Reconstructing Street-Scenes in Real-Time From a Driving Car

Vladyslav Usenko, Jakob Engel, Jörg Stückler, and Daniel Cremers
Technische Universität München, {usenko, engelj, stueckle, cremers}@in.tum.de

Abstract

Most current approaches to street-scene 3D reconstruction from a driving car to date rely on 3D laser scanning or tedious offline computation from visual images. In this paper, we compare a real-time capable 3D reconstruction method using a stereo extension of large-scale direct SLAM (LSD-SLAM) with laser-based maps and traditional stereo reconstructions based on processing individual stereo frames. In our reconstructions, small-baseline comparison over several subsequent frames are fused with fixed-baseline disparity from the stereo camera setup. These results demonstrate that our direct SLAM technique provides an excellent compromise between speed and accuracy, generating visually pleasing and globally consistent semi-dense reconstructions of the environment in real-time on a single CPU.

1. Introduction

The 3D reconstruction of environments has been a research topic in computer vision for many years with many applications in robotics or surveying. In particular this problem is essential for creating self-driving vehicles, since 3D maps are required for localization and obstacle detection.

In the beginning, laser-based distance sensors were primarily used for 3D reconstruction [4]. However, these solutions have several drawbacks. Laser scanners usually are quiet expensive and produce a sparse set of measurements which are subject to rolling shutter effects under motion.

Recently, the development of powerful camera and computing hardware as well as tremendous research progress in the field of computer vision have fueled novel research in the area of real-time 3D reconstruction with RGB-D, monocular, and stereo cameras. Whereas traditional visual SLAM techniques are based on keypoints, and camera tracking and dense stereo reconstruction are done individually, in this paper, we demonstrate that accurate and consistent results can be obtained with a direct real-time capable SLAM technique for stereo cameras. To this end, we simultaneously track the 6D pose of the stereo camera and reconstruct semi-dense depth maps using spatial and temporal stereo cues. A loop closure thread running in the background ensures globally consistent reconstructions of the environment.

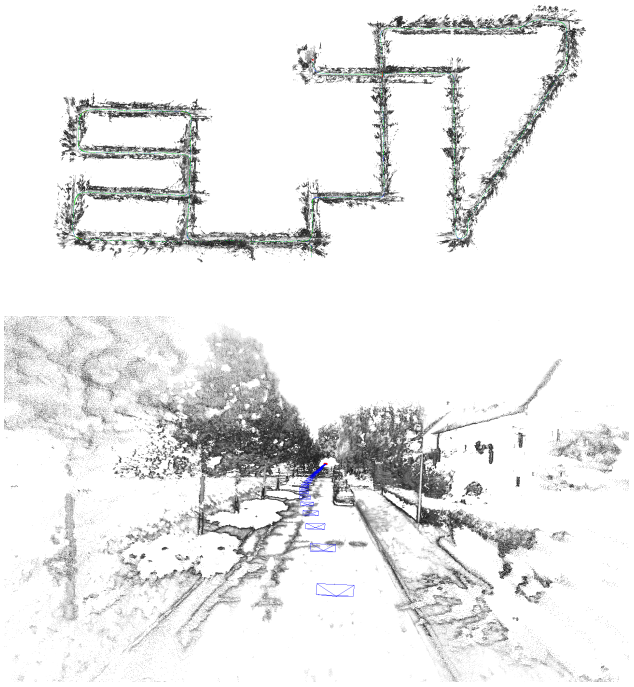

Figure 1. Reconstruction of a street-scene obtained with Large-scale Direct SLAM using a stereo camera. The method uses static and temporal stereo to compute semi-dense maps, which are then used to track camera motion. The figure shows a sample 3D reconstruction on one of the sequences from the popular Kitti dataset.

We evaluate the reconstruction quality of our method on the popular Kitti dataset [8]. It contains images captured from a synchronized stereo pair which is mounted on the roof of a car driving through a city. The dataset also provides point clouds from a Velodyne scanner, which we use as ground-truth to evaluate the accuracy of our reconstruction method. We also compare depth estimation accuracy of our approach with other state-of-the-art stereo reconstruction methods.

