A Multi-View Stereo Benchmark with High-Resolution Images and Multi-Camera Videos

Thomas Schöps[1] Johannes L. Schönberger[1] Silvano Galliani[2]
Torsten Sattler[1] Konrad Schindler[2] Marc Pollefeys[1,4] Andreas Geiger[1,3]
[1]Department of Computer Science, ETH Zürich
[2]Institute of Geodesy and Photogrammetry, ETH Zürich
[3]Autonomous Vision Group, MPI for Intelligent Systems, Tübingen [4]Microsoft, Redmond

Abstract

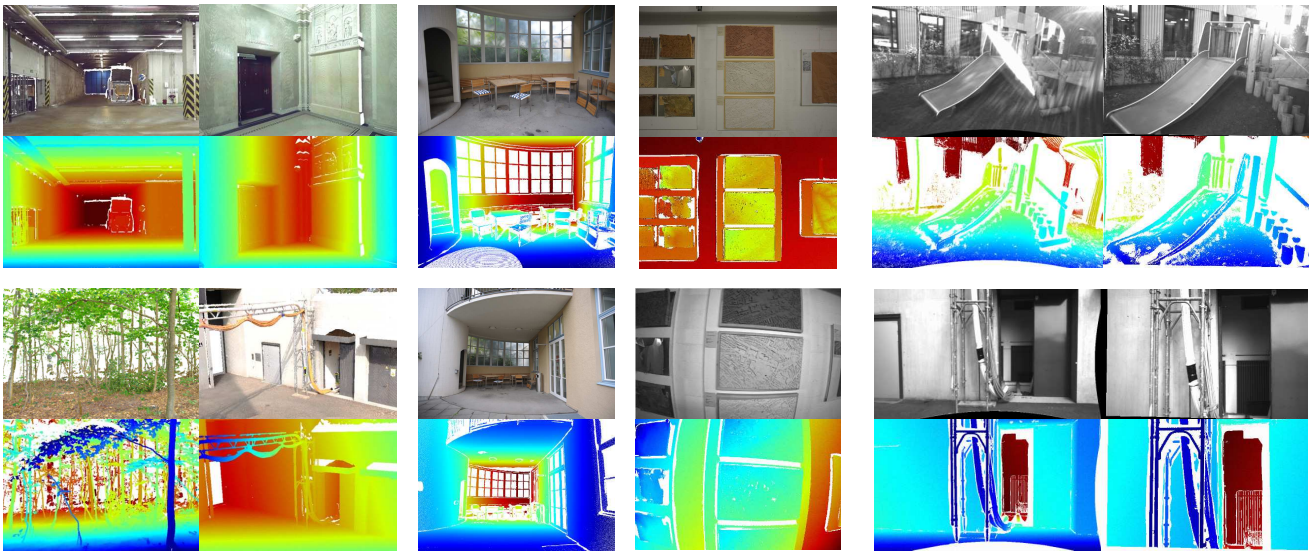
Motivated by the limitations of existing multi-view stereo benchmarks, we present a novel dataset for this task. Towards this goal, we recorded a variety of indoor and outdoor scenes using a high-precision laser scanner and captured both high-resolution DSLR imagery as well as synchronized low-resolution stereo videos with varying fields-of-view. To align the images with the laser scans, we propose a robust technique which minimizes photometric errors conditioned on the geometry. In contrast to previous datasets, our benchmark provides novel challenges and covers a diverse set of viewpoints and scene types, ranging from natural scenes to man-made indoor and outdoor environments. Furthermore, we provide data at significantly higher temporal and spatial resolution. Our benchmark is the first to cover the important use case of hand-held mobile devices while also providing high-resolution DSLR camera images. We make our datasets and an online evaluation server available at http://www.eth3d.net.

1. Introduction

The problem of reconstructing 3D geometry from two or more views has received tremendous attention in computer vision for several decades. Applications range from 3D reconstruction of objects [4] and larger scenes [3, 5, 35] over dense sensing for autonomous vehicles [6–8, 11, 30] or obstacle detection [10] to 3D reconstruction from mobile devices [14, 20, 28, 36, 41]. Despite its long history, many problems in 3D reconstruction remain unsolved to date. To identify these problems and analyze the strengths and weaknesses of the state-of-the-art, access to a large-scale dataset with 3D ground truth is indispensable.

Indeed, the advent of excellent datasets and benchmarks, such as [6, 12, 17, 29, 32–34, 38, 40], has greatly advanced the state of the art of stereo and multi-view stereo (MVS) techniques. However, constructing good benchmark datasets is a tedious and challenging task. It requires the acquisition of images and a 3D scene model, *e.g.*, through a laser scanner or structured light sensor, as well as careful registration between the different modalities. Often, manual work is required [6] to mask occluded regions, sensor inaccuracies or image areas with invalid depth estimates, *e.g.*, due to moving objects. Therefore, existing benchmarks are limited in their variability and are often also domain specific.

This paper presents a novel benchmark for two- and multi-view stereo algorithms designed to complement existing benchmarks across several dimensions (*c.f.* Fig. 1): (i) Compared to previous MVS benchmarks, our dataset offers images acquired at a very high resolution. Using a professional DSLR camera, we capture images at 24 Megapixel resolution compared to 6 Megapixels in Strecha *et al.* [40], 0.5 Megapixels in KITTI [6], and 0.3 Megapixels in Middlebury [38]. This enables the evaluation of algorithms designed for detailed 3D reconstruction. At the same time, it encourages the development of memory and computationally efficient methods which can handle very large datasets. (ii) By now, mobile devices have become powerful enough for real-time stereo [20, 28, 30, 36, 41], creating the need for benchmark datasets that model the acquisition process typical for such hand-held devices. In addition to the DSLR images, we also capture a set of image sequences with four synchronized cameras forming two stereo pairs that move freely through the scene. These videos enable algorithms to exploit the redundancy provided by high frame rates to improve the reconstruction quality. Again, this scenario rewards efficient algorithms which can handle large amounts of data. To study the effect of field-of-view (FOV) and distortion, we recorded stereo imagery using different lenses. (iii) In contrast to the Middlebury benchmarks [33, 38], our scenes are not carefully staged in a controlled laboratory environment. Instead, they provide the full range of challenges of real-word photogrammetric measurements. Rather than moving along a constrained trajectory, *e.g.*, in a



