2020 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)
October 25-29, 2020, Las Vegas, NV, USA (Virtual)

# NBVC: A Benchmark for Depth Estimation from Narrow-Baseline Video Clips

Philippos Mordohai, Konstantinos Batsos, Ameesh Makadia and Noah Snavely

*Abstract*—
**We present a benchmark for online, video-based depth estimation, a problem that is not covered by the current set of benchmarks for evaluating 3D reconstruction, which focus on offline, batch reconstruction. Online depth estimation from video captured by a moving camera is a key enabling technology for compelling applications in robotics and augmented reality. Inspired by progress in many aspects of robotics due to benchmarks and datasets, we propose a new benchmark called NBVC for evaluating methods for online depth estimation from video. Our benchmark is composed of short video sequences with corresponding high-quality ground truth depth maps, derived from the recent Tanks and Temples dataset. We are hopeful that our work will be instrumental in the development of learning-based algorithms for online depth estimation from video clips, and will also lead to improvements in conventional approaches. In addition to the benchmark, we present a superpixel-based plane sweeping stereo algorithm and use it to investigate various aspects of the problem. The paper contains our initial findings and conclusions.**

## I. Introduction

Benchmarks and their corresponding datasets have led to dramatic progress in most areas of robotic perception. While some would argue that progress is occasionally followed by saturation and overfitting, most researchers agree that the availability of widely used datasets with ground truth enables continuous evaluation and improvement of individual algorithms as well as comparisons among various approaches providing valuable information to researchers.

In this paper, we introduce NBVC, a dataset and benchmark for online, dense 3D modeling from video that addresses a need that is not met by current publicly available datasets. We are motivated by applications that require estimation of dense depth maps on mobile devices and robots, including autonomous navigation, driver assistance, robotic perception, and augmented reality. The key characteristic of these scenarios is that imagery is acquired by a camera moving continuously through space, resulting in narrow baselines between successive frames. In addition, the output must be generated as quickly as possible because high latency is unacceptable in many cases.

Philippos Mordohai and Konstantinos Batsos are with the Dept. of Computer Science, Stevens Institute of Technology, NJ, USA {pmordoha,kbatsos}@stevens.edu

Ameesh Makadia is with Google Research, USA makadia@gmail.com

Noah Snavely is with the Computer Science Department at Cornell Tech, NY, USA and with Google Research, USA snavely@cs.cornell.edu

Part of this research was carried out while the first author was visiting Google Research. It was also supported in part by the National Science Foundation (IIS 1527294 and IIS 1637761).

While there is strong interest in applications of real-time, video-based 3D reconstruction, there is no suitable dataset with ground truth that would enable researchers to evaluate their algorithms. To obtain competitive results on the Middlebury [1], DTU MVS [2], ETH3D [3] and the Tanks and Temples (TnT) [4] benchmarks authors resort to batch processing to maximize data utilization, while the EuRoC MAV dataset [5] has not been adopted for dense reconstruction. The recent datasets focusing on self-driving [6], [7], [8], [9] are large, but not well-suited for UAV-mounted or handheld cameras due to differences in camera motion. We derive our dataset from the raw data in the TnT dataset, which comprises video sequences from a single camera but restrict the camera motion to focus on low-latency 3D reconstruction.

Despite the similarity in content, there are substantial differences between our dataset and TnT. Even though the TnT videos can be used in an online fashion, all current entries in the leaderboard treat the video frames as an image collection, aiming to leverage large baselines and maximize coverage on the surfaces of interest. The evaluation criteria are precision and recall, without considering run time. This setup favors batch processing, large-scale bundle adjustment and computationally expensive multi-view stereo (MVS) algorithms. We consider this research direction largely orthogonal to our objectives. An additional difference is that TnT does not provide camera poses, instead encouraging authors to tackle pose estimation and dense reconstruction jointly. We plan to release camera poses (see Section III), since in the scenarios we are interested in, additional navigation sensors, such as GPS, gyroscopes and accelerometers, are often available along with software that fuses their estimates with video to generate pose estimates. In contrast to TnT, we plan to include throughput in the evaluation metrics.

We call our benchmark NBVC because our focus is on online depth estimation from *Narrow-Baseline Video Clips*. We expect methods that use our data to take advantage of the known sequence of the frames, for example by assuming small displacements between adjacent frames, by tracking features or by using optical flow. Due to the short length of the clips, a viewpoint-based representation of the reconstruction in the form of depth maps is effective for many applications. Therefore, we provide ground truth depth maps with the training set and use per pixel depth accuracy in our evaluation metrics.

The NBVC benchmark comprises 433 short clips, containing at least 60 frames each for a total of 37,955 frames, extracted from six scenes from the training set of Tanks and

Fig. 1. Frames (left) and ground truth depth maps (right) from the Ignatius and Church scenes

Temples. It is, therefore, of larger scale than each of the KITTI benchmarks that provide 400 shorter stereo sequences for training and testing. NBVC contains more images than the high-resolution ETH3D benchmark which provides 898 total images from 25 scenes, as well as its low-resolution counterpart that contains 10,008 total images from 10 scenes. It should be noted, however, that we expect only a small number of images from each clip of NBVC to be used at a given time. A few examples are shown in Figs. 1 and 2. See Section III for details on how NBVC was created. We plan to release the images and camera poses, the ground truth of the training set, as well as the source code of the baseline algorithms, and to create a website to host the benchmark following the guidelines of recent benchmarks [3].