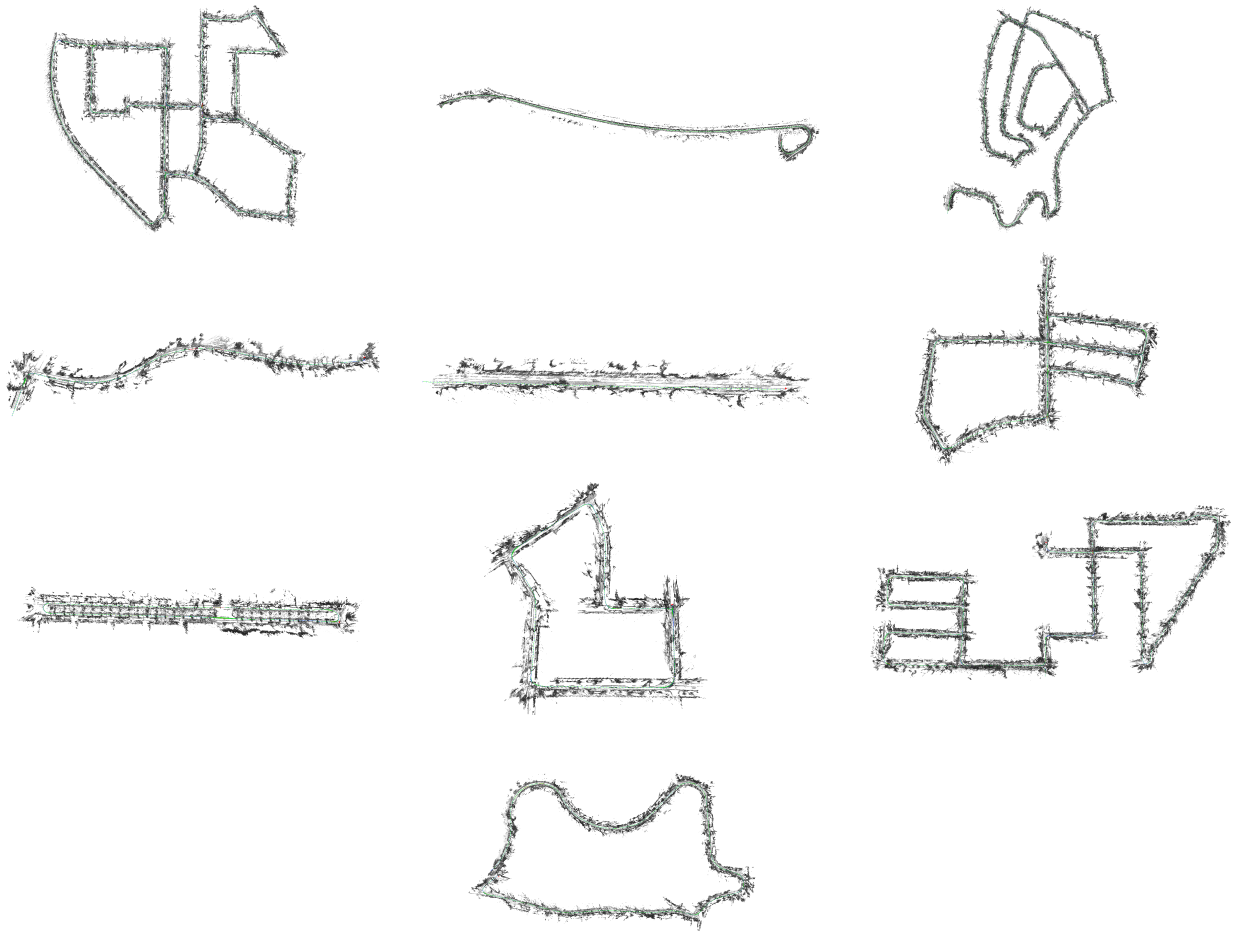

Figure 2. Top-down view on the reconstructions on the Kitti sequences 00-09 obtained with our method.

1.1. Related Work

Traditionally, laser scanners have been used for obtaining 3D reconstructions of environments. Several research communities tackle this problem using laser scanners, including geomatics (e.g. [22]), computer graphics (e.g. [21]), and robotics (e.g. [17]). The popularity of laser scanners stems from the fact that they directly measure 3D point clouds with high accuracy. Still, their measurement frequency is limited since typically the laser beam is redirected in all directions using mechanically moving parts. In effect, either scanning time is slow, scanners become huge with many individual beams, or sparse scans are produced.

For the alignment of the 3D point clouds produced by static laser scanners, typically variants of the Iterative Closest Point (ICP) algorithm [2] are used. An extensive overview and comparisons of different variants can be found in [18], but most of them are not capable of real-time reconstruction of large-scale environments, such as cities.

If the laser scanner itself moves during acquisition, e.g. on a self-driving car, spatial referencing of the individual 3D measurements becomes more difficult. An accurate odometry and mapping system that utilizes laser scans during motion has been presented in [23]. Real-time capability is achieved by running 6D motion tracking of the laser scanner at high frequency on a small time-slice of points and registering all available data at lower frequency. Recently, this algorithm has been extended to use stereo images to improve the odometry accuracy [24].

Camera based 3D reconstruction methods can be divided into sparse and dense methods. Sparse methods typically use bundle adjustment techniques to simultaneously track a motion of the camera and estimate positions of the observed keypoints. Parallel Tracking and Mapping (PTAM [13]) is a prominent example which requires just a single monocular camera to perform 3D reconstruction, however in the monocular case, the scale of the map is not observable. This problem is tackled in FrameSLAM [14] and RSLAM [16] which also apply bundle adjustment. Through the use of stereo cameras with known baseline, these methods are capable to estimate metric scale. Some stereo SLAM meth-

ods estimate camera motion based on sparse interest points [12], but use a dense stereo-processing algorithms in a second stage to obtain dense 3D maps [15, 10].

Over the last decades, a large amount of the algorithms for stereo reconstructions have been developed. Methods using fixed-baseline rectified images became very popular because of their low computational costs and accurate results. These algorithms can be divided into local and global. Local methods such as block-matching (e.g. [1]) use only the local neighborhood of the pixel when searching for the corresponding pixel in a different image. This makes them fast, but usually resulting reconstructions are not spatially consistent. Global methods (e.g. [11, 19]) on the contrary use regularization to improve spatial consistency, which results in better reconstructions but also requires much higher computational costs.

