iteration using Hartley's normalization [Hartley, 1997], but did not observe any improvement on accuracy. We also experimented with pre-conditioning the normal equations to improve the Hessian's condition number [Brooks, 2008], but did not observe any improvements. The normalization experi-

[2] We experimented with a cubic kernel, but did not observe any improvements in accuracy to justify the increased computational cost.

ments were conducted on synthetic data with known ground truth [Peris et al., 2012; Martull et al., 2012].

4 Experiments and Results

In this section, we evaluate the performance of our algorithm on different datasets including outdoor and indoor environments. For the outdoors data sets we do not perform any keyframing, even when the robot is stationary. We also do not perform any global optimization/bundle adjustment. If the algorithm has good frame-frame accuracy, then bundle adjustment will only improve results.

In the following, we will call our algorithm DDS: Direct Disparity Space VO.

4.1 KITTI data

We demonstrate the performance of our algorithm on the KITTI odometry benchmark [Geiger et al., 2012] in comparison to two open source algorithms targeted for robotics applications: (1) VISO2 [Geiger et al., 2011] and (2) FOVIS [Huang et al., 2011]. We use both algorithms with the author's default parameters, which perform well.

VISO2 is one of the leading stereo VO algorithms and is a representative example of a feature-based stereo VO method. The algorithm extracts and matches features between the fours images of two stereo frames. Mutual consistency is enforced during this matching, then an estimate of the camera pose is obtained using RANSAC and nonlinear optimization. The final pose estimate is further refined via minimizing the re-projection error in both, the left and the right images. Finally, an Extended Kalman filter with a constant velocity model is used to filter estimates over time.

FOVIS is another excellent VO algorithm that was initially developed for high frame-rate RGB-D images, with an extension for stereo data. FOVIS combines multiple ideas into a single system. First, the algorithm obtains an initial estimate of rotation using a direct method [Mei et al., 2009]. This is then used to as an initialization for feature-based pose estimation using absolute orientation [Horn, 1987]. Finally, the estimated pose is refined by minimizing the reprojection error, bi-directionally, between both frames. The algorithm also implements a keyframing strategy based on the percentage of inliers to reduce drift from accumulated inter-frame estimation noise.

Results on KITTI data are summarized in fig. 4. Our algorithm's average translation error is 2.35% and the average rotation error is 0.0058 degree per meter, which are accurate for a frame-frame method. In particular, our rotation error is close to, and sometimes better,

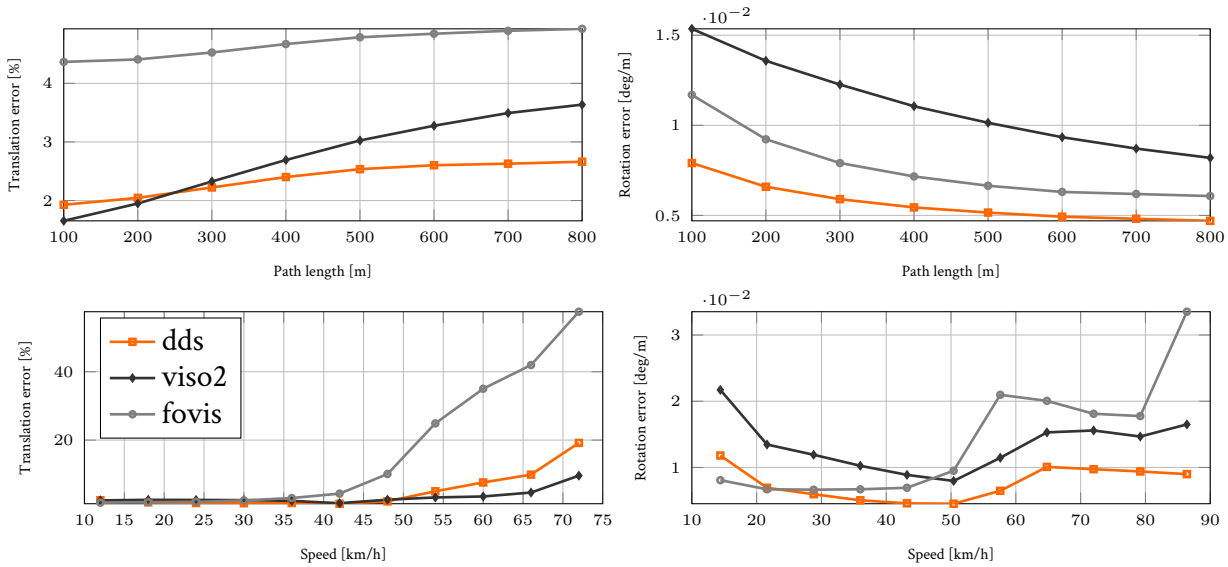


Figure 4: Results for the 11 training sequences of the KITTI odometry benchmark.

than some multi-frame methods on the KITTI benchmark. The main sources of error appear in estimating the translation of the camera at high vehicle speeds. The high speed driving produces larger baseline between images and violates the small motion assumption. Note, for KITTI data our algorithm is initialized with the identity transformation.

Interestingly, rotation error for both FOVIS and ours (DDS) are better than VISO2. This is potentially due to a more accurate rotation estimation results when using image intensities directly. In fact, most of VISO2's rotation drift appears to be in roll estimates and consequently camera height. We hypothesize that this is related to the small vertical FOV of the camera. In contrast, direct methods are able to better exploit this reduced FOV by including more information and not relying on the accuracy feature subpixel localization.

4.2 Wean hall (indoor data)

This dataset was collected with a Bumblebee2 stereo color camera of resolution 640×480 px^2 at $\approx$ 30 Hz [Alismail et al., 2010].[3] A summary of the data is shown in table 1. The camera was mounted on a LAGR robot.[4] For ground truth, we use a 2D estimate of robot pose using calibrated wheel odometry combined with an accurate gyroscope. This is an approximate ground truth, but it is reasonable as the indoor environment is flat. The camera's raw output

[3] Data available on http://www.cs.cmu.edu/~halismai/

[4] See http://www.nrec.ri.cmu.edu/projects/lagr/. Our robot is slightly modified to use a single stereo camera titled towards the ground, and equipped with an accurate fiber optic gyro from KVH, model # DSP-3000.

Table 1: Wean hall data summary

focal length	$\approx 3.88\,\text{mm}$	baseline	$0.12\,\text{m}$
# frames	6510	distance	$\approx 294\,\text{m}$

Figure 5: Example images from the Wean hall dataset.

is a Bayer pattern, which we interpolate to a color image prior to use for motion estimation. The Bayer pattern causes a reduction in resolution in comparison to native grayscale output.

The dataset features strong specular reflections on the ground, lack of texture in some areas as well as high frequency repetitive texture in others. An example is shown in fig. 5. The robot was driven at an average speed of $\approx 0.7\,\text{m}\,\text{s}^{-1}$.

Due to the high frame-rate of the camera, we implement a keyframing strategy based on the magnitude of the estimated motion. The pose of each non-keyframe is initialized with the current estimate of pose until the motion magnitude is large enough. Upon keyframing, we reset the pose initialization to the identity. For the results shown here, we keyframe when the translation magnitude is $30\,\text{cm}$, or when any of the rotation angles exceed 5 degrees.

4.3 Efficiency

Our algorithm is implemented using a combination of Matlab and unoptimized C++. The performance is on par with VISO2 and FOVIS as shown in fig. 8. On one frame, our algorithm took 135ms to estimate pose, in which it was stuck in a local minima taking close to 130 iterations as shown in fig. 9 (we require $\approx$ 1ms per iteration). We expect that tunning the termination criteria of the algorithm will avoid this situation, but this is not necessary for higher frame-rate data.

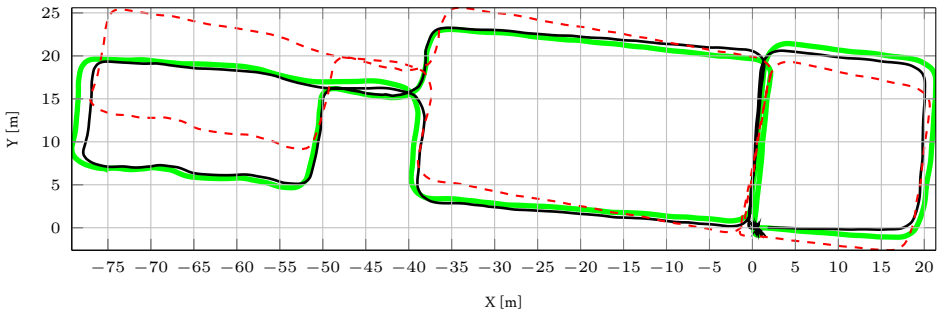


Figure 6: Top view of estimated path for ground truth (——), VISO2 (- - -) and DDS (——).