(a) Scene type (b) View point (c) Camera type (d) Field of view

Figure 1. Examples demonstrating the variety of our dataset in terms of appearance and depth. (a) Colored 3D point cloud renderings of different natural and man-made scenes. (b) DSLR images taken from different view points. (c) DSLR image (top) and image from our multi-camera rig (bottom) of the same scene. (d) Camera rig images with different fields-of-view.

circle around an object, our cameras undergo unconstrained 6-DoF motion. As a result, MVS algorithms need to be able to account for stronger variations in viewpoint. In contrast to Strecha's dataset [40], our benchmark covers a wider spectrum of scenes, ranging from office scenes over man-made outdoor environments to natural scenes depicting mostly vegetation. The latter type is especially interesting as there exist fewer priors which are applicable to this scenario. In addition, our scenes comprise fine details (*e.g.*, trees, wires) which are challenging for existing techniques.

The contributions of this paper include (i) a benchmark, which is made publicly available together with a website for evaluating novel algorithms on a hold out test set, (ii) a highly accurate alignment strategy that we use to register images and video sequences against 3D laser scan point clouds, and (iii) an analysis of existing state-of-the-art algorithms on this benchmark. Our benchmark provides novel challenges and we believe that it will become an invaluable resource for future research in dense 3D reconstruction with a focus on big data and mobile devices.

2. Related Work

In this section, we review existing two- and multi-view stereo datasets. An overview which compares key aspects of our dataset to existing benchmarks is provided in Tab. 1.

Two-View Stereo Datasets. One of the first datasets for two-view stereo evaluation was the Tsukuba image pair [27] for which 16 levels of disparity have been manually annotated. Unfortunately, manual annotation does not scale to large realistic datasets due to their complexity [21].

Towards more realism, Scharstein *et al*. [33] proposed the Middlebury stereo evaluation, comprising 38 indoor scenes at VGA resolution with ground truth correspondences obtained via a structured light scanner. A new version of the Middlebury dataset [32] has recently been released, featuring ground truth disparities for 33 novel scenes at a resolution of 6 Megapixels. Unfortunately, the amount of human labor involved in staging the scenes and recording the ground truth is considerable. Thus, these datasets are relatively small in size. Besides, their variability is limited as the setup requires controlled structured lighting conditions. In contrast, we are interested in general scenes and introduce a dataset that features both indoor *and* outdoor environments.

Geiger *et al*. [6, 24] recorded the KITTI datasets using a mobile platform with a laser scanner mounted on a car. This automated the recording process. However, the benchmark images are of low resolution (0.5 Megapixels) and the ground truth annotations are sparse ($< 50\%$ of all image pixels). Besides, a fixed sensor setup on a car limits the diversity of the recorded scenes to road-like scenarios.

Rendered images, as used in the MPI Sintel stereo benchmark [1], provide an alternative to real recordings and have been used for learning complex models [23]. Yet, creating realistic 3D models is difficult and the degree of realism required is still not well understood [43].

Multi-View Stereo Datasets. The Middlebury dataset of Seitz *et al*. [38] was the first common benchmark for evaluating multi-view stereo on equal grounds. They captured hundreds of images per object using a robot that uniformly sampled the hemisphere enclosing the scene. Reference

Benchmark	Setting	Resolution	Online Eval.	6DoF Motion	MVS	Stereo	Video	Varying FOV
Middlebury MVS [38]	Laboratory	0.3 Mpx	✓		✓			
Middlebury [32, 33]	Laboratory	6 Mpx	✓			✓		
DTU [17]	Laboratory	2 Mpx			✓			
MPI Sintel [1]	Synthetic	0.4 Mpx	✓	✓		✓	✓	
KITTI [6, 24]	Street scenes	0.5 Mpx	✓		✓	✓	✓	
Strecha [40]	Buildings	6 Mpx		✓	✓			
ETH3D (Proposed)	Varied	0.4 / 24 Mpx	✓	✓	✓	✓	✓	✓

Table 1. Comparison of existing state-of-the-art benchmarks with our new dataset. Among other factors, we differentiate between different scene types (*e.g.*, staged scenes captured in a laboratory vs. synthetic scenes), whether the camera undergoes a restricted or a full 6 degrees-of-freedom (DoF) motion, or whether cameras with different fields-of-view (FOV) are used.

data has been created by stitching several line laser scans together. Unfortunately, this benchmark provides a limited image resolution (VGA), and its data, captured in a controlled laboratory environment, does not reflect many of the challenges in real-world scenes. Besides, only two toy scenes with Lambertian surface properties are provided, resulting in overfitting and performance saturation.

As a consequence, Strecha *et al*. [40] proposed a new MVS benchmark comprising 6 outdoor datasets which include $\sim$30 images at 6 Megapixel resolution, as well as ground truth 3D models captured by a laser scanner. While this dataset fostered the development of efficient methods, it provides relatively easy (*i.e*., well-textured) scenes and the benchmark's online service is not available anymore.

To compensate for the lack of diversity in [38, 40] and the well textured, diffuse surfaces of [40], Jensen *et al*. [17] captured a multitude of real-world objects using a robotic arm. Yet, their controlled environment shares several limitations with the original Middlebury benchmark and reduces the variety of scenes and viewpoints.

Simultaneously to our work, Knapitsch *et al*. proposed a new benchmark for challenging indoor and outdoor scenes [19]. Their benchmark provides high-resolution video data and uses ground truth measurements obtained with a laser scanner. While our benchmark focuses on evaluating both binocular stereo and MVS, theirs jointly evaluates Structure-from-Motion (SfM) and MVS. Knapitsch *et al*. captured their video sequences with a high-end camera and carefully selected camera settings to maximize video quality. In contrast, our videos were captured with cameras commonly used for mobile robotics and always use auto-exposure. Thus, both benchmarks complement each other.

3. Data Acquisition and Registration

We follow [6, 24, 25, 38, 40] and capture the ground truth for our dataset using a highly accurate 3D laser scanner. This section describes the data acquisition and our approach to robustly and accurately register images and laser scans.