2. Reference Methods

In the following, we discuss some of the most prominent real-time capable stereo reconstruction algorithms which we will use as reference for comparative evaluation with our method.

2.1. Block Matching

The Block Matching (BM) algorithm is one of the most simple local algorithms for stereo reconstruction. First implementations date back to the eighties [1] and comprehensive descriptions can be found in [3] and [20]. It estimates a disparity by searching for the most similar rectangular patch in the second stereo image using some patch comparison metric. Ideally a metric should be tolerant to intensity differences, such as change of camera gain or bias, and also tolerant to deformations when dealing with slanted surfaces. One typical example of such a metric is normalized cross-correlation.

The outcome of the algorithm is sensitive to the selected local window size. If the window size is small, low-textured areas will not be reconstructed since no distinct features will fit into a window. On the other hand a large window size will result in over-smoothed and low-detail reconstructions in highly textured areas.

2.2. Semi-Global Block Matching

Semi-Global Block Matching (SGBM) proposed in [11] is an algorithm that estimates optimal disparities along 1D lines using a polynomial time algorithm. This allows the algorithm to produce more accurate results than local methods, but since disparities are optimized only along one direction the results are typically worse than global methods that regularize in the 2D image domain.

2.3. Efficient Large-Scale Stereo Matching

The Efficient Large-Scale Stereo (elas) Matching method proposed in [9] uses a set of support points that can be reliably matched between the images for speeding up disparity computations on high-resolution images. Based on the matched support points, maximum a-posteriory estimation yields a dense disparity map.

3. Method

Our approach to real-time 3D reconstruction is based on a stereo-extension to the LSD-SLAM [6] method. In LSD-SLAM, key-frames along the camera trajectory are extracted. The motion of the camera is tracked towards a reference key-frame using direct image alignment. Concurrently, semi-dense depth at high-gradient pixels in the reference key-frame is estimated from stereo towards the current frame. For SLAM on the global scale, the relative poses between the key-frames is estimated using scale-aware direct image alignment, and the scale and view poses of the reference key-frames are estimated using graph optimization.

In our stereo-extension to LSD-SLAM, scale is directly observable through the fixed baseline of the stereo camera. Also for depth estimation, now also static stereo through the fixed baseline complements the temporal stereo of the tracked frames towards the reference key-frame. In temporal stereo, disparity can be estimated along epipolar lines whose direction depends on the motion direction of the camera. This extends the fixed direction for disparity estimation of the static stereo camera. On the other hand, the baseline of the stereo camera can be precisely calibrated in advance and allows for determing reconstruction scale unambiguously.

In this paper we use a stereo version of the algorithm, since the monocular version [7] does not perform well on the Kitti sequences and fails frequently. This is presumably due to the large inter-frame motion along the line-of-sight of the camera.

In the following we will briefly describe the main steps of our stereo LSD-SLAM method. Special focus is given to semi-dense depth estimation and 3D reconstruction.

3.1. Notation

We briefly introduce basic notation that we will use throughout the paper. Matrices will be denoted by bold capital letters ($\mathbf{R}$), while bold lower case letters are used for vectors ($\boldsymbol{\xi}$). The operator $[\cdot]_n$ selects the n-th row of a matrix. The symbol d denotes the inverse $d = z^{-1}$ of depth z.

The global map is maintained as a set of keyframes $\mathcal{K}_i = \{I_i^l, I_i^r, D_i, V_i\}$. They include a left and right stereo image $I_i^{l/r} : \Omega \to \mathbb{R}$, an inverse depth map $D_i : \Omega_{D_i} \to \mathbb{R}^+$ and a variance map $V_i : \Omega_{D_i} \to \mathbb{R}^+$. Depth and variance

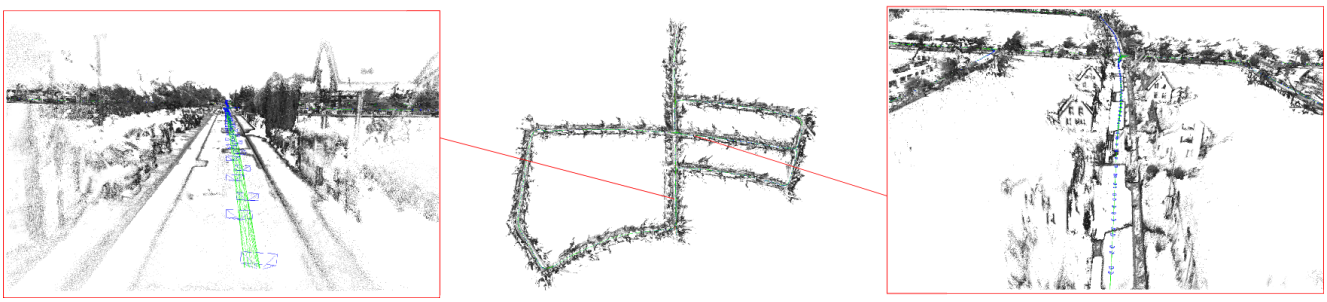

Figure 3. Top down view on the reconstruction obtained with out method on Kitti sequence 05 with two close views.

are only maintained for one of the images in the stereo pair. We assume the image domain $\Omega \subset \mathbb{R}^2$ to be given in stereo-rectified image coordinates, i.e., the intrinsic and extrinsic camera parameters are known a-priori. The domain $\Omega_{D_i} \subset \Omega$ is the semi-dense restriction to the pixels which are selected for depth estimation.