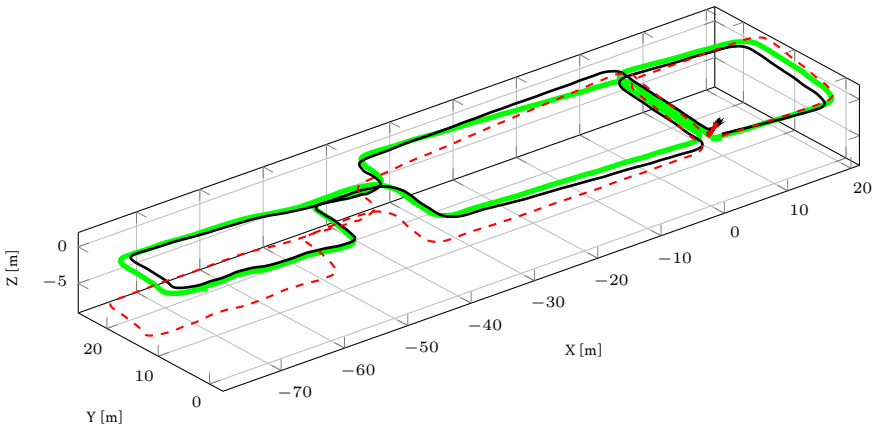


Figure 7: 3D view of camera trajectory for ground truth (——), VISO2 (- - -) and DDS (——). Note the reduced drift in camera height estimates using our algorithm.

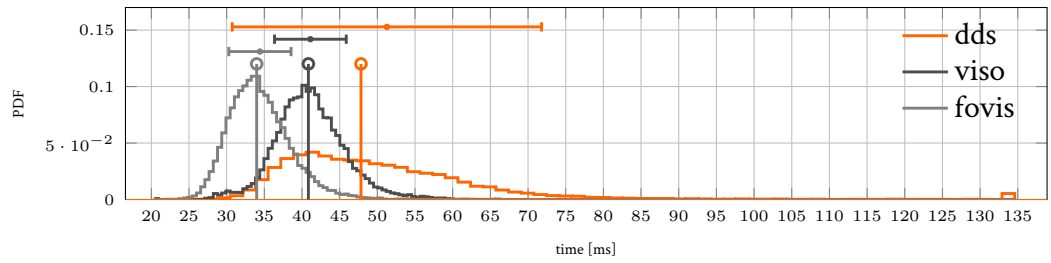


Figure 8: Timing results on KITTI data. The mean and standard deviation is shown with a dot and a horizontal bar. The median is shown with a vertical bar. Median run time for DDS is 47.8, VISO2 is 48.84 and FOVIS is 33.98 ms.

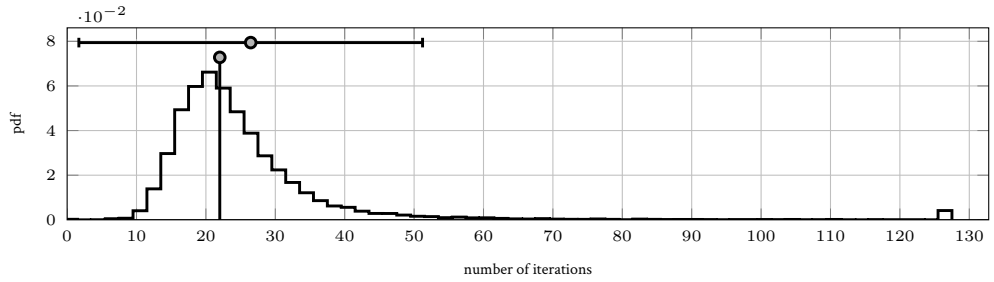


Figure 9: Number of iterations on KITTI data for our algorithm. The mean number of iterations is 26.46 with standard deviation of 24.76, and a median of 22 iterations.

As we do not implement any keyframing strategy for KITTI data due to the large baseline between frames, we have to recompute the Jacobian at every new frame. When we are able to discard frames due to a higher frame-rate, the Jacobian remains constant for a longer period of time and the run time improves as shown in fig. 10. The figure shows the running time has two modes. The first one is $\approx$ 3ms, which occurs when we do not update the current keyframe. The other mode is $\approx$ 12ms, which occurs when we add a new keyframe, which requires re-computing the Jacobian. In addition to not having to re-compute the Jacobian, the closely spaced images allow the algorithm to converge faster. There are some spurious spots where the algorithm gets stuck in a local minima and require $\approx$ 40ms. The overall run time is $\approx$ 120 Hz on average.

4.4 Pixel selection

Not all pixels contribute equally to the cost function; only a small number of pixels contribute towards the error function [Dellaert and Collins, 1999]. The simplest approach to pixel selection is to discard pixels with gradient magnitude less than a pre-specified fixed threshold. The rational is that a pixel with zero gradient does not contribute to the error function. However,

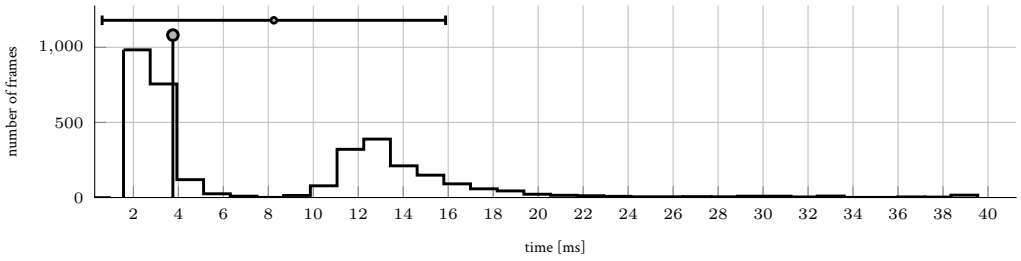


Figure 10: Timing results on high frame-rate indoor data. The mean running time is 8.25 ± 7.63ms and the median is 3.75ms.

the mere magnitude of the gradient is not a sufficient predictor of a pixel's performance. For instance, restricting the optimization to pixels with a high gradient magnitude might bias the solution in undesirable ways.

To illustrate this, we run our algorithm using all available pixels with an absolute gradient magnitude greater than a threshold. The average performance on the KITTI benchmark training data is illustrated in fig. 11.

As shown in the figure, including all possible pixels is suboptimal. Similarly, selecting pixels with a very high gradient magnitude is suboptimal as well. A good threshold for the tested optics and the benchmark environment is within 10% of the image dynamic range.

4.5 Dense 3D reconstruction

We demonstrate a simple scheme to obtain sufficiently dense 3D reconstruction using our algorithm. The output of our algorithm after every keyframe consists of an estimate of the camera pose, as well as the set of disparity space point and their IRLS weights upon convergence.

After every keyframe, we select points with weights of at least $0.75/1.0$ and within a range close to the camera (at most 30 m). We triangulate the points, and project them to the world coordinates using the current estimate of the keyframe's pose.

Since our pixel selection scheme is not based on features, the selected points over multiple frames do not correspond to a single 3D point space. Hence, the overlap between frames will consist of mostly distinct 3D points that produce a dense reconstruction of the environment.

Examples of our reconstruction are shown in fig. 16, fig. 17, fig. 18, fig. 14, fig. 15 and fig. 19. Note, disparity maps were obtained via block matching and include a large amount of noise and outliers. The 3D reconstruction results indicate the accuracy of the method over a short sequence frames as well as robustness against outliers.

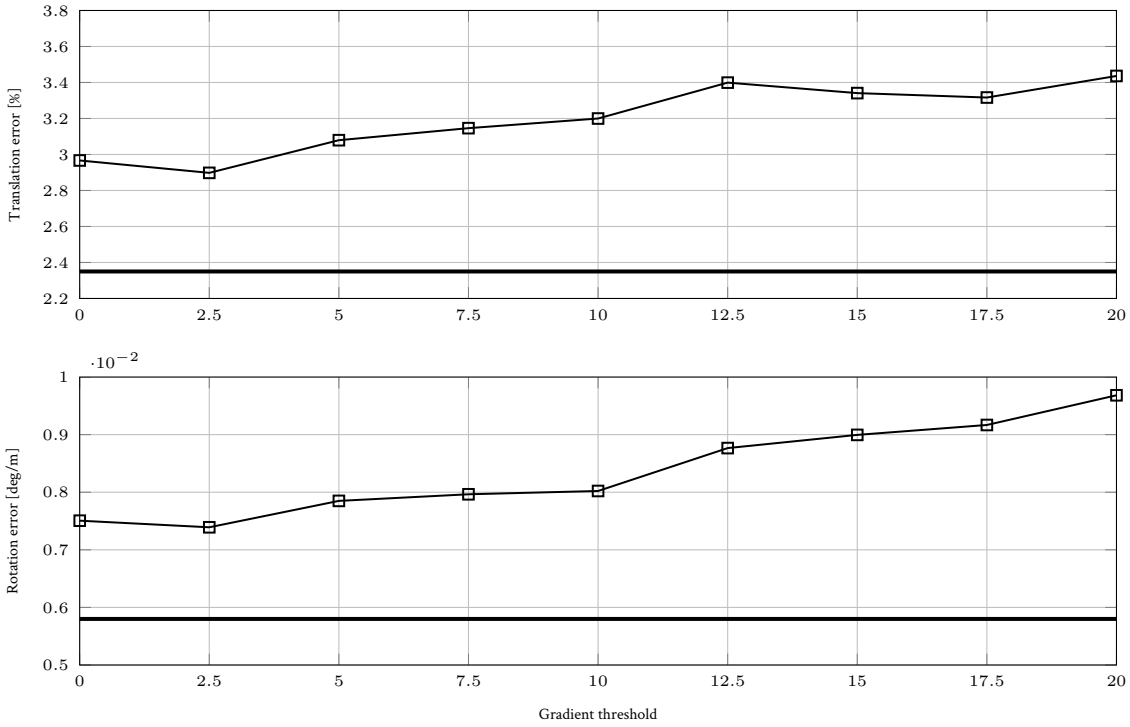


Figure 11: Average performance on KITTI training data with different absolute gradient magnitude cutoff thresholds. The input images are converted to floating point prior to computing the gradients, and their range is kept $\in [0, 255]$. Detailed evaluation plots are shown in fig. 13