3.1. Data Acquisition

We recorded the ground truth scene geometry with a Faro Focus X 330 laser scanner. For each scene, depending on the occlusions within it, we recorded one or multiple 360° scans with up to $\sim$28 million points each. In addition to the depth measurements, we recorded the color of each 3D point provided by the laser scanner's integrated RGB camera. Recording a single scan took $\sim$9 minutes.

For the high-resolution image data, we used a professional Nikon D3X DSLR camera on a tripod. We kept the focal length and the aperture fixed, such that the intrinsic parameters can be shared between all images with the same settings. The camera captures photos at a resolution of 6048×4032 Pixels with a 85° FOV.

For the mobile scenario, we additionally recorded videos using the multi-camera setup described in [9]: We use four global-shutter cameras, forming two stereo pairs, which are hardware-synchronized via an FPGA and record images at $\sim$13.6Hz. The cameras of the first stereo pair have a FOV of 54° each, while the other two cameras have a FOV of 83°. All cameras capture images at a resolution of 752×480 pixels. As common and necessary for mobile devices, we set the exposure settings to automatic, allowing the device to adapt to illumination changes.

3.2. Registration

To use the recorded data for our benchmark, we first remove errors from the laser scans and mask problematic areas in the images. Next, we align the scans taken from different positions with each other and register the camera images against the laser scan point clouds. We employ a fully automatic three-stage alignment procedure for this task. The first stage estimates a rough initial alignment between the laser scans and the camera images. We then refine the registration of the laser scans, followed by a refinement of the intrinsic and extrinsic calibration of the cameras. In the following, we describe each of these steps in detail.

Preprocessing. The raw laser scans contain artifacts caused by beams reflected from both foreground and background objects, resulting in the interpolation of foreground and background depths at occlusion boundaries. Furthermore, reflecting objects and glass frequently cause systematic outliers. Therefore, we filter the scans with the statistical outlier removal procedure from [31]. This removes all points from the cloud whose average distance to their

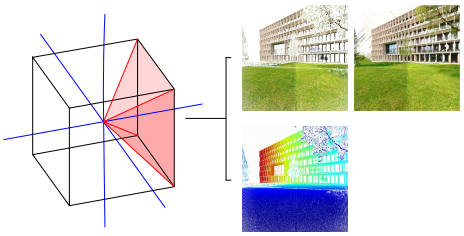


Figure 2. Left: Illustration of a cube map. One of the 6 virtual cameras is highlighted in red, coordinate axes are shown in blue. Right: Sparse color image and depth map (left) and the inpainted image (right) for one virtual camera.

k nearest neighbors is larger than a threshold. The point density of our scans differs depending on the distance from the scanner to the surface. Thus, we compute the threshold used for outlier removal for each point from its local neighborhood rather than using one single global value. In a final step, we manually remove undetected systematic errors. We also inspect each image and annotate regions which should not be used. These regions include moving objects, *e.g.*, moving branches in the wind, objects not represented correctly in the laser scan such as transparent surfaces, or regions for which occlusion reasoning (as described later) fails due to the sparsity of the measured 3D point cloud.

Initial Laser Scan and Image Alignment. We use the COLMAP SfM pipeline [35,37] to obtain an initial estimate of the laser scan poses as well as the extrinsics and intrinsics of the cameras. It is well known [15,39] that rendered views can be robustly matched against real imagery using classical descriptors like SIFT [22]. To register the laser scans with the images, we thus include rendered cube map images for each scan position into the SfM reconstruction. Cube maps use the six faces of a cube to create an omnidirectional view of the environment. The six projection centers of the virtual cube map cameras coincide with the origin of the laser scanner. We render the colored point cloud of a laser scan into these cameras, resulting in six sparse color and depth images per laser scan (*c.f.* Fig. 2). We fill in missing pixels not covered by a laser scan point by using the color of the nearest neighbor. While more complex rendering methods [39] could be used, we found that this strategy already suffices for feature matching in SfM. To obtain an initial estimate of the scale of the SfM model, we compare the depth of SfM points projected into the cube maps to the rendered depth maps.

Refinement of Laser Scan Alignment. The projection centers of the cube maps correspond to the origin of each laser scan. Thus, the SfM reconstruction from the previous step provides an initial relative alignment of the scans. We refine this alignment by jointly optimizing the rigid body poses of all laser scans via point-to-plane ICP [2] on the point clouds. Visual inspection verified that the resulting alignments are virtually perfect, which can be expected given the high accuracy of the laser scanner and the massive amount of information per scan. Thus, we fix the scan poses from here on.

Refinement of Image Alignment. In the last step of the pipeline, we refine the extrinsic and intrinsic parameters of the cameras while keeping the laser scan point cloud fixed. For this step, we use an extended version of the dense image alignment approach proposed by Zhou & Koltun [47].

Zhou & Koltun first sample a set of points $\mathcal{P}$ from a mesh surface and then optimize the camera parameters and the intensity $c(\mathbf{p})$ of each 3D point to minimize the cost function

$$\sum_{\mathbf{p}\in\mathcal{P}} \sum_{i\in\mathcal{I}(\mathbf{p})} \left(I_i(\pi_i(\mathbf{p})) - c(\mathbf{p})\right)^2 \quad . \tag{1}$$

Here, $\mathcal{I}(\mathbf{p})$ denotes the set of images in which point $\mathbf{p} \in \mathcal{P}$ is visible, $I_i(\pi_i(\mathbf{p}))$ is the intensity of image i at pixel coordinate $\pi_i(\mathbf{p})$ corresponding to $\mathbf{p}$'s projection. $c(\mathbf{p})$ denotes the intensity of $\mathbf{p}$, which belongs to the variables that are optimized. In our case, we use the joint point cloud from all scans for $\mathcal{P}$. To determine the visibility $\mathcal{I}(\mathbf{p})$, we compute a screened Poisson surface reconstruction [18] for $\mathcal{P}$. Since thin objects such as wires are often not captured by the reconstruction, we augment the mesh-based representation with splats, *i.e.*, oriented discs, generated for all scan points far away from the Poisson surface. $\mathcal{I}(\mathbf{p})$ is then determined from depth map renderings of the mesh and the splats. All points whose depth is smaller than that of the depth map rendering at their projected positions, plus a small tolerance of $1cm$, are assumed to be visible in the image.

