

Stereo DSO: Large-Scale Direct Sparse Visual Odometry with Stereo Cameras

Rui Wang*, Martin Schwörer*, Daniel Cremers
Technical University of Munich
{wangr, schwoere, cremers}@in.tum.de

Abstract

We propose *Stereo Direct Sparse Odometry (Stereo DSO)* as a novel method for highly accurate real-time visual odometry estimation of large-scale environments from stereo cameras. It jointly optimizes for all the model parameters within the active window, including the intrinsic/extrinsic camera parameters of all keyframes and the depth values of all selected pixels. In particular, we propose a novel approach to integrate constraints from static stereo into the bundle adjustment pipeline of temporal multi-view stereo. Real-time optimization is realized by sampling pixels uniformly from image regions with sufficient intensity gradient. Fixed-baseline stereo resolves scale drift. It also reduces the sensitivities to large optical flow and to rolling shutter effect which are known shortcomings of direct image alignment methods. Quantitative evaluation demonstrates that the proposed Stereo DSO outperforms existing state-of-the-art visual odometry methods both in terms of tracking accuracy and robustness. Moreover, our method delivers a more precise metric 3D reconstruction than previous dense/semi-dense direct approaches while providing a higher reconstruction density than feature-based methods.

1. Introduction

1.1. Real-time Visual Odometry

While traditionally robotic systems such as self-driving cars have been largely relying on Laser or Lidar to actively sense their environment and perform self localization and mapping, more recently camera-based SLAM and odometry algorithms have witnessed a drastic boost in performance. Although such passive sensors require a sufficiently illuminated and textured scene to infer 3D structure and motion, they comprise several advantages including higher resolution, higher sensing range and sensing rate as well as lower weight, size and hardware cost. This makes them

*These authors contributed equally.

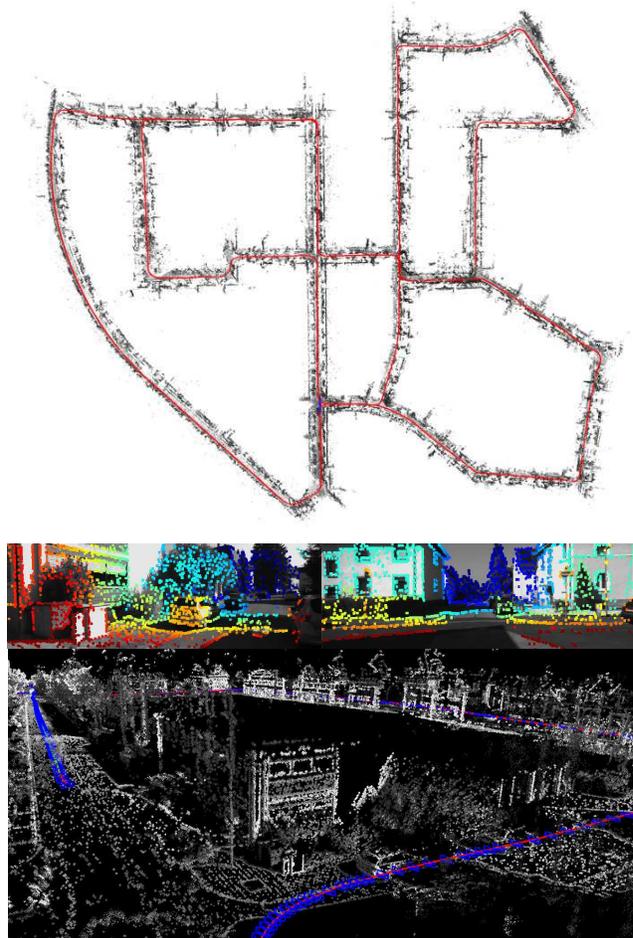


Figure 1. Results of Stereo Direct Sparse Odometry on sequence 00 of the KITTI Dataset. On top is the estimated camera trajectory. To distinguish the sparsity from previous dense or semi-dense direct methods, we show the depth maps used for tracking in the middle. An example of a reconstructed scene is shown on the bottom.

far more versatile – the smaller size and weight, for example, enable applications such as visual-inertial autonomous navigation of nanocopters [6]. As a result, there is a huge

demand for real-time capable visual odometry and visual SLAM algorithms. Among the most desirable properties are maximal accuracy, robustness (to changes in scene structure, lighting and faster motion) and density of the reconstructed environment. In this paper, we propose what we believe to be the currently most accurate and robust real-time visual odometry method.

1.2. Related Work

The first real-time capable visual SLAM and odometry systems were pioneered around 2000 [2, 24, 5, 3]. The key idea underlying these structure and motion techniques is to select a set of keypoints (typically corner-like structures), track them across frames and jointly infer their 3D location and the camera motion. More recently, we have witnessed a boost of increasingly performant solutions stemming from numerous advances both in computing hardware and in algorithmic sophistication. For example, Klein and Murray [17] parallelized the motion and 3D structure estimations. Strasdat et al. proposed to expand the concept of keyframes to integrate scale [26] and proposed a double window optimization [25]. Mei et al. [20] developed a relative SLAM approach for constant-time estimation. More recently, ORB-SLAM [21, 22] introduced an efficient visual SLAM solution based on ORB features and innovations such as map reuse. It has gained a lot of popularity due to its high tracking accuracy and robustness and is among the state-of-the-art methods for visual SLAM.

While the traditional structure and motion/visual SLAM algorithms were based on heuristically selected keypoints, more recently a number of so-called direct methods were proposed [27, 23, 10, 12, 8]. These aim at computing geometry and motion directly from the images thereby skipping the intermediate keypoint selection step. Algorithmically they typically rely on robust cost functions and optimization through Gauss-Newton iteration as done also for RGB-D based SLAM in [14, 15, 18]. While the methods [27, 23] rely on variational methods and total variation regularization to generate dense reconstructions in real-time (on powerful GPUs), the latter works [10, 12] abstain from dense solutions in order to not oversmooth or hallucinate geometric structures but rather generate semi-dense or sparse reconstructions without the need for GPU support. The extension of direct monocular SLAM algorithms to large-scale environments was proposed in LSD-SLAM [8, 9]. The key idea is to incrementally track the camera and simultaneously perform a pose graph optimization in order to keep the entire camera trajectory globally consistent. While this kind of stripped-down version of bundle adjustment (which includes motion, but excludes structure) does not remove the drift, it appears to spread it out across the computed trajectory. Since then, the semi-dense direct VO/SLAM has been further extended by support for omnidirectional cameras [1]

and tightly coupling with IMU [28]. A rolling shutter calibration method for LSD-SLAM has been recently proposed in [16].