We denote pixel coordinates by $\boldsymbol{u} = (u_x\, u_y\, 1)^T$. A 3D position $\boldsymbol{p} = (p_x\, p_y\, p_z\, 1)^T$ is projected into the image plane through the mapping

$$\boldsymbol{u} = \pi(\boldsymbol{p}) := \mathbf{K}\left((p_x/p_z)\,(p_y/p_z)\,1\right)^T, \tag{1}$$

where K is the camera matrix. The mapping $\boldsymbol{p} = \pi^{-1}(\boldsymbol{u}, d) := \left(\left(d^{-1}\mathbf{K}^{-1}\boldsymbol{u}\right)^T 1\right)^T$ inverts the projection with the inverse depth d. We parametrize poses $\boldsymbol{\xi} \in \mathfrak{se}(3)$ as elements of the Lie algebra of SE(3) with exponential map $\mathbf{T}_{\boldsymbol{\xi}} = \exp(\widehat{\boldsymbol{\xi}}) \in \mathrm{SE}(3)$. We will use $\mathbf{R}_{\boldsymbol{\xi}}$ and $\mathbf{t}_{\boldsymbol{\xi}}$ for the respective rotation matrix and translation vector.

3.2. Tracking through Direct Image Alignment

The relative pose between images I_{curr} and I_{ref} is estimated by minimizing pixel-wise residuals

$$E(\boldsymbol{\xi}) = \sum_{u \in \Omega} \rho\left(r_{\boldsymbol{u}}(\boldsymbol{\xi})^T \Sigma_{r,\boldsymbol{u}}^{-1} r_{\boldsymbol{u}}(\boldsymbol{\xi})\right) \tag{2}$$

which we solve as a non-linear least-squares problem. We implement a robust norm ρ on the square residuals such as the Huber-norm using the iteratively re-weighted least squares method.

One common choice of residuals are the photometric residuals

$$r_{\boldsymbol{u}}^I(\boldsymbol{\xi}) := I_{\text{ref}}(\boldsymbol{u}) - I_{\text{curr}}\left(\pi\left(\mathbf{T}_{\boldsymbol{\xi}}\pi^{-1}\left(\boldsymbol{u}, D_{\text{ref}}(\boldsymbol{u})\right)\right)\right). \tag{3}$$

The uncertainty $\sigma_{r,\boldsymbol{u}}^I$ in this residual can be determined from a constant intensity measurement variance σ_I^2 and the propagated inverse depth uncertainty [5]. Pure photometric alignment uses $r_{\boldsymbol{u}}(\boldsymbol{\xi}) = r_{\boldsymbol{u}}^I(\boldsymbol{\xi})$.

If depth maps are available in both images, we can also measure the geometric residuals

$$r_{\boldsymbol{u}}^G(\boldsymbol{\xi}) := [p']_z - D_{\text{curr}}(\pi(p')) \tag{4}$$

with $p' := \mathbf{T}_{\boldsymbol{\xi}}\pi^{-1}(\boldsymbol{u}, D_{\text{ref}}(\boldsymbol{u}))$. The variance of the geometric residual is determined from the variance in the inverse depth estimates in both frames [5]. Note that we can easily use both types of residuals simultaneously by combining them in a stacked residual. We minimize the direct image alignment objectives using the iteratively reweighted Levenberg-Marquardt algorithm.

We furthermore compensate for lighting changes with an affine lighting model with scale and bias parameters. These parameters are optimized with the pose $\boldsymbol{\xi}$ in an alternating way.

3.3. Depth Estimation

Scene geometry is estimated for pixels of the key frame with high image gradient, since they provide stable disparity estimates. We initialize the depth map with the propagated depth from the previous keyframe. The depth map is subsequently updated with new observations in a pixel-wise depth-filtering framework. We also regularize the depth maps spatially and remove outliers [5].

We estimate disparity between the current frame and the reference keyframe using the pose estimate obtained through tracking. These estimates are fused in the keyframe. Only pixels are updated with temporal stereo, whose expected inverse depth error is sufficiently small. This also constrains depth estimates to pixels with high image gradient along the epipolar line, producing a semi-dense depth map.

For direct visual odometry with stereo cameras we estimate depth both from static stereo (i.e., using images from different physical cameras, but taken at the same point in time) as well as from temporal stereo (i.e., using images from the same physical camera, taken at different points in time).

We determine the static stereo disparity at a pixel by a correspondence search along its epipolar line in the other stereo image. In our case of stereo-rectified images, this search can be performed very efficiently along horizontal lines. As correspondence measure we use the SSD photometric error over five pixels along the scanline.