Eq. 1 directly compares pixel intensities and thus assumes brightness constancy. However, this assumption is strongly violated in our setting, since we (1) use multiple cameras, some of which employ an auto-exposure setting, and (2) record outdoor scenes where strong lighting changes require manipulation of shutter time. Rather than directly comparing pixel intensities, we thus compare intensity gradients g, making our objective robust to brightness changes. Similar to computing finite difference gradients in an image, we compute the intensity gradients in the point cloud using local neighborhoods.

However, due to the different image resolutions and high laser scan point density in our dataset, the nearest neighbors of a point $\mathbf{p}$ might project to the same pixel in one image and to relatively far away pixels in another image. In the former case, the points' intensity differences are nearly constant and do not provide enough information for the optimization. In the latter case, under-sampling of the image leads to a meaningless intensity gradient. Consequently, we sample the neighborhoods at appropriate point cloud resolutions. We only add the projection of a point $\mathbf{p}$ to an image as a constraint if all neighbors of $\mathbf{p}$ project roughly one pixel away from it. This avoids the discussed case of over- and under-sampling. To create enough constraints between all

images and for all points, we efficiently sample the neighborhoods from a pre-calculated multi-resolution point cloud and we use a multi-resolution scheme over image pyramids. For each point projection to an image, the image pyramid level with the most suitable resolution is considered. This increases the chance that a gradient g is compared against at least two images and influences the respective camera parameters. Further, the multi-resolution scheme over images enlarges the convergence basin of the optimization. We process the image pyramid coarse-to-fine while coarser resolutions are kept in the objective function.

More concretely, we associate each point cloud level l with a point radius r_l in 3D. Since the changes made by the refinement of the image alignment will be small, we use the initial image alignment to determine the relevant point cloud levels. For each laser scan point $\mathbf{p}$ and each pyramid level h of each image $i \in \mathcal{I}(\mathbf{p})$, we determine the radius $r(i, h, \mathbf{p})$ of a 3D sphere around the 3D point $\mathbf{p}$ such that the projection of the sphere into image i at this pyramid level has a diameter of $\sim$1 pixel. To define the radius r_0 at the highest point cloud resolution, we use the minimum radius among all points and images. The radius of level $l+1$ is defined as $2r_l$. The minimum and maximum radii $r(i, h, \mathbf{p})$ of a point $\mathbf{p}$ define an interval, and a point is associated with level l if r_l falls into this range. At each level l, we greedily merge points within $2r_l$ using the mean position as the position of the resulting 3D point. For each resulting point, we find its 25 nearest neighbors and randomly select 5 of them to define the point's neighborhood. If the average intensity difference between a point and its neighbors is less than 5, we drop the point as it lies in a homogeneous region and thus would not contribute to the optimization.

Let $\mathbf{p}_j$ denote the j-th neighbor of point $\mathbf{p}$. The variables $g(\mathbf{p}, \mathbf{p}_j)$ are now associated to pairs of points and represent their gradient. We modify the cost function in Eq. 1 to take the following form

$$\sum_{\mathbf{p}\in\mathcal{P}} \sum_{i\in\mathcal{I}(\mathbf{p})} \rho\left[\sqrt{\sum_{j=1}^{5} \left(I_i(\pi_i(\mathbf{p}_j)) - I_i(\pi_i(\mathbf{p})) - g(\mathbf{p}, \mathbf{p}_j)\right)^2}\right] \tag{2}$$

where $\mathcal{P}$ contains all points of the multi-resolution point cloud and $\rho[\cdot]$ is the robust Huber loss function. Note that, in contrast to Eq. 1, we now represent *and* optimize for the gradients g of each 3D point rather than the point intensity c. Details on implementing this cost function can be found in the supplementary material, which also provides an illustration of the multi-resolution scheme.

For the sequences recorded with the multi-camera rig, we ensure that the relative poses between the cameras in the rig remain consistent for all images during optimization by a rigid parametrization of the relative camera poses. To speed up the optimization, we optimize $g(\mathbf{p}, \mathbf{p}_j)$ and the camera parameters alternatingly, as proposed in [47].

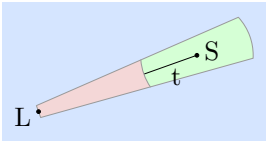


Figure 3. Sketch of the accuracy evaluation given a single scan point, S, measured from the laser scanner position L, with evaluation threshold t. Reconstruction points within the green region are accurate, points in the red region are inaccurate, and points in the blue region are unobserved.

In practice, we found this to result in a good relative alignment between images, but not necessarily in a good absolute alignment to the laser scans. We thus add an additional cost term in analogy to Eq. 2 which minimizes the intensity differences in the images wrt. the intensity differences measured by the laser scanner. This term lowers drift by using the laser scan colors as a global reference. As a limitation, it creates a dependency on the laser scan colors, which themselves may not be perfectly aligned to the scan geometry. However, we empirically found the resulting alignments to be of satisfying quality for our needs.

4. Tasks and Evaluation Protocols

Our benchmark consists of three scenarios corresponding to different tasks for (multi-view) stereo algorithms:

- High-resolution multi-view stereo with relatively few images recorded by a DSLR camera.
- Low-resolution multi-view stereo on video data ("many-view") recorded with a multi-camera rig.
- Low-resolution two-view stereo on camera pairs of a multi-camera rig.

Each frame of the two-view stereo evaluation consists of all 4 images taken at the same time by the multi-camera rig. These 4 cameras form 2 stereo pairs such that both cameras in each pair have the same FOV. Both multi-view stereo scenarios are evaluated in 3D with the same evaluation protocol, while the two-view stereo scenario is evaluated in 2D with a separate protocol, as detailed in the following.

Multi-View Stereo Evaluation Protocol. We compare the MVS reconstruction, given as a point cloud, against the laser scan ground truth of the scene. Only laser scan points visible in at least two images are used for evaluation.

We evaluate the reconstruction in terms of *accuracy* and *completeness*. Both measures are evaluated over a range of distance thresholds from 1cm to 50cm. To determine *completeness*, we measure the distance of each ground truth 3D point to its closest reconstructed point. *Completeness* is defined as the amount of ground truth points for which this distance is below the evaluation threshold.