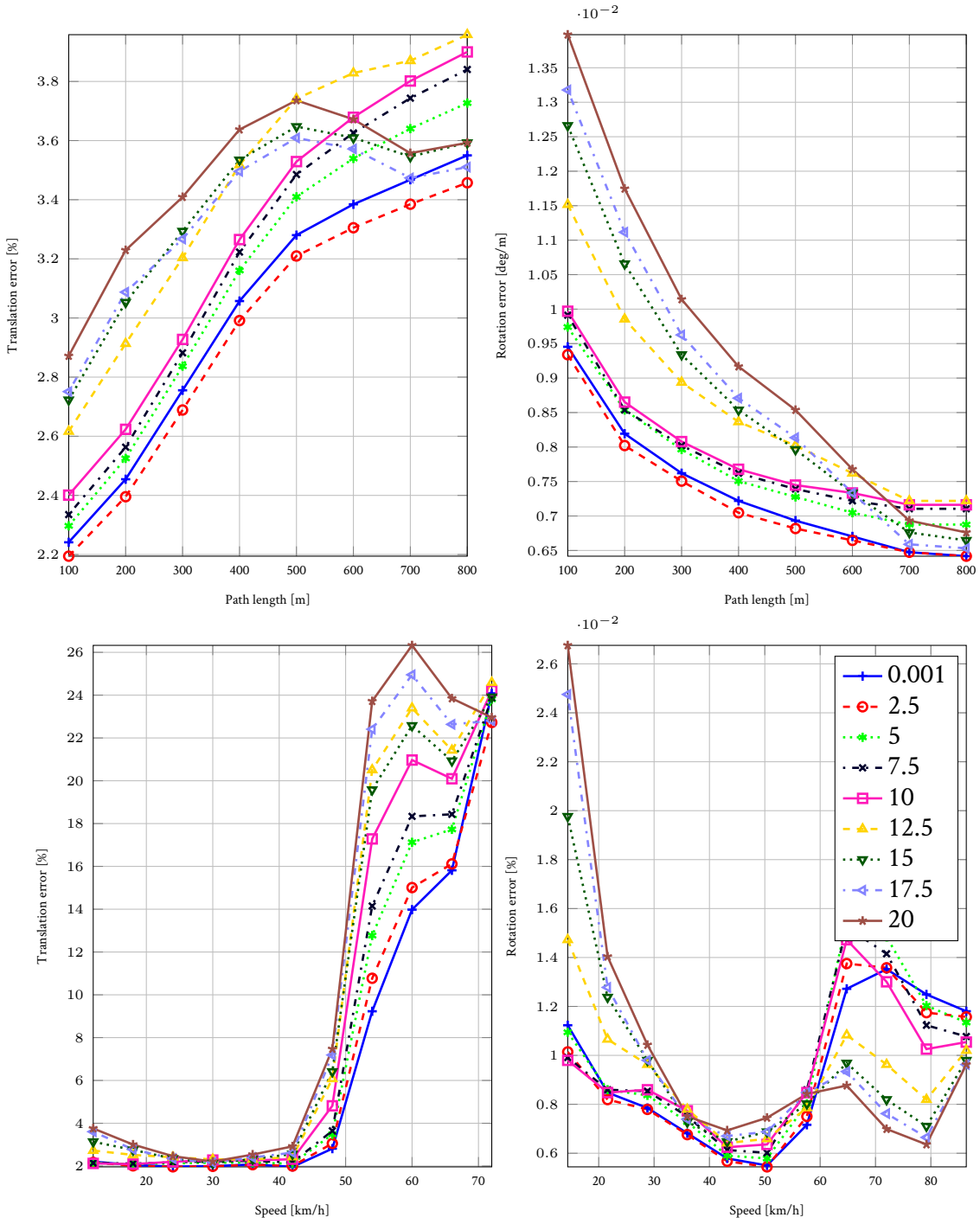


Figure 12: Detailed performance on KITTI benchmark for various absolute gradient magnitude cutoff thresholds. See fig. 11 for a summary.

Figure 13

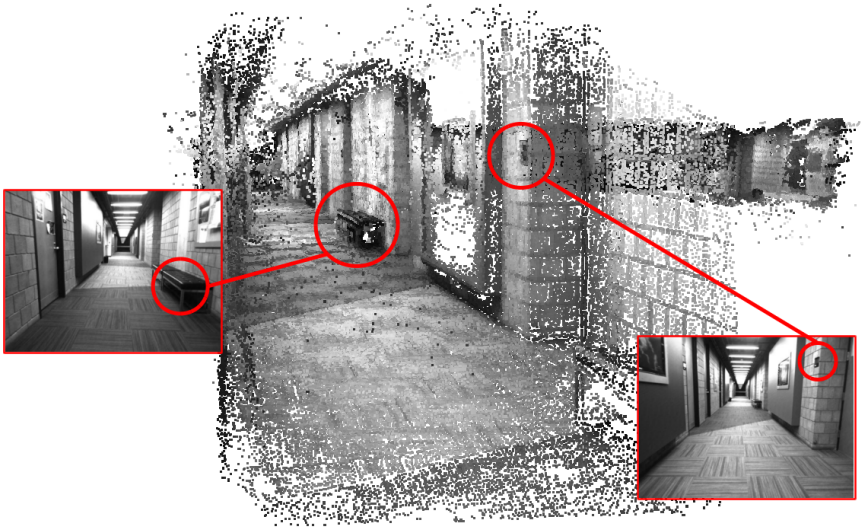


Figure 14: Example of dense 3D reconstruction of our indoor dataset.

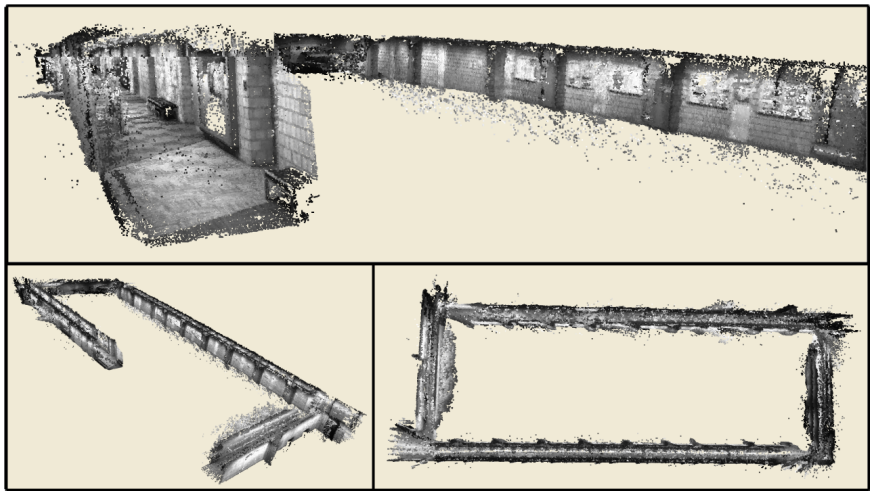


Figure 15: Example of dense 3D reconstruction from different novel views. In top figure we can easily distinguish poster boards mounted on the corridor as well as details of carpet. Inset on bottom right shows a top view of a single loop in the corridor. The bright line in the middle of the corridor is caused by lighting reflections on a glossy floor.