Static stereo is integrated in two ways. If a new stereo

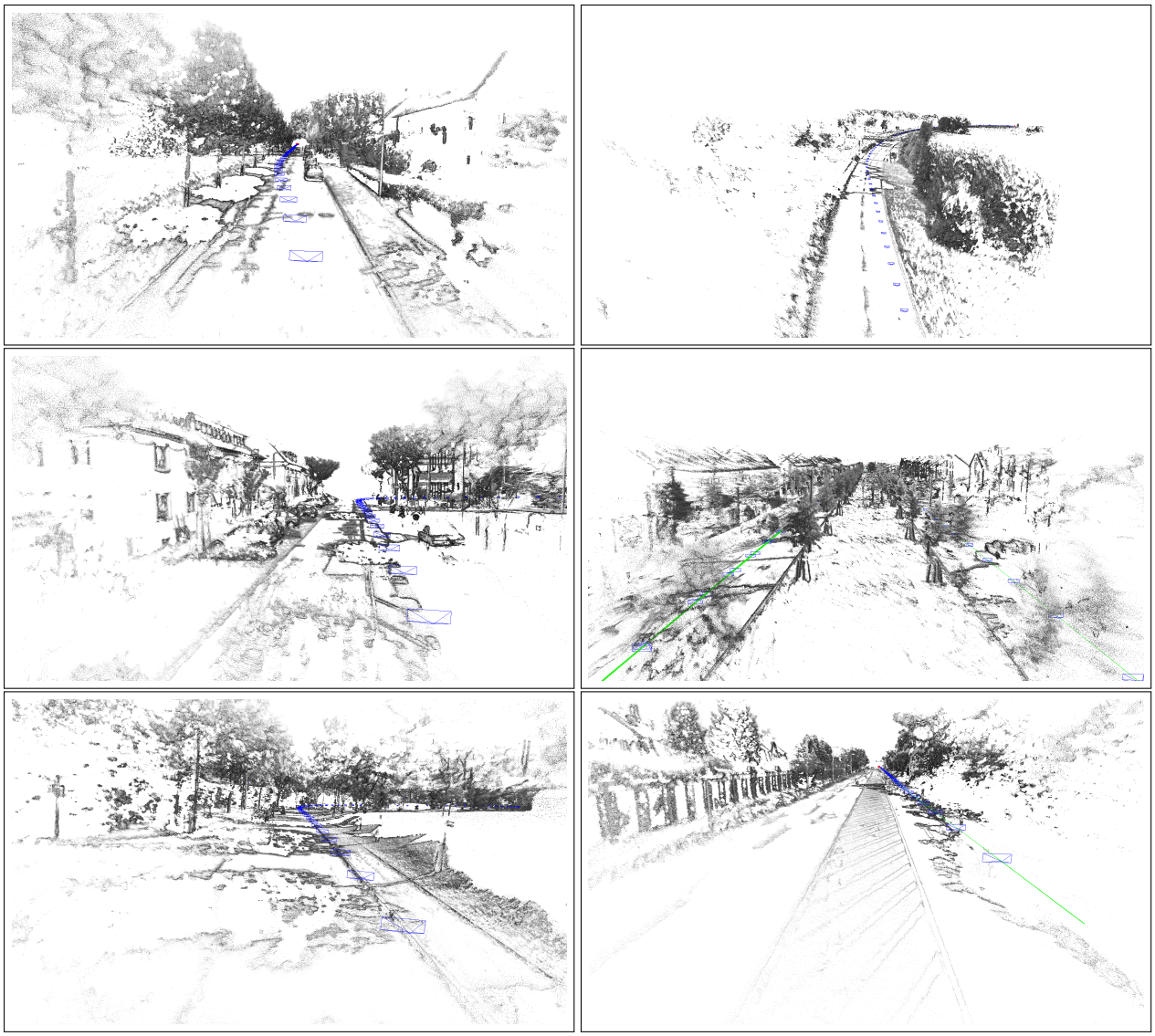

Figure 4. Close-by views on the point clouds generated by our method from the Kitti sequences 08, 01, 08, 06, 02 and 04.

keyframe is created, the static stereo in this keyframe stereo pair is used to initialize the depth map. During tracking, static stereo in the current frame is propagated to the reference frame and fused with its depth map.

There are several benefits of complementing static with temporal stereo in a tracking and mapping framework. Static stereo makes reconstruction scale observable. It is also independent of camera movement, but is constrained to a constant baseline, which limits static stereo to an effective operating range. Temporal stereo requires non-degenerate camera movement, but is not bound to a specific range as demonstrated in [5]. The method can reconstruct very small and very large environments at the same time. Finally, through the combination of static with temporal stereo, multiple baseline directions are available: while static stereo typically has a horizontal baseline – which does not allow for estimating depth along horizontal edges, temporal stereo allows for completing the depth map by providing other motion directions.

3.4. Key-Frame-Based SLAM

With the tracking and mapping approach presented in the previous section, it is possible to extract keyframes along the camera trajectory and to estimate their poses consistently in a global reference frame. The method operates by tracking the camera motion towards reference keyframes. Once the motion of the camera is sufficiently large, and the image overlap is too small, the current camera image is selected as a new reference keyframe. The old reference key frame is added to a keyframe pose-graph. We estimate rel-

ative pose constraints between the keyframes through photometric and geometric direct image alignment. We test as set of possible loop-closure candidates that are either found as nearby view poses or through appearance-based image retrieval.

In monocular LSD-SLAM, we need to use Sim(3) pose representation due to the scale ambiguity. Conversely, for stereo LSD-SLAM, we can directly use SE(3) parametrization.