Accuracy is defined as the fraction of reconstruction points which are within a distance threshold of the ground

Figure 4. Two examples of laser scan renderings colored by differently aligned images. Top row, from left to right: Original laser scan colors, initial alignment, 7-DoF ICP alignment, our alignment. Bottom row: Difference of each image to the laser scan image. Note that the different lighting causes a significant color difference.

truth points. Since our ground truth is incomplete, care has to be taken to prevent potentially missing ground truth points from distorting the results. Consequently, we segment the reconstruction into occupied, free, and unobserved space using an approximation of the laser scanner beams (*c.f.* Fig. 3). We model the shape of the laser beam of each ground truth point as a truncated cone. We make the assumption that the beam volume from the laser scanner origin to a scan point contains only free space.

We extend the beam volume beyond each ground truth point by the intersection of the extended beam cone with a sphere centered at the observed point. The sphere's radius is equal to the evaluation tolerance t. Reconstructed points outside all extended beam volumes are in unobserved space and are therefore discarded in the evaluation. Among the remaining reconstruction points, points are classified as accurate if they are within the extended beam volume and within radius t of a ground truth point. *Accuracy* is then defined as the ratio of accurate points out of all points while ignoring unobserved points.

The definitions of accuracy and completeness provided above are susceptible to the densities of both the reconstructed and the ground truth point clouds. For instance, an adversary could uniformly fill the 3D space with points to achieve high completeness while creating comparatively many more copies of a single reconstructed point known to be accurate to also achieve high accuracy. We thus discretized the space into voxels with small side length. Both measures are first evaluated for each voxel individually. We then report the averages over all voxels. To measure a voxel's completeness, we use the ground truth points in it and all reconstructed points, even those outside the voxel. These roles are reversed to measure accuracy.

Since both accuracy and completeness are important for measuring the quality of a reconstruction, we use the F_1 score as a single measure to rank the results. Given accuracy (precision) p and completeness (recall) r, the F_1 score is defined as the harmonic mean $2 \cdot (p \cdot r)/(p + r)$.

Two-View Stereo Evaluation Protocol. The two-view stereo evaluation is performed on rectified stereo pairs generated from the images of the multi-camera rig. The ground truth is given by the laser scan points projected into the rectified image. Occluded points are dropped using the same occlusion reasoning as in our image alignment procedure. The left disparity image is used for evaluation.

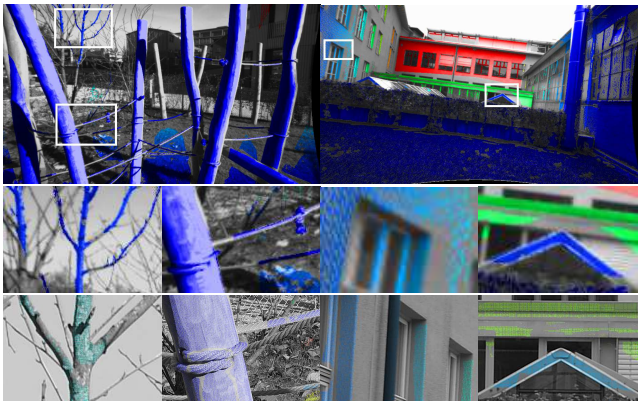


Figure 5. Top row: Overlays of ground truth depth (colored) onto images recorded by the multi-camera rig, showing the accuracy of our alignments. Middle & bottom row: Detailed views for rig and DLSR images, respectively. The ground truth depth is sparse at full DSLR resolution and not all objects are scanned completely.

For this scenario, we evaluate the same metrics used in the Middlebury benchmark [32]: We measure the percentage of pixels having a disparity error larger than a threshold, for thresholds of 0.5, 1, 2, and 4 disparities (*bad 0.5* - *bad 4*), the average absolute error in pixels (*avgerr*), the root-mean-square disparity error in pixels (*rms*), and the error quantiles in pixels for 50%, 90%, 95%, and 99% (*A50* - *A99*).

5. Results

First, we evaluate the accuracy of our image registration pipeline. Due to lack of more precise measurements, this evaluation is performed qualitatively. In Sec. 5.2, we then evaluate state-of-the-art algorithms on our benchmark and discuss the gained insights.

5.1. Image Registration

We compare our alignment strategy to the initial alignment obtained after the laser scan refinement step and to a baseline method. The latter refines the initial camera poses

through a 7-DoF ICP alignment (optimizing for position, rotation and scale) between the reconstructed SfM points and the laser scans. For each alignment result, we project all images onto the laser scans and compute the average color for each scan point to create the qualitative evaluations in Fig. 4. As can be seen, the baseline significantly improves the initial alignment. In turn, our alignment strategy clearly improves upon the baseline.

Fig. 5 shows overlays of images with their ground truth depth map computed from the laser scans. Depth edges are not used in our alignment procedure, thus they serve as a good indicator for the quality of the alignment. We observe that for both the DSLR images as well as the camera rig images, the alignment is generally pixel accurate.

5.2. Evaluation of Stereo Methods

High-Resolution Multi-View Scenario. For this scenario, we evaluate the popular patch-based PMVS [4], Gipuma [5], which is a well-performing PatchMatch-based variant [26], the multi-view stereo method based on pixel-wise view selection in COLMAP [35], and CMPMVS [16], which aims at reconstructing weakly supported surfaces. The results are shown in Fig. 6. We observe that for most scenes, COLMAP and PMVS outperform Gipuma and CMPMVS in terms of accuracy. In terms of completeness, Gipuma achieves a low score for, *e.g.*, *courtyard*, *electro*, and *delivery area*, since its view selection scheme is tailored to object-centric scenes. For most datasets, CMPMVS and COLMAP clearly achieve the best completeness. COLMAP still struggles for weakly textured surfaces and thin structures, such as *electro* (*c.f.* Fig. 8c), *kicker*, *office*, and *pipes*. As shown in Fig. 8d, CMPMVS is able to correctly interpolate some of the weakly textured surfaces, but also hallucinates structures in other parts.