While we are witnessing an ongoing competition between keypoint based algorithms and direct algorithms, recently Direct Sparse Odometry (DSO) [7] was shown to outperform the state-of-the-art keypoint based monocular SLAM algorithm ORB-SLAM [21] in terms of both accuracy and robustness on a fairly large dataset for monocular camera tracking [11].

While this seems to indicate a certain advantage of direct methods, DSO has several shortcomings as a technique for visual odometry or SLAM: Firstly, the mentioned performance gain was demonstrated on a photometrically calibrated dataset. In the absence of this photometric calibration (many datasets do not provide it), the performance of direct methods like DSO substantially degrades. Secondly, being a pure monocular system, DSO invariably cannot estimate the scale of the reconstructed scene or the units of camera motion. Furthermore, the estimated trajectory suffers from substantial scale drift such that even manually providing the best scale does not resolve the problem – see Fig 6. Thirdly, as shown in [7] DSO is quite sensitive to geometric distortions as those induced by fast motion and rolling shutter cameras. While techniques for calibrating rolling shutter exist for direct SLAM algorithms [16], these are often quite involved and far from real-time capable.

1.3. Contribution

In this work, we propose *Stereo DSO* as a novel direct visual odometry method for highly accurate and robust motion and 3D structure estimation in real-time from a moving stereo camera. It addresses the aforementioned shortcomings of previous approaches by leveraging the additional sensor information. Thus it provides an accurate and (due to the stereo initialization) much faster converging scale estimation and is less sensitive to missing photometric calibration or the effects of rolling shutter. In particular:

- We derive a stereo version of DSO. To this end, we detail the proposed combination of temporal multi-view stereo and static stereo and their integration with marginalization using the Schur complement. Unlike previous extension of monocular direct approach [8] to stereo [9] (both apply filtering approaches to the geometry that do not involve bundle adjustment), we propose a novel way to extend the energy function and the entire bundle adjustment procedure in a manner that real-time capability is assured.
- We perform systematic quantitative evaluations on the KITTI dataset and on the Cityscapes dataset. Comparisons to alternative methods like Stereo ORB-SLAM and Stereo LSD-SLAM demonstrate that the proposed

Stereo DSO is superior to these techniques, in particular when evaluated on the KITTI testing set, indicating that the method generalizes better to unknown settings.

- A quantitative evaluation over longer ranges demonstrates that the proposed Stereo DSO algorithm without loop closure optimization outperforms Stereo ORB-SLAM which applies global pose graph optimization and bundle adjustment.

2. Direct Sparse VO with Stereo Cameras

Our Stereo DSO is a system that combines static stereo with multi-view stereo. As is demonstrated in [9], such hybrid approach brings several advantages over each of the separate one:

- Absolute scale can be directly calculated from static stereo from the known baseline of the stereo camera.
- Static stereo can provide initial depth estimation for multi-view stereo.
- Due to the fixed baseline, static stereo can only accurately triangulate 3D points within a limited depth range. This limit is resolved by temporal multi-view stereo.
- They can complement each other in degenerate cases where edges are parallel to epipolar lines.

An overview of our system is shown in Fig 2. Instead of using random depth for initialization [10, 8, 7], our system uses depth estimation from static stereo matching (Sec 2.3). Based on the direct image alignment formulation (Sec 2.2), new stereo frames are first tracked with respect to their reference keyframe in a coarse-to-fine manner (Sec 2.3). The obtained pose estimate is used to refine the depth of recently selected points. Then our system checks whether a new keyframe is needed by the current active window. If not, a non-keyframe will be created, otherwise a new keyframe will be generated and added to the active window (Sec 2.4). For all keyframes in the active window, a joint optimization of their poses, affine brightness parameters, as well as the depths of all the observed 3D points and camera intrinsics is performed. To maintain the size of the active window, old keyframes and 3D points are marginalized out using the Schur complement (Sec 2.5).

2.1. Notation

Throughout this paper we use light, bold lower-case letters and bold upper-case letters to denote scalars (u), vectors (\mathbf{t}) and matrices (\mathbf{R}) respectively. Light upper-case letters are used to represent functions (I).

Camera calibration matrices are denoted by \mathbf{K} . Camera poses are represented by matrices of the special Euclidean group $\mathbf{T}_i \in SE(3)$, which transform a 3D coordinate from

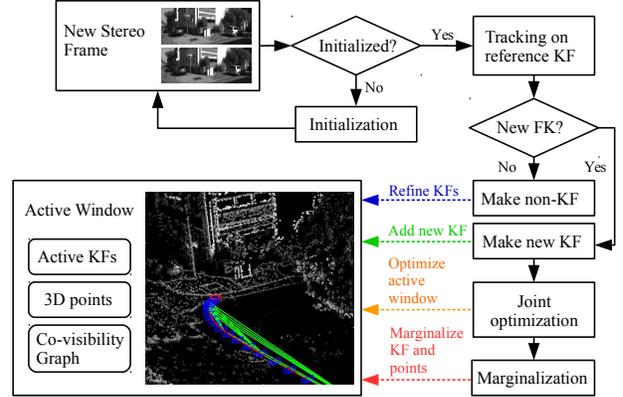


Figure 2. System overview.

the camera coordinate system to the world coordinate system. $\Pi_{\mathbf{K}}$ and $\Pi_{\mathbf{K}}^{-1}$ are used to denote camera projection and back-projection functions. In this paper, a 3D point is represented by its image coordinate \mathbf{p} and inverse depth $d_{\mathbf{p}}$ relative to its host keyframe. The host keyframe is the frame the point got selected from. The inverse depth parameterization has been demonstrated to be advantageous when errors in images are modeled as Gaussian distributions [5, 3].