For each candidate constaint $\mathcal{K}_{j_k}$ between keyframe k and j we independently compute $\boldsymbol{\xi}_{j_k i}$ and $\boldsymbol{\xi}_{i j_k}$ through direct image alignment using photometric and geometric residuals. Only if the two estimates are statistically similar, i.e., if

$$e(\boldsymbol{\xi}_{j_k i}, \boldsymbol{\xi}_{i j_k}) := (\boldsymbol{\xi}_{j_k i} \circ \boldsymbol{\xi}_{i j_k})^T \boldsymbol{\Sigma}^{-1} (\boldsymbol{\xi}_{j_k i} \circ \boldsymbol{\xi}_{i j_k}) \quad (5)$$

$$\text{with} \quad \boldsymbol{\Sigma} := \boldsymbol{\Sigma}_{j_k i} + \text{Adj}_{j_k i} \boldsymbol{\Sigma}_{i j_k} \text{Adj}_{j_k i}^T \quad (6)$$

is sufficiently small, they are added as constraints to the pose-graph. To speed up the removal of incorrect loop-closure candidates, we apply this consistency check after each pyramid level in the coarse-to-fine scheme. Only if it passes, direct image alignment is continued on the next higher resolution. This allows for discarding most incorrect candidates with only very little wasted computational resources.

4. Evaluation

We evaluate our method on the popular and publicly available Kitti dataset [8]. It provides rectified stereo images together with Velodyne laser scans captured from a car driving in various street-scene environments. Specifically, we used the odometry dataset sequences 00-09 which provide camera pose ground-truth that can be used to align several Velodyne scans and to generate denser ground-truth point clouds in this way.

We present qualitative results of the point clouds generated from the semi-dense depth maps over the whole trajectory. We also evaluate the accuracy of the reconstructed point clouds for different processed image resolutions by comparing them with the Velodyne ground-truth. Finally, we compare our reconstructions with other state-of-the-art stereo reconstruction methods in terms of accuracy and runtime. We use a PC with Intel i7-2600K CPU running at 3.4 GHz for run-time evaluation.

4.1. Qualitative Results

In Fig. 2, we show reconstructions over the full length trajectories with our method which estimates the camera motion and semi-dense depth maps in real-time. Loop closures and pose graph optimization running in a separate thread allows for keeping the reconstructions globally consistent.

Figs. 1, 3 and 4 show a closer view on the reconstructed pointclouds. As can be seen, most parts of the high gradient areas of the image are reconstructed. Using both fixed-baseline stereo and temporal stereo allows us to estimate a proper metric scale of the scene.

The main advantage of the proposed method is that the generated semi-dense maps are directly used for camera tracking and loop closures, while other methods, such as [10], rely on external keypoint-based odometry. Our approach results in consistent and visually pleasing 3D reconstructions.

4.2. Reconstruction Accuracy Depending on Image Resolution

In order to evaluate 3D reconstruction accuracy, we generate a ground-truth pointcloud by aligning several Velodyne scans. For every point in the reference point cloud we estimate a local surface normal by fitting a plane to the points in a 20 cm neighborhood. For each keyframe, we reproject the estimated semi-dense point cloud and remove the points that are higher than the maximum height of Velodyne measurements. For all other points, we determine the point-to-plane distance to the closest point in the reference point cloud.

Figure 5 shows the median, 5th and 95th percentiles of the average point-to-plane distance across all keyframes in different sequences. The results show that there is no direct dependency between the resolution of the image and accuracy of the resulting pointcloud. On a smaller resolution images depth-estimation is less sensitive to small repetitive structures, which results in smaller number of outliers. Also with all cases we use a sub-pixel depth estimation to determine depth, which can also lower the difference between different resolutions.

4.3. Comparison to Other Stereo Reconstruction Methods

Fig. 6 gives a comparison of reconstructions generated with our method (LSD) to the point clouds generated by Efficient Large-Scale Stereo Matching (elas, [9]), Semi-Global Block Matching (SGBM, [11]) and regular Block Matching (BM). For accuracy evaluation, we use the same method as in Sec. 4.2. Fig. 7 shows an example of a ground-truth point cloud overlayed with reconstructions from the stereo methods.

In most of the sequences our method has less or equal average point-to-plane distance to ground-truth point cloud than SGBM and BM, but elas often performs slightly better than our method. However, elas is a pure stereo reconstruction algorithm that only uses static stereo cues from the fixed-baseline camera arrangement. Elas also relies on an extra, e.g. keypoint-based, visual tracking method, while our simultaneous tracking and mapping approach uses the

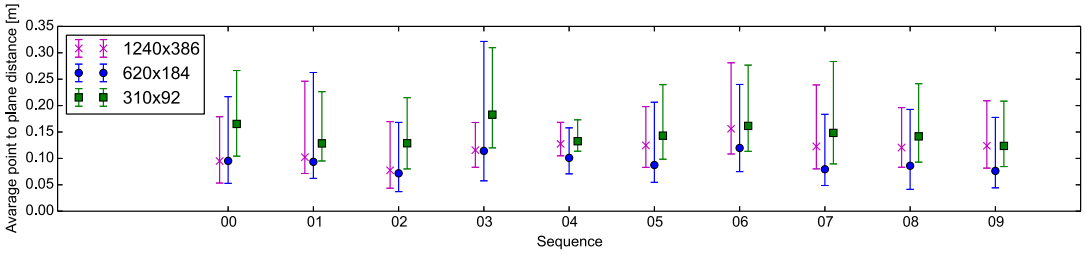

Figure 5. Point-to-plane distance between ground-truth point clouds and reconstructions with our method generated at different image resolutions. We give median, 5th, and 95th percentile on the various Kitti sequences.