Fig. 8b shows the cumulative completeness score over all methods for the *office* scene, illustrating that all existing techniques struggle to achieve high completeness for poorly textured surfaces. We believe that solving this hard but important problem requires higher-level scene understanding. Our benchmark provides a variety of scenes that can be used to evaluate such approaches. In general, we observe that there is significant room for improvement on our datasets.

Tab. 2a compares the relative performance of the different methods on Strecha [40] and our new benchmark, ranked based on [35] and using the F_1 score, respectively, with a 2cm evaluation threshold used in both cases. Evidently, good performance on the two datasets is not necessarily correlated. We thus conclude that our benchmark contains challenges different from the Strecha dataset.

Low-Resolution Multi-View Scenario. For this scenario, we evaluate the same methods as for the previous one. For Gipuma, we downsampled the videos to one fifth the frame rate since it ran out of memory while using all images. The results are shown in Fig. 7. As can be seen, these datasets challenge all algorithms, resulting in lower accuracy scores compared to the high-quality datasets. PMVS and Gipuma produce very incomplete and noisy results while CMPMVS fails completely. This demonstrates the fact that they do not properly exploit the high view redundancy in the videos. COLMAP achieves relatively better results, but there is still significant room for improvement in terms of absolute numbers. Furthermore, all methods take in the order of several minutes to an hour to compute the sequences, underlining the need for more efficient algorithms that can operate in real-time on a mobile device.

Method	Strecha	Ours
PMVS	3 (68.9)	3 (41.2)
Gipuma	4 (48.8)	4 (33.2)
COLMAP	2 (75.9)	**1 (64.7)**
CMPMVS	**1 (78.2)**	2 (48.9)

(a) MVS

Method	Middle.	KITTI	Ours
SPS-Stereo [44]	5 (29.3)	2 (5.3)	**1 (3.4)**
MeshStereo [46]	2 (14.9)	4 (8.4)	3 (7.1)
SGM+D. [13, 42]	4 (29.2)	3 (6.3)	2 (5.5)
MC-CNN [45]	**1 (10.1)**	**1 (3.9)**	4 (8.9)
ELAS [7]	3 (25.7)	5 (9.7)	5 (10.5)

(b) Stereo (% Bad Pixels)

Table 2. Relative rankings on different benchmarks, demonstrating the difference between ours and existing datasets.

Method	Indoor	Outdoor	Mobile	DSLR
CMPMVS	67.2 / 47.3 / 55.5	44.2 / 40.0 / 42.0	14.4 / 7.4 / 9.8	71.6 / **57.6** / 63.8
COLMAP	**90.2** / **51.1** / **65.2**	**80.9** / **53.1** / **64.1**	**69.5** / **41.2** / **51.8**	**91.7** / 56.2 / **69.7**
Gipuma	74.9 / 24.0 / 36.3	52.8 / 20.8 / 29.9	31.1 / 13.4 / 18.7	76.5 / 25.9 / 38.7
PMVS	85.1 / 28.0 / 42.1	72.2 / 27.8 / 40.1	48.7 / 18.8 / 27.2	90.1 / 31.3 / 46.5

Table 3. Category-based MVS evaluation showing accuracy / completeness / F_1 score (in %) at a 2cm evaluation threshold.

Method	bad 0.5	bad 1	bad 2	bad 4	avgerr	rms	A50	A90	A95	A99
SPS-Stereo [44]	57.42	22.25	**4.28**	**2.00**	0.95	**2.11**	**1.08**	**2.40**	**3.60**	**8.77**
SGM+D. [13, 42]	58.56	24.02	7.48	4.51	1.34	3.23	2.47	3.61	8.54	12.95
MeshStereo [46]	**29.91**	**13.98**	7.57	4.67	**0.90**	2.21	1.37	2.44	3.65	9.73
MC-CNN [45]	33.79	14.74	9.26	8.46	8.52	17.14	49.81	30.01	30.49	52.19
ELAS [7]	42.26	22.77	11.54	5.05	1.12	3.11	4.87	5.26	4.49	11.31
SPS-Stereo [44]	56.91	21.29	**3.43**	**1.43**	**0.83**	**1.61**	2.22	**1.36**	**2.11**	**6.52**
SGM+D. [13, 42]	57.79	22.43	5.48	2.65	1.03	2.43	**1.14**	7.05	6.46	10.69
MeshStereo [46]	**28.99**	**13.23**	7.09	4.38	0.87	2.18	1.61	2.55	4.46	10.65
MC-CNN [45]	32.51	13.85	8.92	8.59	8.48	17.03	49.01	27.63	28.79	45.79
ELAS [7]	41.20	21.56	10.50	4.56	1.06	3.03	4.54	3.98	5.24	9.68

Table 4. Results of two-view stereo methods on our dataset. We show the metric averages over all stereo pairs for all regions (upper part) and non-occluded regions (lower part). The tables are ordered by the *bad 2* criterion.

Dataset Diversity. Tab. 3 provides an analysis of the different MVS algorithms for different scenario categorizations. COLMAP performs best on average as well as best for most individual categories. We also observe that the performance of the algorithms can vary significantly between the different scenarios, indicating the need for benchmarks such as ours that cover a wide variety of scenes.

Two-View Scenario. For this scenario, we evaluated five methods, [7, 44–46] and a version of SGM stereo [13] on Daisy descriptors [42]. These include methods belonging to the state-of-the-art on KITTI and Middlebury Stereo. We did not tune their parameters except for setting the maxi-