2.2. Direct Image Alignment Formulation

Suppose a point set \mathcal{P}_i in a reference frame I_i is observed in another frame I_j , the basic idea of direct image alignment can be formulated as

$$E_{ij} = \sum_{\mathbf{p} \in \mathcal{P}_i} \omega_{\mathbf{p}} \left\| I_j[\mathbf{p}'] - I_i[\mathbf{p}] \right\|_{\gamma}, \quad (1)$$

where $\|\cdot\|_{\gamma}$ is the Huber norm and $\omega_{\mathbf{p}}$ is a weighting that down-weights high image gradients

$$\omega_{\mathbf{p}} = \frac{c^2}{c^2 + \|\nabla I_i(\mathbf{p})\|_2^2}, \quad (2)$$

with some constant c . \mathbf{p}' is the projection of \mathbf{p} in I_j calculated by

$$\mathbf{p}' = \Pi_{\mathbf{K}}(\mathbf{T}_{ji} \Pi_{\mathbf{K}}^{-1}(\mathbf{p}, d_{\mathbf{p}})), \quad (3)$$

with $d_{\mathbf{p}}$ the inverse depth of \mathbf{p} and \mathbf{T}_{ji} the transformation that transforms a point from frame i to frame j :

$$\mathbf{T}_{ji} := \begin{bmatrix} \mathbf{R}_{ji} & \mathbf{t} \\ 0 & 1 \end{bmatrix} = \mathbf{T}_j^{-1} \mathbf{T}_i. \quad (4)$$

Conventional direct methods, both dense and semi-dense, tend to use as many pixels from each image as possible. While bringing heavy computational burden to the system, its benefit saturates fast. Therefore, in [7] the authors proposed a strategy to select a fixed number of points from each frame, uniformly across all the regions with sufficient gradient. For each selected point, a small neighborhood around it is used to calculate the photometric error in

(1). In this paper we follow the same approach, but use the stereo image pair to verify the selected points and assist the depth initialization. More details are provided in Sec 2.4.

As the photometric error is calculated directly on pixel intensities, it is very sensitive to sudden illumination changes between consecutive frames. Ideally the exposure time of each frame, as well as the camera response function (which can be highly non-linear) are directly accessible from the hardware [11], which can be used to correct such effect. When this information is not available (as for most existing datasets), similar to [7], we introduce two parameters a_i, b_i for each image to model an affine brightness change. The energy function in (1) is then modified to

$$E_{ij} = \sum_{\mathbf{p} \in \mathcal{P}_i} \sum_{\tilde{\mathbf{p}} \in \mathcal{N}_{\mathbf{p}}} \omega_{\tilde{\mathbf{p}}} \left\| I_j[\tilde{\mathbf{p}}'] - b_j - \frac{e^{a_j}}{e^{a_i}} (I_i[\tilde{\mathbf{p}}] - b_i) \right\|_{\gamma}, \quad (5)$$

with $\mathcal{N}_{\mathbf{p}}$ the 8-point pattern of \mathbf{p} as defined in [7] and $\tilde{\mathbf{p}}'$ the projection of the pattern point $\tilde{\mathbf{p}}$ into I_j . a_i, a_j, b_i, b_j are estimated in the windowed optimization as is shown in Sec 2.5.

2.3. Tracking

Each time a new stereo frame is fed into the system, direct image alignment [10] is used to track it with respect to the newest keyframe in the active window: a constant motion model is used to assign an initial pose to the new frame to be tracked. All the points inside the active window are projected into the new frame. Then the pose of the new frame is optimized by minimizing the energy function (5) while keeping the depth values fixed. The optimization is performed with Gauss-Newton on an image pyramid in a coarse-to-fine order.

To initialize the whole system, i.e., to track the second frame with respect to the initial one using (5), the inverse depth values of the points in the first frame are required. Previous monocular direct VO approaches use random depth values for initialization [10, 8, 7], thus usually need a certain pattern of the initial camera movement. In this work we use static stereo matching to estimate a semi-dense depth map for the first frame. As at this stage the affine brightness transfer factors between the stereo image pair are unknown, correspondences are searched along the horizontal epipolar line using the NCC of the 3×5 neighborhood.

2.4. Frame Management

If a new stereo frame is successfully tracked, we use the same criteria as in [7] to determine if a new keyframe is required. The basic idea is to check if the scene or the illumination has sufficiently changed. Scene changing is evaluated by the mean squared optical flow, as well as the mean squared optical flow without rotation between the current

frame and the last keyframe in the active window. Illumination change is quantized by the relative brightness factor $|a_j - a_i|$.

To create a new keyframe, a sparse set of points is selected from the image, which will be called candidate points in the rest of the paper. To select points evenly distributed across the image and only points that have sufficient image gradient, the image is divided into small blocks and for each block an adaptive threshold is calculated. Instead of using square blocks of fixed size [7], we use blocks with size that is proportional to the image size. We find this helpful for images with dissimilar width and height like the ones from KITTI. A point is selected if it surpasses the threshold of the block and it has the largest absolute gradient in its neighborhood.

Before a candidate point is activated and optimized in the windowed optimization, its inverse depth is constantly refined by the following non-keyframes. In the monocular case, the candidate point is usually initialized to have a depth range from 0 to infinity, corresponding to a large depth variance. In our case, we use static stereo matching with NCC to obtain a better depth initialization for the candidate points, which significantly increases the tracking accuracy.