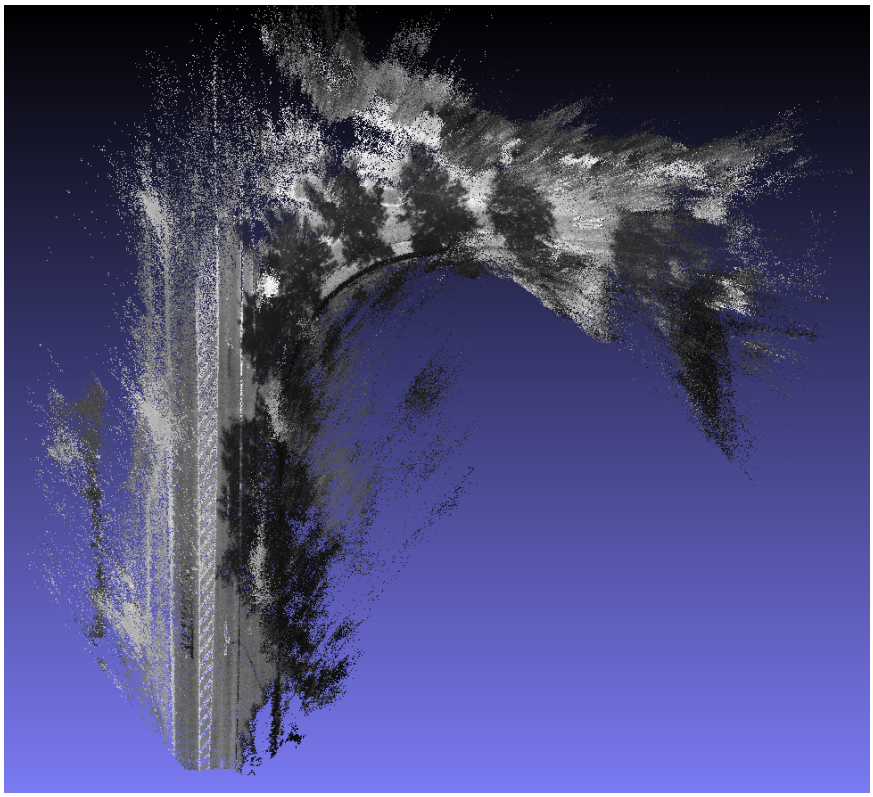


Figure 16: Top view of a segment from KITTI sequence # 03. Reconstruction of line marking on the road is consistent over multiple frames.

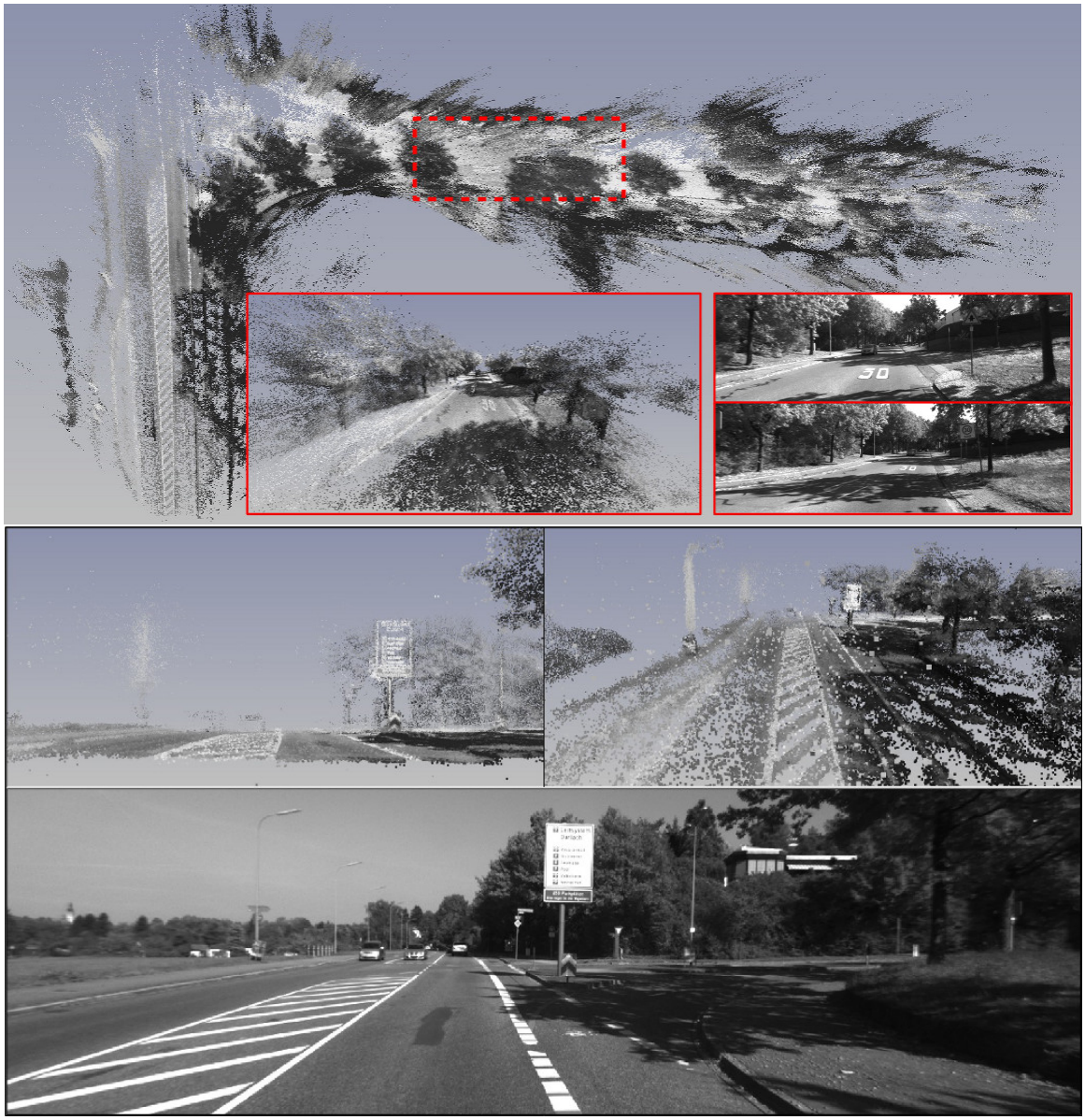


Figure 17: Top view and close-up view of of a segment from KITTI sequence # 02. Reconstruction of line marking on the road is consistent over multiple frames as well the visible traffic signs and tree shadows.

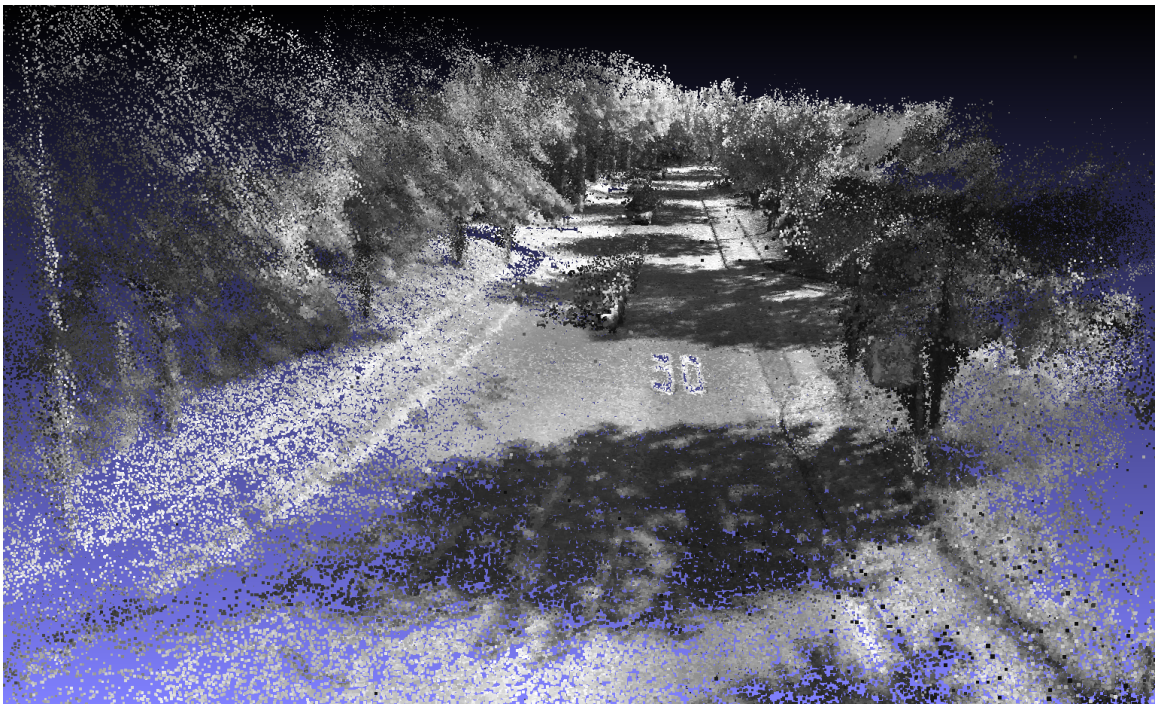


Figure 18: Close up view from KITTI sequence # 03 showing clear markers on the road, traffic signs and tree shadows.