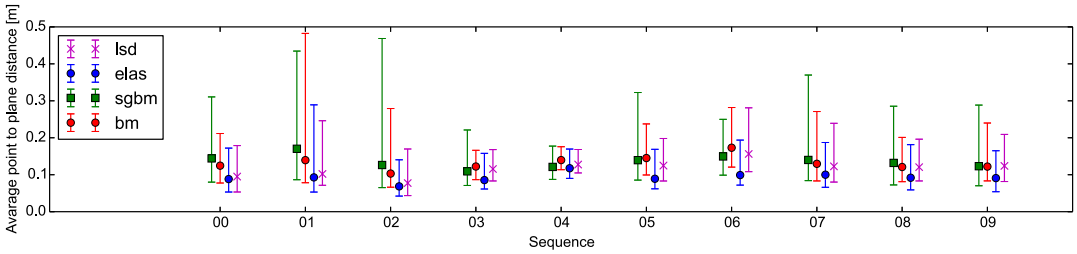

Figure 6. Point-to-plane distance between ground-truth point clouds and various stereo-reconstruction methods (including ours). We give median, 5th, and 95th percentile on the various Kitti sequences.

	310 x 92	620 x 184	1240 x 368
Mapping time LSD	5.8ms	23ms	103ms

Table 1. Depth estimation run-time of LSD for different image resolutions.

	LSD	SGBM	BM	elas
Mapping time	103ms	145ms	24ms	142ms

Table 2. Depth estimation run-time for different stereo methods.

3D reconstruction directly for tracking and vice versa.

4.4. Run-Time

Table 1 shows the run-time required for a stereo reconstruction by our method for a single frame depending on the image resolution. It can be seen that run-time scales almost linearly with the number of pixels in the image. In Table 2, we compare the run-times of the competing methods. The block matching is the fastest method, but it is also slightly less accurate than the other methods. Our method shows a second result, and both SGBM and elas exhibit similar run-times which are approximately 1.4 times higher than for our method.

5. Conclusions

In this paper, we compare real-time 3D reconstruction of street scenes using a semi-dense large-scale direct SLAM method for stereo cameras with traditional stereo and laser-based approaches. We demonstrate qualitative reconstruction results on the Kitti dataset and compare the accuracy of our method to the state-of-the-art stereo reconstruction methods. These approaches often rely on a separate visual odometry or SLAM method, e.g. based on sparse interest points, in order to recover the camera trajectory. Dense depth is then obtained in the individual stereo frames. To our knowledge, our approach is the first method that can recover metrically accurate and globally consistent 3D reconstructions of large-scale environments from a stereo-camera system in real-time on a single CPU.

Acknowledgements

This work has been partially supported by grant CR 250/9-2 (Mapping on Demand) of German Research Foundation (DFG) and grant 16SV6394 (AuRoRoll) of BMBF.

References

[1] S. T. Barnard and M. A. Fischler. Computational stereo. *ACM Comput. Surv.*, 14(4):553–572, Dec. 1982. 3

[2] P. Besl and N. D. McKay. A method for registration of 3-d shapes. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 14(2):239–256, 1992. 2

[3] M. Brown, D. Burschka, and G. Hager. Advances in computational stereo. *Pattern Analysis and Machine Intelligence, IEEE Transactions on*, 25(8):993–1008, Aug 2003. 3

[4] M. Buehler, K. Iagnemma, and S. Singh, editors. *The DARPA Urban Challenge: Autonomous Vehicles in City*