When old points are removed from the active window by marginalization (Sec 2.5), candidate points are activated and added to the joint optimization. Each activated point is hosted in one keyframe and is observed by several other keyframes in the active window. Each time an active point is observed in another keyframe, it creates a photometric energy factor defined as the inner part of (5):

$$E_{ij}^{\mathbf{p}} = \omega_{\mathbf{p}} \left\| I_j[\mathbf{p}'] - b_j - \frac{e^{a_j}}{e^{a_i}} (I_i[\mathbf{p}] - b_i) \right\|_{\gamma}. \quad (6)$$

For simplicity of notation, we omit the summation over the neighborhood $\mathcal{N}_{\mathbf{p}}$ in the above equation as well as in the rest of this paper. A factor graph of the energy function is shown in Fig 3, where each factor (represented as small square in the middle) depends on the inverse depth of the point, the cameras poses of the host keyframe and the keyframe that observes this point, as well as their affine brightness correction factors. The constraints from static stereo (denoted with red lines) introduce scale information into the system. Moreover, they also provide good geometric priors to temporal multi-view stereo.

2.5. Windowed Optimization

Putting all the energy factors together, the final energy function to be minimized in the windowed optimization is

$$E = \sum_{i \in \mathcal{F}} \sum_{\mathbf{p} \in \mathcal{P}_i} \sum_{j \in \text{obs}(\mathbf{p})} E_{ij}^{\mathbf{p}}, \quad (7)$$

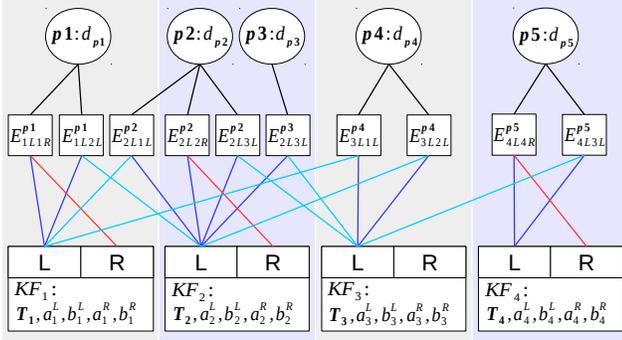


Figure 3. Factor graph of the energy function. In this example, 5 points are observed by 4 keyframes. Each energy factor is related to one point and two keyframes, thus depends on the inverse depth of the point, the camera poses of the two keyframes and their affine brightness correction factors, as well as the camera intrinsic parameters (we assume the same for the left and the right cameras, and omit it here for simplicity). Constraints from host keyframes and static stereo are shown in dark blue and red respectively. Remaining constraints in light blue are the ones from the keyframes the points are observed.

where \mathcal{F} is the set of the keyframes in the current window and $obs(\mathbf{p})$ is the set of the keyframes in \mathcal{F} that can observe \mathbf{p} . The energy is optimized iteratively using Gauss-Newton algorithm:

$$\delta \xi = -(\mathbf{J}^T \mathbf{W} \mathbf{J})^{-1} \mathbf{J}^T \mathbf{W} \mathbf{r}, \quad (8)$$

$$\xi^{new} = \delta \xi \boxplus \xi, \quad (9)$$

where \mathbf{r} contains the stacked residuals, \mathbf{J} is the Jacobian and \mathbf{W} is the diagonal weight matrix. The parameters we want to optimize are enclosed in

$$\xi = (\mathbf{T}_{0, \dots, N_f-1}, d_{0, \dots, N_p-1}, \mathbf{c}, a_{0, \dots, N_f-1}^L, b_{0, \dots, N_f-1}^L, a_{0, \dots, N_f-1}^R, b_{0, \dots, N_f-1}^R), \quad (10)$$

where \mathbf{c} is the vector containing the global camera intrinsics, L and R denote the parameter of the left and right camera frame, and N_f and N_p are the numbers of keyframes and active points in the current window, respectively. The \boxplus -operator on \mathbf{T} is defined using Lie Algebra $se(3)$ as

$$\boxplus : se(3) \times SE(3) \rightarrow SE(3), \mathbf{x} \boxplus \mathbf{T} := exp(\hat{\mathbf{x}}) \mathbf{T}, \quad (11)$$

whereas on the rest parameters it is simply the conventional addition.

Temporal Multi-View Stereo. Each residual from temporal multi-view stereo is defined as

$$r_k^t = I_j[\mathbf{p}'(\mathbf{T}_i, \mathbf{T}_j, d, \mathbf{c})] - b_j - \frac{e^{a_j}}{e^{a_i}} (I_i[\mathbf{p}] - b_i). \quad (12)$$

Similar to [7] the Jacobian is defined as

$$\mathbf{J}_k^t = \begin{bmatrix} \frac{\partial I_j}{\partial \mathbf{p}'} & \frac{\partial \mathbf{p}'(\delta \xi \boxplus \xi)}{\partial \delta \xi_{geo}^t} & \frac{\partial r_k}{\partial r_k(\delta \xi \boxplus \xi)} \end{bmatrix}, \quad (13)$$

where the geometric parameters ξ_{geo}^t are $(\mathbf{T}_i, \mathbf{T}_j, d, \mathbf{c})$ and the photometric parameters $\xi_{photo}^t = (a_i, a_j, b_i, b_j)$.

Static Stereo. For static stereo the residual is modified to

$$r_k^s = I_i^R[\mathbf{p}'(\mathbf{T}_{ji}, d, \mathbf{c})] - b_i^R - \frac{e^{a_i^R}}{e^{a_i^L}} (I_i[\mathbf{p}] - b_i^L). \quad (14)$$

The Jacobian has the same form as in (13) but now with less geometric parameters $\xi_{geo} = (d, \mathbf{c})$, because the relative transformation between the left and right cameras \mathbf{T}_{ji} is fixed. Therefore, \mathbf{T}_{ji} is not optimized in the windowed optimization.