Figure 19: First row shows a top view from our indoor dataset. Images along the corridor are shown in the second row. The first image shows that floor of the model is flat as expected. The last row shows a side view of the corridor. The 3D points are obtained automatically from the algorithm without post-processing or filtering. The corridor length is ≈ 40 m.

4.6 Visual odometry from stereo thumbnails

We evaluate the robustness of the algorithm using low resolution images on the scale of a thumbnail (178×54) and compare it against VISO2 using full resolution images (1241×376). Results are shown in fig. 20 and fig. 21. Considering the low resolution of the stereo, performance of our algorithm remains accurate and robust.

5 Discussion & Future Work

5.1 Points at infinity

Scene points on the plane at infinity are independent of camera translation and can be used to estimate camera rotation and calibration [Hartley and Zisserman, 2004]. In contrast to other work (*e.g.* [Kaess et al., 2009]), our algorithm can make use of points at infinity without special handling. This is particularly useful for outdoor applications and indoor applications with short-baseline stereo.

5.2 Effect of the stereo algorithm

In this work we did not observe a need to use sophisticated stereo matching algorithms. We suspect, however, that an improved stereo, such as SGM [Hirschmuller, 2005] (which can operate at real time on an FPGA [Wedel et al., 2008]), might increase the algorithm's robustness and accuracy. Enhanced stereo might also improve convergence speed. This is not guaranteed as (semi) global stereo algorithms tend to over smooth the estimated disparities. This issues remains to be validated experimentally.

5.3 Improvements

In this work we dealt with the problem of pose estimation only (camera tracking). Two important improvements are possible. The first would be structure/disparity refinement. We can include disparity refinement in the same pose tracking framework by using observations form the right image. Another possibility is modeling disparities with some surface representation (*c.f.* [Silveira et al., 2008]), or filtering [Matthies et al., 1989; Vaudrey et al., 2008]. The second important improvement is integrating information from multiple frames in a bundle adjustment/filtering framework [Strasdat et al., 2012; Maier et al., 2014]. This, in fact, is necessary to reduce drift over long sequences. Both improvements are good avenues of future work.

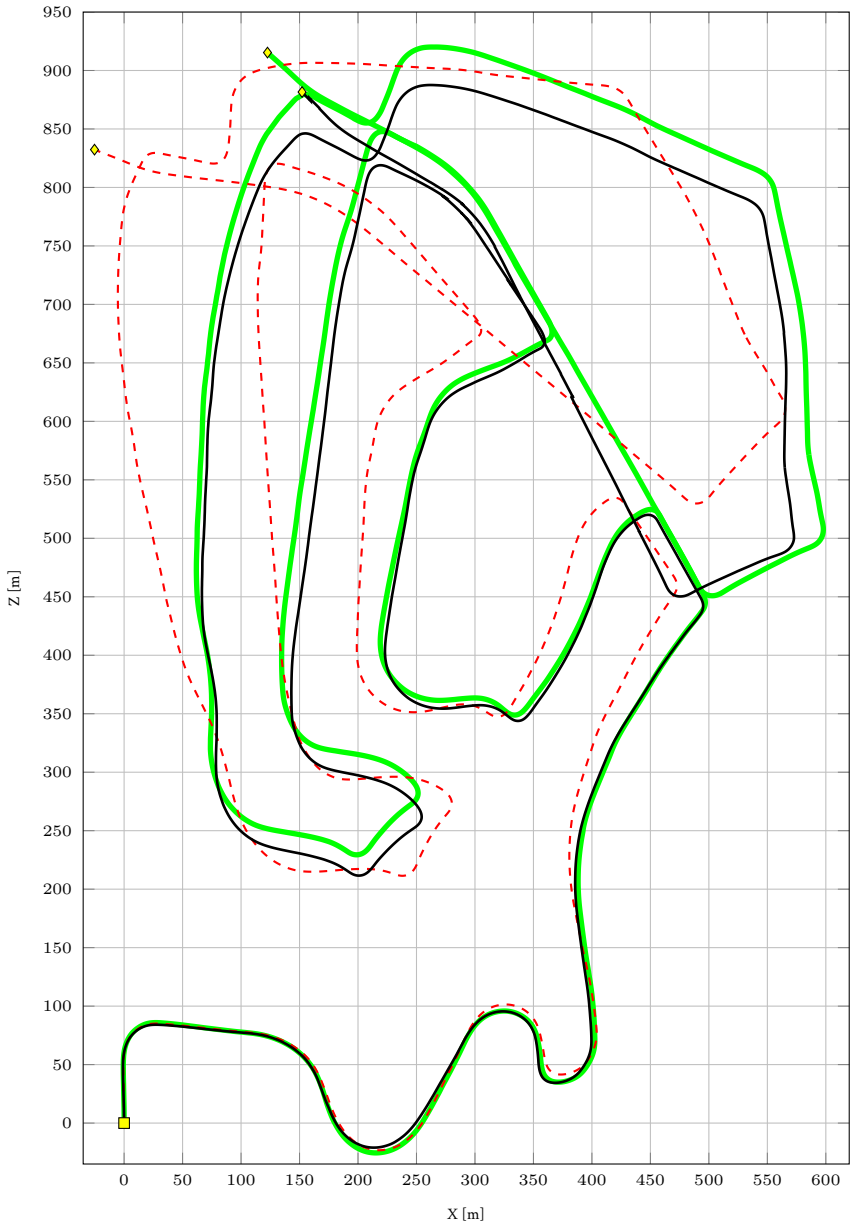


Figure 20: Result on KITTI Seq. 02. Ground truth is shown in (——), VISO2 path is in (- - -), and DDS is in (——). The (□) indicates the start of the sequence, and (◇) indicates the final location. Our results are generated from an image of size 178×54, while VISO2 results are generated from the full resolution 1241×376.

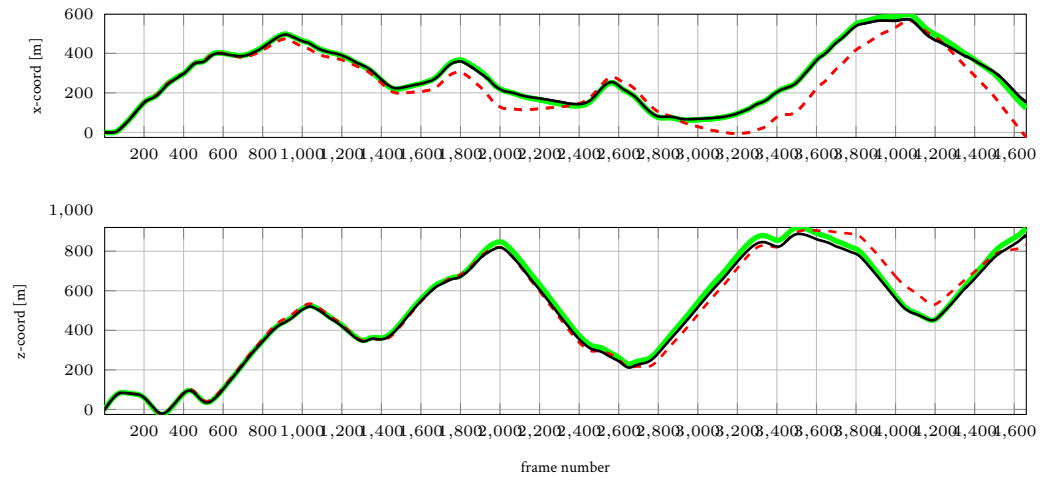


Figure 21: KITTI Seq. 02 x- and z-coordinates for ground truth (——), VISO2 (- - -) and DDS (——).

6 Conclusions

In this work, we presented a *direct* framework for visual odometry using a warping function in *disparity space* (DDS). The disparity space concept is general and applicable to other sensors, such as the Kinect, which natively produces disparities. By using a warping function in disparity space, scene points depth does not appear in the Jacobian of the warp. Hence, the parameters of camera rotation are separated from their translation counterparts. This is consistent with main interpretation of a calibrated camera as an angle measuring device. The algorithm is shown to be robust, in particular to low image resolution. Experiments on real outdoor and indoor data illustrate the applicability of the algorithm to various environments with little to none manual parameter tuning.

We have also shown that direct methods *need not* be dense to produce an accurate and robust estimation of camera pose. We implemented a simple pixel selection strategy that reduced the number of pixels significantly allowing the algorithm to run in real-time on a single CPU core.