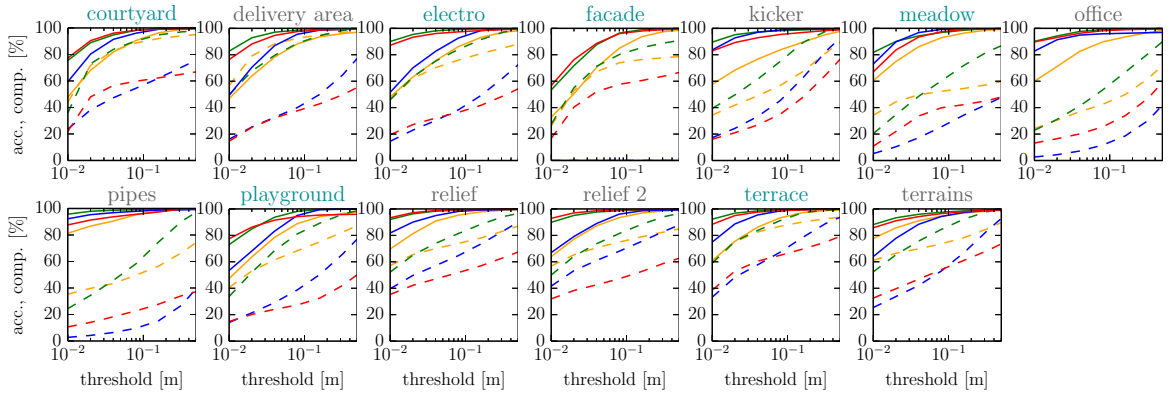


Figure 6. Evaluation of the high-resolution multi-view scenario (indoor and outdoor datasets). Results for CMPMVS, COLMAP, Gipuma, and PMVS are shown as a solid line for accuracy and as a dashed line for completeness.

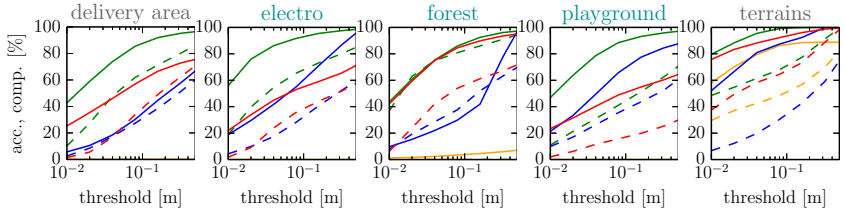
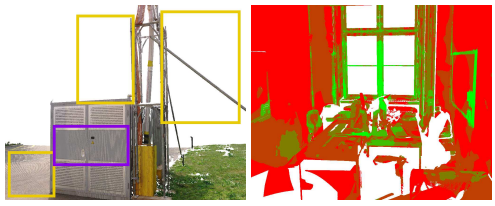
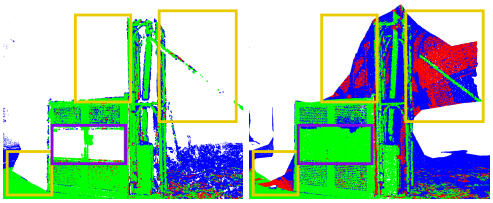


Figure 7. Multi-view evaluation results in the low-resolution scenario. The interpretation of the plots is the same as in Fig. 6.



(a) Electro: Laserscan (b) Office: Completeness



(c) Electro: COLMAP (d) Electro: CMPMVS

Figure 8. Qualitative results. See the text for details.

mum number of disparities. The evaluation results are presented in Table 4. Table 2b compares the relative rankings of the different approaches on KITTI, Middlebury, and the *bad 2* non-occluded results of our benchmark. As can be seen, the rankings differ significantly between ours and previous datasets, indicating that our benchmark complements existing ones. In particular, our data requires algorithms to perform well over a wide variety of scenes. It thus encourages general solutions and prevents overfitting. The latter is especially important given the popularity of learning-based methods: As evident from Tabs. 4 and 2b, [45] performs below average on our benchmark while outperforming all other methods significantly on *both* Middlebury *and* KITTI.

6. Conclusion

In this paper, we proposed an accurate and robust registration procedure to align images and laser scans. Using this algorithm, we created a new and diverse dataset for the evaluation of two-view and multi-view stereo methods. Our benchmark differs from existing datasets in several key aspects: We cover a wide variety of scene types and thus require general solutions which prevent overfitting. In addition, we provide the first benchmark for hand-held (multi-view) stereo with consumer-grade cameras. Experimental results for state-of-the-art algorithms show that our dataset poses various challenges not yet covered by existing benchmarks. One of these challenges is efficient processing of large amounts of data, in the form of both high spatial and high temporal sampling. These challenges are far from being solved and there is significant room for improvement. As a service to the community, we provide the website http://www.eth3d.net for online evaluation and comparison of algorithms.

Acknowledgements. We thank Fatma Güney for running several stereo baselines on KITTI and Middlebury, Lukas Meier for the multi-camera rig mount and the Geosensors and Engineering Geodesy group at ETH for providing the laser scanner. Thomas Schöps was supported by a Google PhD Fellowship. This project received funding from the European Union's Horizon 2020 research and innovation programme under grant No. 688007 (Trimbot2020). This project was partially funded by Google Tango.

References

[1] D. J. Butler, J. Wulff, G. B. Stanley, and M. J. Black. A naturalistic open source movie for optical flow evaluation. In *ECCV*, 2012. 2, 3

[2] Y. Chen and G. Medioni. Object modelling by registration of multiple range images. *Image and vision computing*, 10(3):145–155, 1992. 4

[3] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Towards internet-scale multi-view stereo. In *CVPR*, 2010. 1

[4] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. *PAMI*, 32(8):1362–1376, 2010. 1, 7

[5] S. Galliani, K. Lasinger, and K. Schindler. Massively parallel multiview stereopsis by surface normal diffusion. In *ICCV*, 2015. 1, 7

[6] A. Geiger, P. Lenz, and R. Urtasun. Are we ready for autonomous driving? The KITTI vision benchmark suite. In *CVPR*, 2012. 1, 2, 3

[7] A. Geiger, M. Roser, and R. Urtasun. Efficient large-scale stereo matching. In *ACCV*, 2010. 1, 7

[8] A. Geiger, J. Ziegler, and C. Stiller. StereoScan: Dense 3D reconstruction in real-time. In *IV*, 2011. 1

[9] P. Gohl, D. Honegger, S. Omari, M. Achtelik, M. Pollefeys, and R. Siegwart. Omnidirectional Visual Obstacle Detection using Embedded FPGA. In *IROS*, 2015. 3

[10] C. Häne, T. Sattler, and M. Pollefeys. Obstacle detection for self-driving cars using only monocular cameras and wheel odometry. In *IROS*, 2015. 1

[11] C. Häne, C. Zach, J. Lim, A. Ranganathan, and M. Pollefeys. Stereo depth map fusion for robot navigation. In *IROS*, 2011. 1