Stereo Coupling. To balance the relative weights of temporal multi-view and static stereo, we introduce a coupling factor λ to weight the constraints from static stereo differently. The energy function in (7) thus can be further formulated as

$$E = \sum_{i \in \mathcal{F}} \sum_{\mathbf{p} \in \mathcal{P}_i} \left(\sum_{j \in obs^t(\mathbf{p})} E_{ij}^{\mathbf{p}} + \lambda E_{is}^{\mathbf{p}} \right), \quad (15)$$

where $obs^t(\mathbf{p})$ are the observations of \mathbf{p} from temporal multi-view stereo, and $E_{is}^{\mathbf{p}}$ the energy belonging to the static stereo residuals. The effects of the coupling factor are detailed in Sec 3.1.

Marginalization. To keep the active window of bounded size, old keyframes are removed by marginalization using the Schur complement [19, 7]. Before marginalizing a keyframe, we first marginalize all active points that are not observed by the two latest keyframes together with all active points hosted in the keyframe. Afterwards, the keyframe is marginalized and moved out of the active window. Let $\mathbf{H} = \mathbf{J}^T \mathbf{W} \mathbf{J}$ and $\mathbf{b} = \mathbf{J}^T \mathbf{W} \mathbf{r}$ be the Gauss-Newton system containing only the variables to marginalize and the variables connected to them in the factor graph. If we use α and β to respectively denote the variables to keep and to marginalize, the Gauss-Newton system can be rearranged to

$$\begin{bmatrix} \mathbf{H}_{\alpha\alpha} & \mathbf{H}_{\alpha\beta} \\ \mathbf{H}_{\beta\alpha} & \mathbf{H}_{\beta\beta} \end{bmatrix} \begin{bmatrix} \mathbf{x}_\alpha \\ \mathbf{x}_\beta \end{bmatrix} = \begin{bmatrix} \mathbf{b}_\alpha \\ \mathbf{b}_\beta \end{bmatrix}. \quad (16)$$

Multiplying the second line by $\mathbf{H}_{\alpha\beta} \mathbf{H}_{\beta\beta}^{-1}$ and subtracting it from the first leads to

$$\underbrace{(\mathbf{H}_{\alpha\alpha} - \mathbf{H}_{\alpha\beta} \mathbf{H}_{\beta\beta}^{-1} \mathbf{H}_{\beta\alpha}^T)}_{\hat{\mathbf{H}}_\alpha} \mathbf{x}_\alpha = \underbrace{\mathbf{b}_\alpha - \mathbf{H}_{\alpha\beta} \mathbf{H}_{\beta\beta}^{-1} \mathbf{b}_\beta}_{\hat{\mathbf{b}}_\alpha}. \quad (17)$$

The resulting system $\hat{\mathbf{H}}_\alpha \mathbf{x}_\alpha = \hat{\mathbf{b}}_\alpha$ only depends on the variables to keep \mathbf{x}_α and is added as prior to the subsequent optimizations.

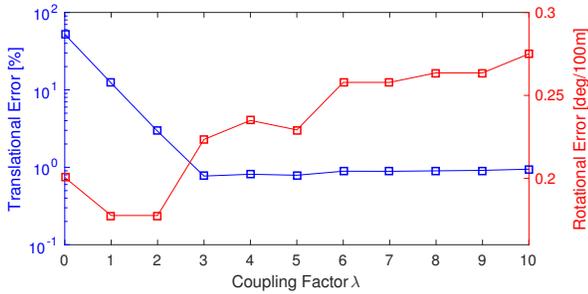


Figure 4. Average translational and rotational errors on Seq. 06 for different coupling factors.

3. Evaluation

We evaluate our method on two popular datasets: the KITTI Odometry Benchmark and the Cityscapes Dataset. Both datasets provide synchronized stereo sequences with rectified high resolution images. We compare our method thoroughly with state-of-the-art stereo VO methods, both feature-based and direct method on KITTI. On Cityscapes, we create several sequences with ground truth camera poses calculated from the provided GPS coordinates. We show both the tracking and 3D reconstruction results on this dataset and demonstrate that our method can be used for large-scale camera tracking and 3D reconstruction.

3.1. KITTI Visual Odometry Benchmark

We evaluate our method on the KITTI Odometry Benchmark [13], where altogether 22 driving sequences are provided. The first 11 (00-10) sequences are provided with ground truth 6D poses as training set, whereas the latter 11 sequences comprise the testing set.

We first test the influence of the stereo coupling factor on an example sequence (Seq. 06). The translational and rotational errors obtained by using different coupling factors are shown in Fig 4: Introducing constraints from static stereo with certain weightings ($\lambda = 1, 2$) significantly reduces both translational and rotational errors. Further increasing the weighting ($\lambda > 3$) makes the method more sensitive to incorrect matchings from static stereo and thus degrade the performance. The estimated trajectories for $\lambda = 0 - 3$ are shown in the supplementary material.

The comparison of the VO accuracy of different stereo methods on the training set are shown in Table 1. We compare our method to Stereo LSD-SLAM and ORB-SLAM2, which are currently the state-of-the-art direct and feature-based stereo VO methods respectively. The results for Stereo LSD-SLAM are cited from [9] (VO only), while the ones for ORB-SLAM2 are obtained by running their code with default settings. For fair comparison we turned off its loop-closure detection and global bundle adjustment. It can be seen that our results are almost always better than LSD-SLAM. Compared to ORB-SLAM2, all our rotational er-

	St. DSO		ORB-SLAM2		St. LSD-VO	
Seq.	t_{rel}	r_{rel}	t_{rel}	r_{rel}	t_{rel}	r_{rel}
00	0.84	0.26	0.83	0.29	1.09	0.42
01	1.43	0.09	1.38	0.20	2.13	0.37
02	0.78	0.21	0.81	0.28	1.09	0.37
03	0.92	0.16	0.71	0.17	1.16	0.32
04	0.65	0.15	0.45	0.18	0.42	0.34
05	0.68	0.19	0.64	0.26	0.90	0.34
06	0.67	0.20	0.82	0.25	1.28	0.43
07	0.83	0.36	0.78	0.42	1.25	0.79
08	0.98	0.25	1.07	0.31	1.24	0.38
09	0.98	0.18	0.82	0.25	1.22	0.28
10	0.49	0.18	0.58	0.28	0.75	0.34
mean	0.84	0.20	0.81	0.26	1.14	0.40

Table 1. Comparison of accuracy on KITTI training set. t_{rel} translational RMSE (%), r_{rel} rotational RMSE (degree per 100m). Both are average over 100m to 800m intervals. Best results are shown as bold numbers.

rors are better, but translational errors are slightly mixed. We claim that this might result from the relatively low frame rate of the dataset.