Appendices

A Comparison with 3D warping

Disparity space warping produces "smoother" Jacobians in comparison to 3D warping with triangulated points. In table 2 we show side-by-side the Jacobian terms when using 3D warping

Table 2: Jacobian when warping with disparity space versus warping with 3D points. Pose is represented with a twist, $\boldsymbol{\theta} = (\omega_x, \omega_y, \omega_z, \nu_x, \nu_y, \nu_z)^\top$. Depth of a 3D point is denoted by z and a disparity space point with $(x, y, d)^\top$. We let $\nabla \mathbf{I} = \mathbf{g}$, $\alpha = xI_x + yI_y$, focal length $f = 1$ and principle point $\mathbf{c} = (0, 0)^\top$.

term	Disparity	3D
$\mathbf{g}\partial\mathbf{w}/\partial\omega_x$	$-I_y + \alpha y$	$-\left(y^2 I_x + xyI_x + z^2 I_y\right)/\boldsymbol{z^2}$
$\mathbf{g}\partial\mathbf{w}/\partial\omega_y$	$-I_x + \alpha x$	$\left(x^2 I_x + xyI_y + z^2 I_x\right)/\boldsymbol{z^2}$
$\mathbf{g}\partial\mathbf{w}/\partial\omega_z$	$yI_x - xI_y$	$\left(xI_y - yI_x\right)/\boldsymbol{z}$
$\mathbf{g}\partial\mathbf{w}/\partial\nu_x$	$\boldsymbol{d}I_x/B$	$I_x/\boldsymbol{z}$
$\mathbf{g}\partial\mathbf{w}/\partial\nu_y$	$\boldsymbol{d}I_y/B$	$I_y/\boldsymbol{z}$
$\mathbf{g}\partial\mathbf{w}/\partial\nu_z$	$\boldsymbol{d}\alpha/B$	$\alpha/\boldsymbol{z^2}$

as commonly performed in RGB-D systems versus warping in disparity space.

An advantage of a disparity space warp is a "more linear" Jacobian. In particular, there are no divisions by the square of depth, which might cause numerical instability. Furthermore, the distribution of disparity values is more uniform than the distribution of triangulated depth. For example, a difference in disparity between two points of δd results in a quadric difference in depth.

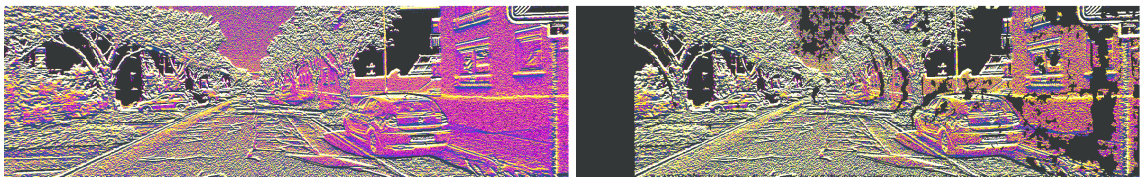
Finally, we note that when using a 3D warp, it is not possible to estimate the rotation of the camera from points at infinity. This is highly undesirable, as points at infinity are invariant to scene depth and hence an excellent source for rotation estimation. In disparity space, we indeed have the correct relationship. The camera rotation estimate does not depend on scene points depth.

A side by side comparison of the Jacobian values on real images is shown in fig. 23 and fig. 24.

(a) Left image

(b) Disparity

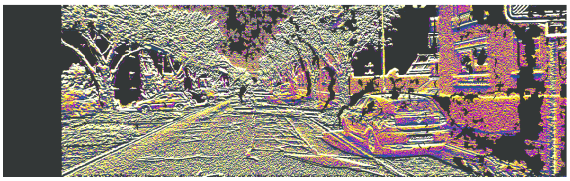
Figure 22: Input left image and computed disparity using SGM [Hirschmuller, 2005]. We use SGM due to its denser output for better visualization. Figure 24 shows the steepest descent images for this input pair.



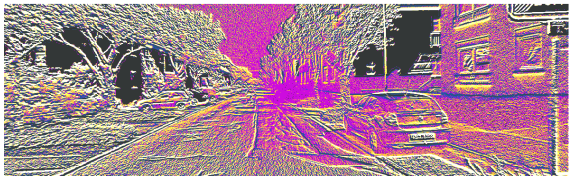
(a) X-axis rotation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \omega_x$

(b) X-axis rotation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \omega_x$

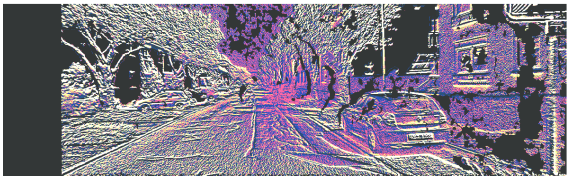
(c) Y-axis rotation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \omega_y$



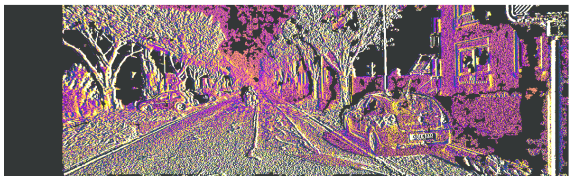
(d) Y-axis rotation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \omega_y$



(e) Z-axis rotation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \omega_z$

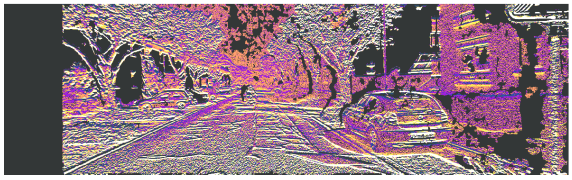


(f) Z-axis rotation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \omega_z$

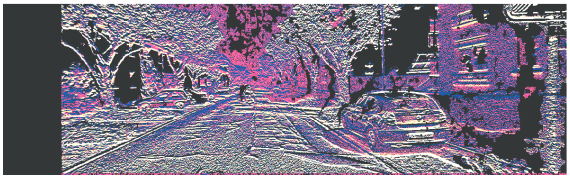


(g) X−axis translation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \nu_x$

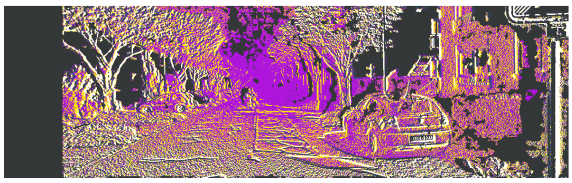
(h) X−axis translation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \nu_x$



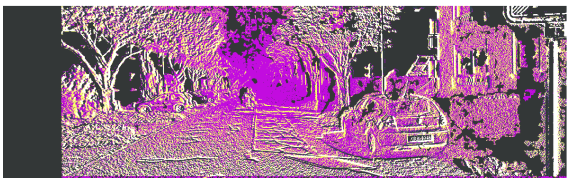
(i) Y-axis translation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \nu_y$



(j) Y-axis translation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \nu_y$

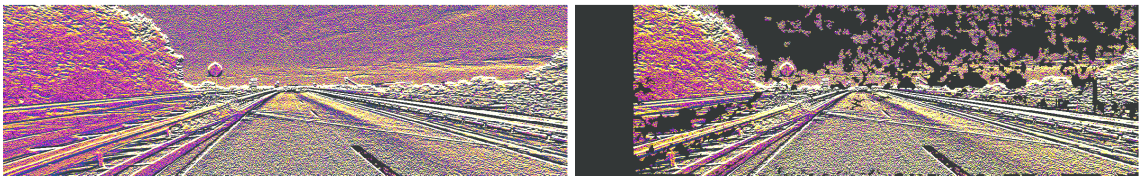


(k) Z-axis translation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \nu_z$



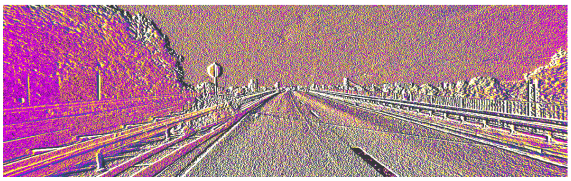
(l) Z-axis translation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \nu_z$

Figure 23: Comparison between disparity space warping and 3D point cloud warping for the input image shown in fig. 1.