[12] H. Hirschmuller and D. Scharstein. Evaluation of Cost Functions for Stereo Matching. In *CVPR*, 2007. 1

[13] H. Hirschmüller. Stereo processing by semiglobal matching and mutual information. *PAMI*, 30(2):328–341, 2008. 7

[14] D. Honegger, H. Oleynikova, and M. Pollefeys. Real-time and low latency embedded computer vision hardware based on a combination of FPGA and mobile CPU. In *IROS*, 2014. 1

[15] Q. Huang, H. Wang, and V. Koltun. Single-view reconstruction via joint analysis of image and shape collections. In *SIGGRAPH*, 2015. 4

[16] M. Jancosek and T. Pajdla. Multi-view reconstruction preserving weakly-supported surfaces. In *CVPR*, 2011. 7

[17] R. Jensen, A. Dahl, G. Vogiatzis, E. Tola, and H. Aanæs. Large scale multi-view stereopsis evaluation. *CVPR*, 2014. 1, 3

[18] M. M. Kazhdan and H. Hoppe. Screened poisson surface reconstruction. *SIGGRAPH*, 32(3):29, 2013. 4

[19] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun. Tanks and temples: Benchmarking large-scale scene reconstruction. *SIGGRAPH*, 36(4), 2017. 3

[20] K. Kolev, P. Tanskanen, P. Speciale, and M. Pollefeys. Turning mobile phones into 3D scanners. In *CVPR*, 2014. 1

[21] L. Ladicky, P. Sturgess, C. Russell, S. Sengupta, Y. Bastanlar, W. Clocksin, and P. Torr. Joint optimisation for object class segmentation and dense stereo reconstruction. In *BMVC*, 2010. 2

[22] D. G. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, 2004. 4

[23] N. Mayer, E. Ilg, P. Haeusser, P. Fischer, D. Cremers, A. Dosovitskiy, and T. Brox. A large dataset to train convolutional networks for disparity, optical flow, and scene flow estimation. In *CVPR*, 2016. 2

[24] M. Menze and A. Geiger. Object scene flow for autonomous vehicles. In *CVPR*, 2015. 2, 3

[25] P. Merrell, P. Mordohai, J.-M. Frahm, and M. Pollefeys. Evaluation of large scale scene reconstruction. In *ICCV*, 2007. 3

[26] C. R. Michael Bleyer and C. Rother. PatchMatch Stereo - Stereo Matching with Slanted Support Windows. In *BMVC*, 2011. 7

[27] Y. Nakamura, T. Matsuura, K. Satoh, and Y. Ohta. Occlusion detectable stereo - occlusion patterns in camera matrix. In *CVPR*, 1996. 2

[28] P. Ondrúška, P. Kohli, and S. Izadi. MobileFusion: Real-time Volumetric Surface Reconstruction and Dense Tracking On Mobile Phones. In *ISMAR*, 2015. 1

[29] C. J. Pal, J. J. Weinman, L. C. Tran, and D. Scharstein. On Learning Conditional Random Fields for Stereo. *IJCV*, 99(3):319–337, 2012. 1

[30] S. Pillai, S. Ramalingam, and J. J. Leonard. High-Performance and Tunable Stereo Reconstruction. In *ICRA*, 2016. 1

[31] R. B. Rusu, Z. C. Marton, N. Blodow, M. E. Dolha, and M. Beetz. Towards 3d point cloud based object maps for household environments. *RAS*, 56(11):927–941, 2008. 3

[32] D. Scharstein, H. Hirschmüller, Y. Kitajima, G. Krathwohl, N. Nešić, X. Wang, and P. Westling. High-resolution stereo datasets with subpixel-accurate ground truth. In *GCPR*, 2014. 1, 2, 3, 6

[33] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *IJCV*, 47:7–42, 2002. 1, 2, 3

[34] D. Scharstein and R. Szeliski. High-accuracy Stereo Depth Maps Using Structured Light. In *CVPR*, 2003. 1

[35] J. L. Schönberger, E. Zheng, M. Pollefeys, and J.-M. Frahm. Pixelwise view selection for unstructured multi-view stereo. In *ECCV*, 2016. 1, 4, 7

[36] T. Schöps, T. Sattler, C. Häne, and M. Pollefeys. 3D modeling on the go: Interactive 3D reconstruction of large-scale scenes on mobile devices. In *3DV*, 2015. 1

[37] J. L. Schönberger and J.-M. Frahm. Structure-from-motion revisited. In *CVPR*, 2016. 4

[38] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, 2006. 1, 2, 3

[39] D. Sibbing, T. Sattler, B. Leibe, and L. Kobbelt. Sift-realistic rendering. In *3DV*, 2013. 4

[40] C. Strecha, W. von Hansen, L. J. V. Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *CVPR*, 2008. 1, 2, 3, 7

[41] P. Tanskanen, K. Kolev, L. Meier, F. Camposeco, O. Saurer, and M. Pollefeys. Live metric 3D reconstruction on mobile phones. In *ICCV*, 2013. 1

[42] E. Tola, V. Lepetit, and P. Fua. Daisy: An efficient dense descriptor applied to wide baseline stereo. *PAMI*, 32(5):815–830, May 2010. 7

[43] T. Vaudrey, C. Rabe, R. Klette, and J. Milburn. Differences between stereo and motion behaviour on synthetic and real-world stereo sequences. In *IVCNZ*, 2008. 2

[44] K. Yamaguchi, D. McAllester, and R. Urtasun. Efficient joint segmentation, occlusion labeling, stereo and flow estimation. In *ECCV*, 2014. 7

[45] J. Zbontar and Y. LeCun. Stereo matching by training a convolutional neural network to compare image patches. *Journal of Machine Learning Research*, 17:1–32, 2016. 7, 8

[46] C. Zhang, Z. Li, Y. Cheng, R. Cai, H. Chao, and Y. Rui. Meshstereo: A global stereo model with mesh alignment regularization for view interpolation. In *ICCV*, 2015. 7

[47] Q. Zhou and V. Koltun. Color map optimization for 3d reconstruction with consumer depth cameras. In *SIGGRAPH*, 2014. 4, 5