In Fig 5 we show our results on the testing set. We show the same plots as recommended by the benchmark, where translational errors and rotational errors with respect to different distance intervals and driving speeds over the entire set are plotted. It is worth noting that the results for the other two methods are obtained by their SLAM system with loop closure (for ORB-SLAM2 also with global bundle adjustment), while ours are from pure VO. As can be seen from the plots, our method performs the best under all settings.

To show the benefit of using a stereo camera, Fig 6 qualitatively compares the results of different monocular VO methods with our method on the sequences 00 and 06. Both state-of-the-art feature-based and direct monocular VO methods can not handle the scale drift properly. Results on all training sequences can be found in the supplementary material.

3.2. Cityscapes Dataset

To evaluate our method on more realistic data, we further run our Stereo DSO on the Cityscapes Dataset [4]. Although this dataset is dedicatedly designed for scene understanding and image segmentation, it also provides a long sequence (over 100,000 frames) captured using a stereo camera system. We choose this dataset as it provides industrial level images with high dynamic-range (HDR) and fairly high frame rate (17Hz). Moreover, in contrast to most existing datasets that provide street view sequences with global shutters, the images of Cityscapes were captured using rolling shutter cameras, which has been considered as a main challenge to direct methods. Although we do not address the rolling shutter calibration problem specifically in

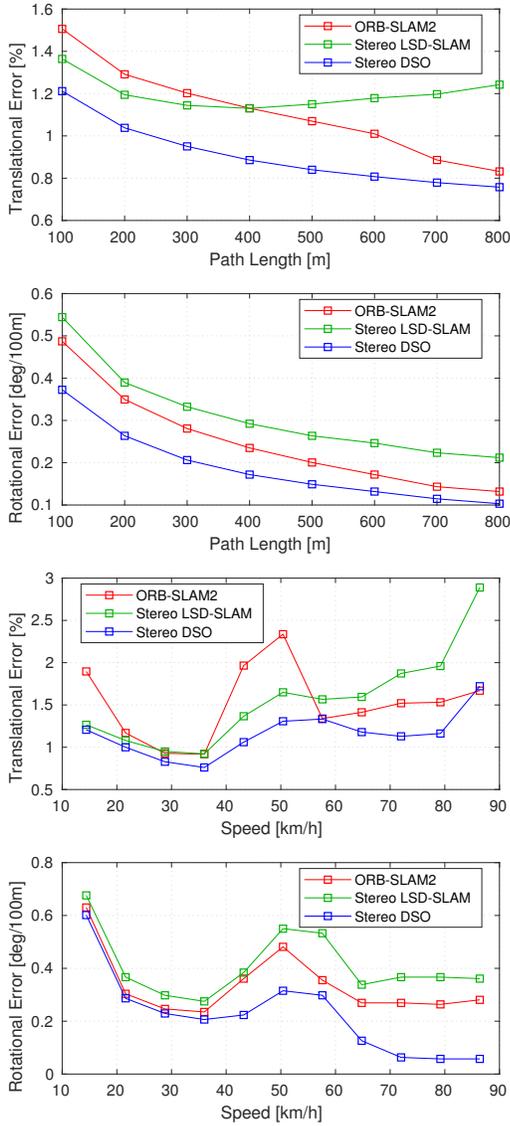


Figure 5. Average translational and rotational errors with respect to driving intervals (top two) and driving speed (bottom two) on KITTI testing set (Seq. 11-21). In all the cases, our VO results (without loop closure) are better than the SLAM results (with loop closure, for ORB-SLAM2 also with global bundle adjustment) of LSD-SLAM and ORB-SLAM2. It is surprising to see that even for large intervals our method performs better, as in such cases loop closure usually reduces the errors significantly.

this paper, our method implicitly makes use of two advantages of modern stereo camera systems: Firstly, they have very fast pixel clock which reduces the rolling shutter effect; Secondly, the corresponding rows of the left and right images are synchronized. Therefore, static stereo can compensate the errors introduced by the rolling shutter effect across multiple views (multi-view stereo).

As for this dataset only inaccurate GPS/vehicle odometry information is provided, it is hard to evaluate the VO per-

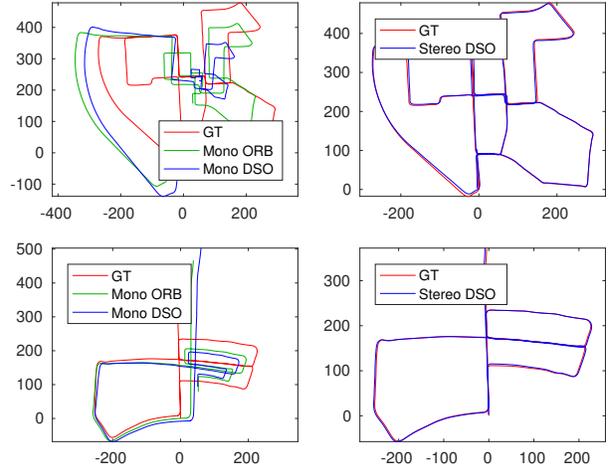


Figure 6. Qualitative results on KITTI sequences 00 (top) and 05 (bottom). For the monocular VO results we perform a similarity alignment to the ground truth.