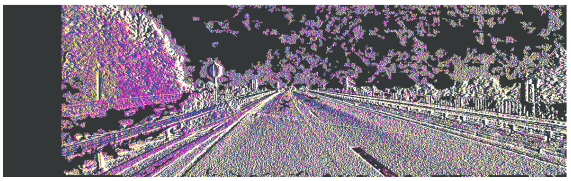


(a) X-axis rotation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \omega_x$

(b) X-axis rotation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \omega_x$

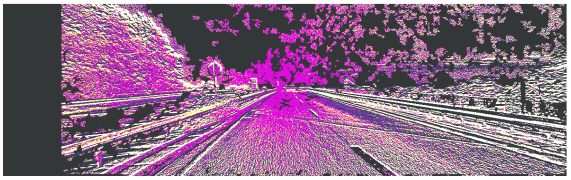


(c) Y-axis rotation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \omega_y$

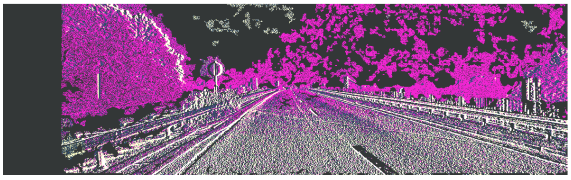


(d) Y-axis rotation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \omega_y$

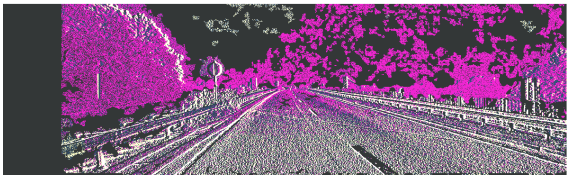
(e) Z-axis rotation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \omega_z$



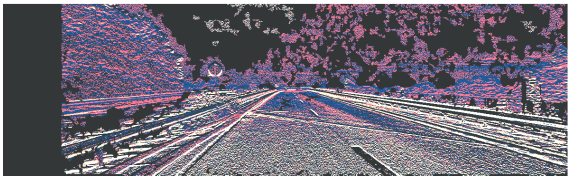
(f) Z-axis rotation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \omega_z$



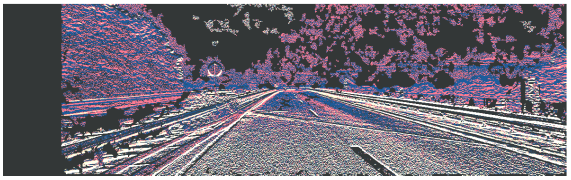
(g) X−axis translation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \nu_x$



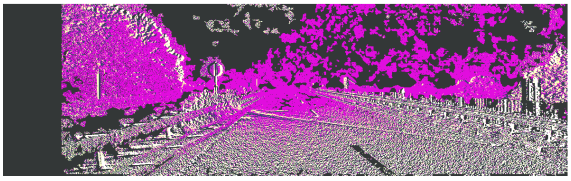
(h) X−axis translation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \nu_x$



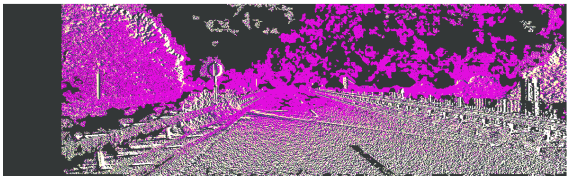
(i) Y-axis translation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \nu_y$



(j) Y-axis translation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \nu_y$



(k) Z-axis translation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \nu_z$



(l) Z-axis translation $\nabla \mathbf{I} \partial \mathbf{w} / \partial \nu_z$

Figure 24: Comparison between disparity space warping and 3D point cloud warping for the input image shown in fig. 22. On right side is the result of disparity space, and on the left is the result of 3D warping. Usable pixels for translation estimation are the same for the two methods. For rotations, however, 3D warping discards many points due to zero, or missing disparities.

References

Hatem Alismail, Brett Browning, and M Bernardine Dias. Evaluating Pose Estimation Methods for Stereo Visual Odometry on Robots. In *the 11th Int'l Conf. on Intelligent Autonomous Systems (IAS-11)*, 2010.

Cedric Audras, A Comport, Maxime Meilland, and Patrick Rives. Real-time dense appearance-based SLAM for RGB-D sensors. In *Australasian Conf. on Robotics and Automation*, 2011.

Simon Baker and Iain Matthews. Lucas-kanade 20 years on: A unifying framework. *International Journal of Computer Vision*, 56(3):221–255, 2004.

Simon Baker, Ralph Gross, and Iain Matthews. Lucas-Kanade 20 Years On: A Unifying Framework: Part 3. Technical Report CMU-RI-TR-03-35, Robotics Institute, Pittsburgh, PA, November 2003.

Adrien Bartoli. Groupwise geometric and photometric direct image registration. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(12):2098–2108, 2008.

Albert E. Beaton and John W. Tukey. The Fitting of Power Series, Meaning Polynomials, Illustrated on Band-Spectroscopic Data. *Technometrics*, 16(2):pp. 147–185, 1974. ISSN 00401706.

M.J. Black and P. Anandan. A framework for the robust estimation of optical flow. In *Computer Vision, 1993. Proceedings., Fourth International Conference on*, pages 231–236, May 1993. doi: 10.1109/ICCV.1993.378214.

Jose-Luis Blanco. A tutorial on se (3) transformation parameterizations and on-manifold optimization. *University of Malaga, Tech. Rep*, 2010.

R Brooks. *Efficient and Reliable Methods for Direct Parameterized Image Registration*. PhD thesis, McGill University, 2008.

Andrew I Comport, Ezio Malis, and Patrick Rives. Real-time quadrifocal visual odometry. *The International Journal of Robotics Research*, 29(2-3):245–266, 2010.

Alberto Crivellaro and Vincent Lepetit. Robust 3D Tracking with Descriptor Fields. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2014.

F. Dellaert, S. Thrun, and C. Thorpe. Jacobian images of super-resolved texture maps for model-based motion estimation and tracking. In *Applications of Computer Vision, 1998. WACV '98. Proceedings., Fourth IEEE Workshop on*, pages 2–7, Oct 1998.

Frank Dellaert and Robert Collins. Fast image-based tracking by selective pixel integration. In *Proceedings of the ICCV Workshop on Frame-Rate Vision*, pages 1–22, 1999.

David Demirdjian and Trevor Darrell. Motion estimation from disparity images. In *In Proc. of Int'l Conf. on Computer Vision (ICCV)*, 2001.

Jakob Engel, Thomas Schöps, and Daniel Cremers. LSD-SLAM: large-scale direct monocular SLAM. In David J. Fleet, Tomás Pajdla, Bernt Schiele, and Tinne Tuytelaars, editors, *Computer Vision - ECCV 2014*, volume 8690 of *Lecture Notes in Computer Science*, pages 834–849. Springer, 2014. ISBN 978-3-319-10604-5. doi: 10.1007/978-3-319-10605-2_54.

Georgios D Evangelidis and Emmanouil Z Psarakis. Parametric image alignment using enhanced correlation coefficient maximization. *IEEE Trans. on Pattern Analysis and Machine Intelligence*, 30(10), 2008.

Zheng Fang and Sebastian Scherer. "Experimental Study of Odometry Estimation Methods using RGB-D Cameras". In *2013 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, September 2014.

Martin A. Fischler and Robert C. Bolles. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. *Commun. ACM*, 24(6), 1981. ISSN 0001-0782. doi: 10.1145/358669.358692.

Christian Forster, Matia Pizzoli, and Davide Scaramuzza. SVO: Fast Semi-Direct Monocular Visual Odometry. In *Proc. IEEE Intl. Conf. on Robotics and Automation*, 2014.

Andreas Geiger, Julius Ziegler, and Christoph Stiller. StereoScan: Dense 3d Reconstruction in Real-time. In *Intelligent Vehicles Symposium (IV)*, 2011.

Andreas Geiger, Philip Lenz, and Raquel Urtasun. Are we ready for Autonomous Driving? The KITTI Vision Benchmark Suite. In *Conference on Computer Vision and Pattern Recognition (CVPR)*, 2012.

Michael A Gennert and Shahriar Negahdaripour. Relaxing the brightness constancy assumption in computing optical flow, 1987.

Ankur Handa, Richard A. Newcombe, Adrien Angeli, and Andrew J. Davison. Real-Time Camera Tracking: When is High Frame-Rate Best? In *European Conf. on Computer Vision (ECCV)*, volume 7578, pages 222–235. 2012. ISBN 978-3-642-33785-7. doi: 10.1007/978-3-642-33786-4_17.

Robert M. Haralick, Chung-Nan Lee, Karsten Ottenberg, and Michael Nölle. Review and analysis of solutions of the three point perspective pose estimation problem. *Int'l Journal of Computer Vision (IJCV)*, 1994. ISSN 0920-5691.

R.I Hartley. In defense of the eight-point algorithm. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 19(6):580–593, Jun 1997. ISSN 0162-8828. doi: 10.1109/34.601246.

Richard Hartley and Andrew Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, second edition, 2004.

Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. RGB-D mapping: Using Kinect-style depth cameras for dense 3D modeling of indoor environments. *IJRR*, 31 (5):647–663, 2012.

H. Hirschmuller. Accurate and efficient stereo processing by semi-global matching and mutual information. In *Computer Vision and Pattern Recognition*, 2005. doi: 10.1109/CVPR.2005.56.

Berthold KP Horn. Closed-form solution of absolute orientation using unit quaternions. *JOSA A*, 4(4):629–642, 1987.

Berthold KP Horn and Brian G Schunck. Determining optical flow. *Artificial intelligence*, 17 (1):185–203, 1981.

A Howard. Real-time stereo visual odometry for autonomous ground vehicles. In *Int'l Conf. on Intelligent Robots and Systems*, 2008. doi: 10.1109/IROS.2008.4651147.