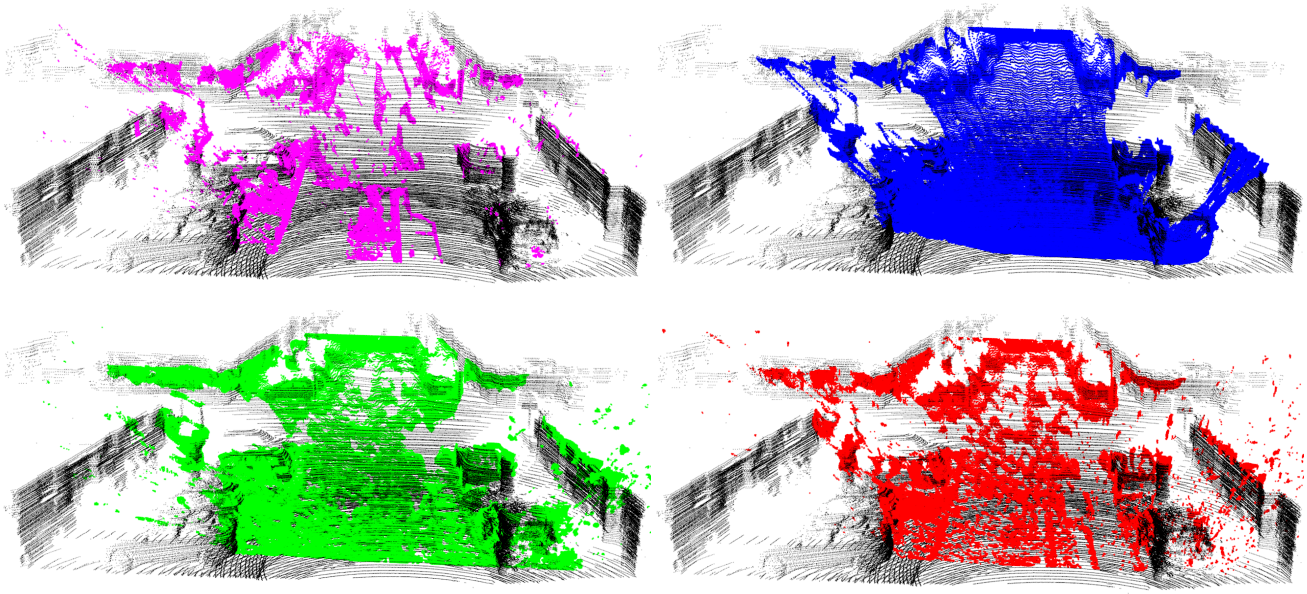

Figure 7. Single keyframe 3D reconstructions using LSD (purple), elas (blue), SGBM (green) and BM (red) are compared to the point cloud produced by a Velodyne laser scanner. Laser point clouds from several frames are aligned and combined into a denser point cloud. The point clouds generated by the stereo reconstruction methods are cut at the maximum measurement height of the Velodyne scanner.

Traffic, George Air Force Base, Victorville, California, USA, volume 56 of *Springer STAR*, 2009. 1

[5] J. Engel, T. Schöps, and D. Cremers. LSD-SLAM: Large-scale direct monocular SLAM. In *European Conference on Computer Vision (ECCV)*, September 2014. 4, 5

[6] J. Engel, J. Stückler, and D. Cremers. Large-scale direct SLAM with stereo cameras. In *Int. Conf. on Intelligent Robot Systems (IROS)*, 2015. 3

[7] J. Engel, J. Sturm, and D. Cremers. Semi-dense visual odometry for a monocular camera. In *Int. Conf. on Computer Vision (ICCV)*, 2013. 3

[8] A. Geiger. Are we ready for autonomous driving? the KITTI vision benchmark suite. In *Int. Conf. on Computer Vision and Pattern Recognition (CVPR)*, 2012. 1, 6

[9] A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In *Asian Conference on Computer Vision (ACCV)*, 2010. 3, 6

[10] A. Geiger, J. Ziegler, and C. Stiller. Stereoscan: Dense 3D reconstruction in real-time. In *Proceedings of the IEEE Intelligent Vehicles Symposium*, pages 963–968, 2011. 3, 6

[11] H. Hirschmuller. Stereo processing by semiglobal matching and mutual information. *IEEE Trans. on PAMI*, 30(2):328–341, 2008. 3, 6

[12] B. Kitt, A. Geiger, and H. Lategahn. Visual odometry based on stereo image sequences with RANSAC-based outlier rejection scheme. In *Intelligent Vehicles Symp. (IV)*, 2010. 3

[13] G. Klein and D. Murray. Parallel tracking and mapping for small AR workspaces. In *Int. Symp. on Mixed and Augmented Reality (ISMAR)*, 2007. 2

[14] K. Konolige and M. Agrawal. Frameslam: From bundle adjustment to real-time visual mapping. *Transaction on Robotics*, 24(5):1066–1077, Oct. 2008. 2

[15] M. Pollefeys et al. Detailed real-time urban 3D reconstruction from video. *Int. J. Comput. Vision (IJCV)*, 78(2-3):143–167, 2008. 3

[16] C. Mei, G. Sibley, M. Cummins, P. Newman, and I. Reid. RSLAM: A system for large-scale mapping in constant-time using stereo. *Int. J. Comput. Vision (IJCV)*, 2010. 2

[17] A. Nüchter. *3D robotic mapping : the simultaneous localization and mapping problem with six degrees of freedom*. STAR. Springer, Berlin, Heidelberg, 2009. 2

[18] F. Pomerleau, F. Colas, R. Siegwart, and S. Magnenat. Comparing ICP variants on real-world data sets. *Auton. Robots*, 34(3):133–148, Apr. 2013. 2

[19] R. Ranftl, S. Gehrig, T. Pock, and H. Bischof. Pushing the limits of stereo using variational stereo estimation. In *Intelligent Vehicles Symposium*, pages 401–407, 2012. 3

[20] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *International Journal of Computer Vision*, 47:7–42, 2002. 3

[21] G. K. L. Tam, Z.-Q. Cheng, Y.-K. Lai, F. C. Langbein, Y. Liu, D. Marshall, R. R. Martin, X.-F. Sun, and P. L. Rosin. Registration of 3D point clouds and meshes: A survey from rigid to nonrigid. *IEEE Transactions on Visualization and Computer Graphics*, 19(7):1199–1217, 2013. 2

[22] P. W. Theiler and K. Schindler. Automatic registration of terrestrial laser scanner point clouds using natural planar surfaces. *ISPRS Annals of Photogrammetry, Remote Sensing and Spatial Information Sciences*, I-3:173–178, 2012. 2

[23] J. Zhang and S. Singh. Loam: Lidar odometry and mapping in real-time. In *Robotics: Science and Systems*, 2014. 2

[24] J. Zhang and S. Singh. Visual-lidar odometry and mapping: Low-drift, robust, and fast. *IEEE Intl. Conf. on Robotics and Automation (ICRA)*, May 2015. 2