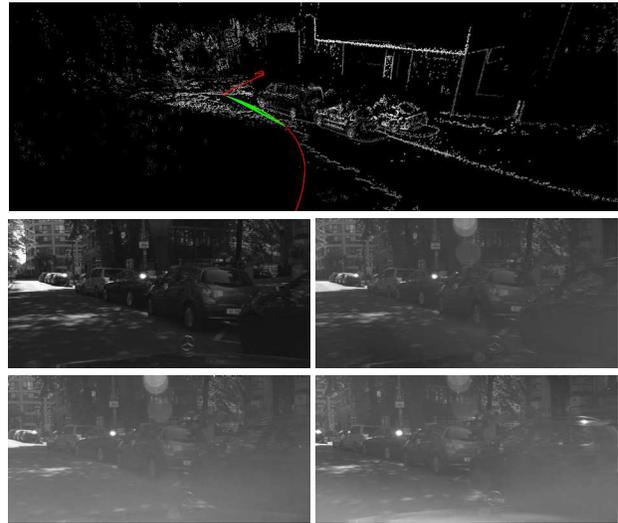


Figure 7. An example of severe brightness change due to camera auto exposure/gain control. Despite the affine brightness correction our method still fails here due to the extreme brightness change as well as the strong rotational motion of the car.

formance accurately. Besides, this dataset has lots of moving objects such as cars driving right in front of the camera as well as severe uncalibrated brightness changes (Fig 7). Although our method has considered brightness changes in the energy function, it seems the simple affine brightness modeling is not sufficient to handle extreme cases. Our method fails in the scene as shown in Fig 7, where the severe brightness change is combined with a strong rotational movement of the car.

To evaluate our tracking accuracy, we divide the long sequence into several smaller ones with length of 3000 to 6000 frames. For each small sequence, we calculate the

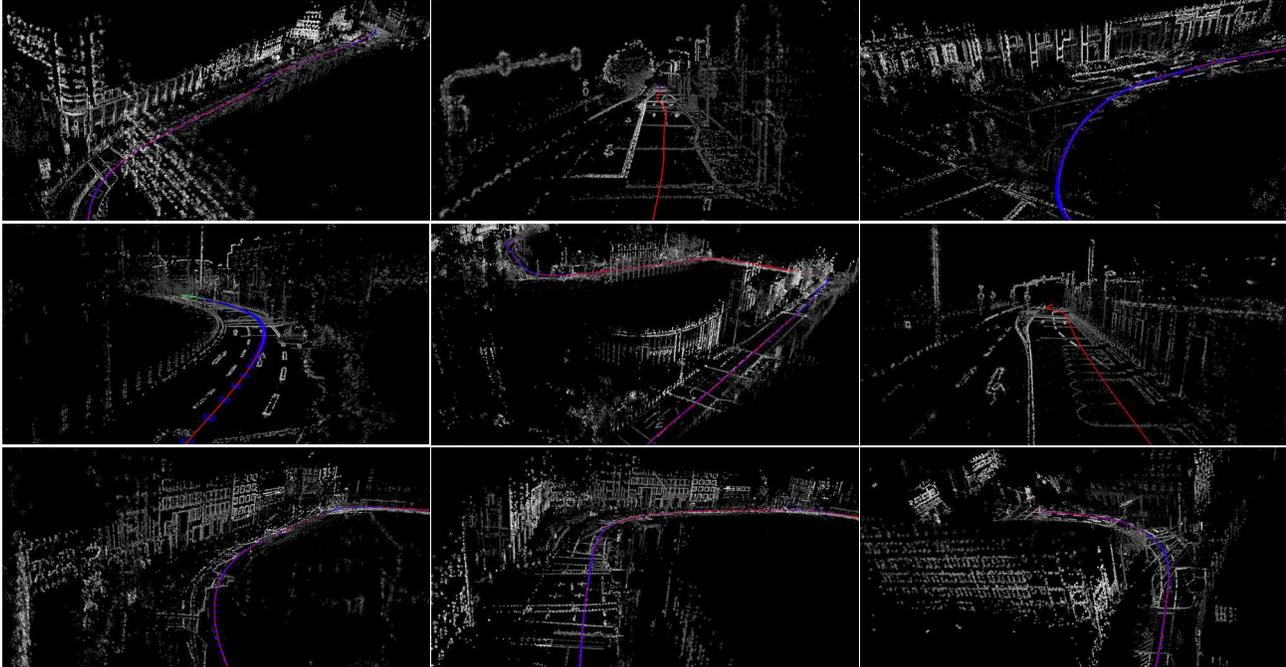


Figure 8. Examples of qualitative results on the Cityscapes Dataset.

ground truth camera poses from the GPS coordinates using the Mercator projection and align these poses to our trajectories using a $SE(3)$ transformation. Some results of the estimated camera trajectories can be found in Fig 9. As a big advantage of our method over the feature-based methods, our VO approach creates precise and much denser 3D reconstructions. Fig 8 shows some reconstruction results on the Frankfurt sequence. Although the reconstructions are sparser than the ones from previous dense or semi-dense approaches, they are much more accurate due to the bundle adjustment in the windowed optimization. More evaluation results can be found in the supplementary material.

4. Conclusion

In this work, we introduced Stereo Direct Sparse Odometry as a direct large-scale capable method for accurately tracking and mapping from a stereo camera in real-time. We detailed the technical implementation including the integration of temporal multi-view stereo and static stereo within a marginalization framework using the Schur complement. Thorough qualitative and quantitative evaluations on the KITTI dataset and the Cityscapes dataset demonstrate that Stereo DSO is the currently most accurate and robust method for tracking a stereo camera in challenging real-world scenarios. In particular, an evaluation on the KITTI testing set showed that even without closing large loops, Stereo DSO provides more accurate results than Stereo ORB-SLAM2 with loop closing and global bundle adjustment.

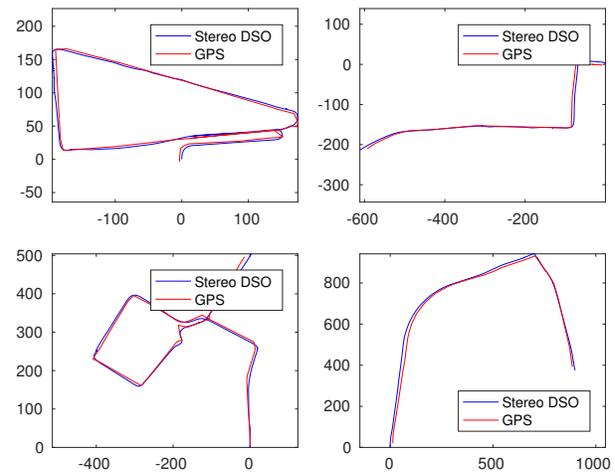


Figure 9. Estimated trajectories on the Cityscapes Frankfurt stereo sequence. The sequences are obtained by dividing the long sequence into several smaller segments of 5000 to 6000 frames.