Ma, Yi and Soatto, Stefano and Kosecka, Jana and Sastry, S. Shankar. *An Invitation to 3-D Vision: From Images to Geometric Models*. Springer Verlag, 2003. ISBN 0387008934.

Albert S Huang, Abraham Bachrach, Peter Henry, Michael Krainin, Daniel Maturana, Dieter Fox, and Nicholas Roy. Visual odometry and mapping for autonomous flight using an RGB-D camera. In *International Symposium on Robotics Research (ISRR)*, pages 1–16, 2011.

Peter Huber. *Robust Statistics*. Wiley, New York, 1974.

M. Irani and P. Anandan. About Direct Methods. In *Vision Algorithms: Theory and Practice*, volume 1883, pages 267–277, 2000. ISBN 978-3-540-67973-8. doi: 10.1007/3-540-44480-7_18.

M. Kaess, Kai Ni, and F. Dellaert. Flow separation for fast and robust stereo odometry. In *IEEE Conf. on Robotics and Automation*, pages 3539–3544, May 2009. doi: 10.1109/ROBOT.2009.5152333.

Christian Kerl, Jurgen Sturm, and Daniel Cremers. Dense visual slam for RGB-D cameras. In *Int'l Conf. on Intelligent Robots and Systems*, 2013.

Sebastian Klose, Philipp Heise, and Alois Knoll. Efficient compositional approaches for real-time robust direct visual odometry from RGB-D data. In *IEEE/RSJ Int'l Conf. on Intelligent Robots and Systems*, 2013.

Tony Lindeberg. *Scale-space theory in computer vision*. Springer, 1994. ISBN 978-1-4419-5139-7.

Bruce D. Lucas and Takeo Kanade. An Iterative Image Registration Technique with an Application to Stereo Vision (DARPA). In *Proc. of the 1981 DARPA Image Understanding Workshop*, pages 121–130, April 1981.

R. Maier, J. Sturm, and D. Cremers. Submap-based Bundle Adjustment for 3D Reconstruction from RGB-D Data. In *German Conference on Pattern Recognition (GCPR)*, Münster, Germany, September 2014.

Sarah Martull, Martin Peris, and Kazuhiro Fukui. Realistic CG stereo image dataset with ground truth disparity maps. In *ICPR workshop TrakMark2012*, volume 111, pages 117–118, 2012.

L. Matthies and Steven Shafer. Error Modeling in Stereo Navigation. *IEEE Journal of Robotics and Automation*, 1987.

Larry Matthies, Takeo Kanade, and Richard Szeliski. Kalman Filter-based Algorithms for Estimating Depth from Image Sequences. *International Journal of Computer Vision*, (3):209 – 236, 1989.

Christopher Mei, Gabe Sibley, Mark Cummins, Paul M Newman, and Ian D Reid. A Constant-Time Efficient Stereo SLAM System. In *BMVC*, pages 1–11, 2009.

M. Meilland, AI Comport, and Patrick Rives. A spherical robot-centered representation for urban navigation. In *IROS*, Oct 2010.

David R. Musser. Introspective Sorting and Selection Algorithms. *Software: Practice and Experience*, 27(8):983–993, 1997. ISSN 1097-024X.

D. Nister, O. Naroditsky, and J. Bergen. Visual odometry. In *Computer Vision and Pattern Recognition (CVPR)*, volume 1, June 2004. doi: 10.1109/CVPR.2004.1315094.

Clark F Olson, Larry H Matthies, Marcel Schoppers, and Mark W Maimone. Stereo ego-motion improvements for robust rover navigation. In *IEEE Proc. of Int'l Conf. on Robotics and Automation*, volume 2, 2001.

M. Peris, A Maki, S. Martull, Y. Ohkawa, and K. Fukui. Towards a simulation driven stereo vision system. In *Pattern Recognition (ICPR), 2012 21st International Conference on*, pages 1038–1042, Nov 2012.

D. Scaramuzza and F. Fraundorfer. Visual Odometry [Tutorial]. *Robotics Automation Magazine, IEEE*, 18:80–92, Dec 2011. ISSN 1070-9932. doi: 10.1109/MRA.2011.943233.

Laura Sevilla-Lara and Erik Learned-Miller. Distribution fields for tracking. In *Computer Vision and Pattern Recognition (CVPR), 2012 IEEE Conference on*, pages 1910–1917. IEEE, 2012.

J. Shi and C. Tomasi. Good features to track. In *Proc. of Computer Vision and Pattern Recognition (CVPR)*, pages 593–600, Jun 1994.

G. Silveira, E. Malis, and P. Rives. An efficient direct approach to visual SLAM. *IEEE Transactions on Robotics*, 2008.

F Steinbrucker, Jürgen Sturm, and Daniel Cremers. Real-time visual odometry from dense RGB-D images. In *ICCV Workshops, 2011 IEEE International Conference on*, 2011.

F. Steinbrucker, C. Kerl, D. Cremers, and J. Sturm. Large-Scale Multi-resolution Surface Reconstruction from RGB-D Sequences. In *Computer Vision (ICCV), 2013 IEEE International Conference on*, pages 3264–3271, Dec 2013. doi: 10.1109/ICCV.2013.405.

Hauke Strasdat, J. M. M. Montiel, and Andrew J. Davison. Visual SLAM: Why Filter? *Image Vision Comput.*, February 2012. ISSN 0262-8856. doi: 10.1016/j.imavis.2012.02.009.

Jürgen Sturm, Erik Bylow, Fredrik Kahl, and Daniel Cremers. CopyMe3D: Scanning and Printing Persons in 3D. In Joachim Weickert, Matthias Hein, and Bernt Schiele, editors, *Pattern Recognition*, volume 8142 of *Lecture Notes in Computer Science*, pages 405–414. Springer Berlin Heidelberg, 2013. ISBN 978-3-642-40601-0.

Richard Szeliski. *Computer Vision: Algorithms and Applications*. Springer-Verlag New York, Inc., New York, NY, USA, 1st edition, 2010. ISBN 1848829345, 9781848829343.

P.H.S. Torr and A. Zisserman. Feature Based Methods for Structure and Motion Estimation. In *Vision Algorithms: Theory and Practice*, volume 1883, pages 278–294. Springer Berlin Heidelberg, 2000. ISBN 978-3-540-67973-8. doi: 10.1007/3-540-44480-7_19.

Bill Triggs, Philip F. Mclauchlan, Richard I. Hartley, and Andrew W. Fitzgibbon. Bundle Adjustment – A Modern Synthesis. *LNCS*, January 2000.

Tommi Tykkälä, Cédric Audras, and Andrew I Comport. Direct iterative closest point for real-time visual odometry. In *ICCV 2011 Workshops*, 2011.

Tommi Tykkälä, Andrew I. Comport, Joni-Kristian Kämäräinen, and Hannu Hartikainen. Live RGB-D camera tracking for television production studios. *J. Visual Communication and Image Representation*, 25(1):207–217, 2014.

Robert J. Valkenburg, Alan M. McIvor, and P. Wayne Power. Evaluation of subpixel feature localization methods for precision measurement. In *Proc. SPIE*, volume 2350, pages 229–238, 1994.

Tobi Vaudrey, Hernan Badino, and Stefan Gehrig. Integrating Disparity Images by Incorporating Disparity Rate. In Gerald Sommer and Reinhard Klette, editors, *Robot Vision*, volume 4931 of *Lecture Notes in Computer Science*, pages 29–42. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-78156-1. doi: 10.1007/978-3-540-78157-8_3.

Tobi Vaudrey, Sandino Morales, Andreas Wedel, and Reinhard Klette. Generalised residual images' effect on illumination artifact removal for correspondence algorithms. *Pattern Recognition*, 44(9):2034 – 2046, 2011. ISSN 0031-3203. Computer Analysis of Images and Patterns.

Andreas Wedel, Clemens Rabe, Tobi Vaudrey, Thomas Brox, Uwe Franke, and Daniel Cremers. Efficient Dense Scene Flow from Sparse or Dense Stereo Data. In *European Conf. on Computer Vision*. Springer Berlin Heidelberg, 2008. ISBN 978-3-540-88681-5. doi: 10.1007/978-3-540-88682-2_56.

T. Whelan, H. Johannsson, M. Kaess, J.J. Leonard, and J. McDonald. Robust real-time visual odometry for dense RGB-D mapping. In *IEEE Proc. of Intl' Conf. on Robotics and Automation (ICRA)*, May 2013.

R. Wolke and H. Schwetlick. Iteratively Reweighted Least Squares: Algorithms, Convergence Analysis, and Numerical Comparisons. *SIAM Journal on Scientific and Statistical Computing*, 9 (5), 1988.

Zhengyou Zhang. Parameter estimation techniques: A tutorial with application to conic fitting. *Image and vision Computing*, 15(1), 1997.