In future work, we plan to extend our approach to a full SLAM system by adding loop closing and a database for map maintenance. Besides, we also consider explicit dynamic object handling to further boost the VO accuracy and robustness.

Acknowledgments. This work was supported by the ERC Consolidator Grant “3D Reloaded”. We would like to thank Jakob Engel and Vladyslav Usenko for their supports and fruitful discussions.

References

- [1] D. Caruso, J. Engel, and D. Cremers. Large-scale direct SLAM for omnidirectional cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 141–148. IEEE, 2015.
- [2] A. Chiuso, P. Favaro, H. Jin, and S. Soatto. Structure from motion causally integrated over time. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 24(4):523–535, 2002.
- [3] J. Civera, A. J. Davison, and J. M. Montiel. Inverse depth parametrization for monocular SLAM. *IEEE Transactions on Robotics*, 24(5):932–945, 2008.
- [4] M. Cordts, M. Omran, S. Ramos, T. Rehfeld, M. Enzweiler, R. Benenson, U. Franke, S. Roth, and B. Schiele. The cityscapes dataset for semantic urban scene understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 3213–3223, 2016.
- [5] A. J. Davison, I. D. Reid, N. D. Molton, and O. Stasse. MonoSLAM: Real-time single camera SLAM. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 29(6), 2007.
- [6] O. Dunkley, J. Engel, J. Sturm, and D. Cremers. Visual-inertial navigation for a camera-equipped 25g nano-quadrotor. In *IROS2014 Aerial Open Source Robotics Workshop*, 2014.
- [7] J. Engel, V. Koltun, and D. Cremers. Direct sparse odometry. *arXiv preprint arXiv:1607.02565*, 2016.
- [8] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *European Conference on Computer Vision*, pages 834–849. Springer, 2014.
- [9] J. Engel, J. Stückler, and D. Cremers. Large-scale direct SLAM with stereo cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1935–1942, 2015.
- [10] J. Engel, J. Sturm, and D. Cremers. Semi-dense visual odometry for a monocular camera. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 1449–1456, 2013.
- [11] J. Engel, V. Usenko, and D. Cremers. A photometrically calibrated benchmark for monocular visual odometry. In *arXiv:1607.02555*, July 2016.
- [12] C. Forster, M. Pizzoli, and D. Scaramuzza. SVO: Fast Semi-Direct Monocular Visual Odometry. In *IEEE International Conference on Robotics and Automation (ICRA)*, 2014.
- [13] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun. Vision meets robotics: The KITTI dataset. *The International Journal of Robotics Research*, 32(11):1231–1237, 2013.
- [14] C. Kerl, J. Sturm, and D. Cremers. Dense visual SLAM for RGB-D cameras. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 2100–2106. IEEE, 2013.
- [15] C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for RGB-D cameras. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 3748–3754. IEEE, 2013.
- [16] J.-H. Kim, C. Cadena, and I. Reid. Direct semi-dense SLAM for rolling shutter cameras. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1308–1315. IEEE, 2016.
- [17] G. Klein and D. Murray. Parallel tracking and mapping for small ar workspaces. In *IEEE/ACM International Symposium on Mixed and Augmented Reality (ISMAR)*, pages 225–234. IEEE, 2007.
- [18] S. Klose, P. Heise, and A. Knoll. Efficient compositional approaches for real-time robust direct visual odometry from RGB-D data. In *IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 1100–1106. IEEE, 2013.
- [19] S. Leutenegger, S. Lynen, M. Bosse, R. Siegwart, and P. Furgale. Keyframe-based visual-inertial odometry using non-linear optimization. *The International Journal of Robotics Research*, 34(3):314–334, 2015.
- [20] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid. RSLAM: A system for large-scale mapping in constant-time using stereo. *International Journal of Computer Vision*, 94(2):198–214, 2011.
- [21] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardos. ORB-SLAM: a versatile and accurate monocular SLAM system. *IEEE Transactions on Robotics*, 31(5):1147–1163, 2015.
- [22] R. Mur-Artal and J. D. Tardos. ORB-SLAM2: an Open-Source SLAM System for Monocular, Stereo and RGB-D Cameras. *arXiv preprint arXiv:1610.06475*, 2016.
- [23] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. Dtm: Dense tracking and mapping in real-time. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2320–2327. IEEE, 2011.
- [24] D. Nistér, O. Naroditsky, and J. Bergen. Visual odometry. In *Proceedings of the 2004 IEEE Computer Society Conference on Computer Vision and Pattern Recognition, 2004. CVPR 2004.*, volume 1, pages I–I. IEEE, 2004.
- [25] H. Strasdat, A. J. Davison, J. M. Montiel, and K. Konolige. Double window optimisation for constant time visual SLAM. In *2011 IEEE International Conference on Computer Vision (ICCV)*, pages 2352–2359. IEEE, 2011.
- [26] H. Strasdat, J. Montiel, and A. J. Davison. Scale drift-aware large scale monocular SLAM. *Robotics: Science and Systems (RSS)*, 2010.
- [27] J. Stühmer, S. Gumhold, and D. Cremers. Real-time dense geometry from a handheld camera. In *Joint Pattern Recognition Symposium*, pages 11–20. Springer, 2010.
- [28] V. Usenko, J. Engel, J. Stückler, and D. Cremers. Direct visual-inertial odometry with stereo cameras. In *IEEE International Conference on Robotics and Automation (ICRA)*, pages 1885–1892. IEEE, 2016.