In order to explore the data, we estimate depth maps using conventional plane sweeping stereo [10] and a novel superpixel-based plane sweeping stereo algorithm that does not require exhaustive photoconsistency computation. We refer to the algorithms as PS and SBPS, respectively, and present them in Section IV. Plane sweeping is well suited for this setting because it can achieve high throughput, it can be applied to an arbitrary number of images and it does not require epipolar rectification. In Section V, we present results by varying several parameters of the algorithms and the camera configuration, such as the baseline, number of matching views, and the matching function. We show that a small fraction of pixels with depth estimates, as low as 5%, is adequate for fitting planes to the superpixels. We also run experiments using COLMAP [11] and MVSNet [12], noting that they have not been designed for our target applications. We expect that the results of Section V will be useful to researchers, and that they will be able to reach conclusions relevant to their work using the benchmark.

The main contributions of the paper are:

- the NBVC dataset for online depth estimation from Narrow-Baseline Video Clips,
- a simple, but effective, superpixel based plane sweeping stereo algorithm, and
- a thorough investigation of the effects of several aspects of camera configuration on depth map estimation accuracy on the above data.

## II. Related Work

In this section, we review relevant benchmarks and approaches for video-based and multi-view stereo that would benefit from our dataset. We refer readers to surveys on multi-view stereo [13], [1] for broader coverage on the topic.

The first multi-view benchmark was hosted at Middlebury College [1] and comprised two scenes with withheld ground truth and three image sets of different density for each scene. A second effort was undertaken by Strecha et al. [14], but was only active for a few years and the ground truth for only two scenes was released. The EuRoC MAV dataset [5] contains data captured by MAV-mounted cameras and ground truth acquired by LIDAR in two indoor locations. It has been used for structure from motion experiments, but not dense reconstruction. The first dataset large enough for supervised learning was the DTU MVS Data Set [2]. It contains 80 scenes captured under different lighting conditions from 49 or 64 viewpoints. While it is valuable for training, the lack of a test set with withheld ground truth and of a leaderboard has limited its impact.

These weaknesses are not shared by the most recent benchmarks. ETH3D [3] includes a low- and a high-resolution dataset, each with multiple scenes imaged from several viewpoints. Both datasets are divided evenly into training and test sets totaling dozens of scenes, hundreds of high-resolution and thousands of low-resolution images. Ground truth has been released only for the training sets. The Tanks and Temples dataset [4], which is the foundation of our benchmark, was acquired by a high-end camera indoors and outdoors. The dataset contains thousands of images of each of the 21 scenes, as well as ground truth acquired by a LIDAR sensor. The goal of Tanks and Temples is to assess the precision and recall of dense reconstruction using a large number of views. We aim to evaluate depth map estimation accuracy from short video clips without the benefits of long baselines.

As stated in Section I, we consider the recent datasets focusing on self-driving [6], [7], [8], [9] complementary to ours due to differences in camera motion and their restricted domain. Our benchmark is clearly more relevant for depth estimation from UAVs or handheld cameras for AR.

The most relevant algorithms to our research are those that generate 3D models from video by processing the frames in sequence. While impressive results in terms of processing speed and visual quality have been achieved [15], [16], [17], [18], [19], [20], [21], [22], [23], evaluation is almost exclusively qualitative. We believe that the inability to objectively and automatically evaluate these algorithms has hindered progress. Our benchmark can have significant impact by enabling the necessary evaluations.

In the past few years, stereo methods that do not require exhaustive photoconsistency computations have been published. PatchMatch stereo [24] can estimate disparity values and surface normals relying on randomized search and propagation. After random initialization, photoconsistent planes are discovered by random sampling and are then

propagated to neighboring pixels. PatchMatch has inspired binocular [25], [26] and multi-view [27], [28], [29], [30], [31], [32], [11] algorithms that do not estimate photoconsistency exhaustively. However, a fraction of the depth range of every single pixel is explored and the algorithms are not particularly fast due to sampling for depths and normals.

The Local Plane Sweep algorithm of Sinha et al. [33] clusters matched interest points to form disparity plane hypotheses. Local plane sweep problems are then defined around each plane. ELAS [34] evaluates photoconsistency, over the entire disparity range, for only a subset of the pixels. Reconstructed pixels are used to form a piecewise planar approximation of the scene. Segmentation is used in the algorithm of LeGendre et al. [35], which fits planes to image segments based on a sparse set of reconstructed 3D points. As opposed to SBPS, it is strictly binocular since it requires rectified images, but includes an additional step in which plane hypotheses are propagated among neighboring segments.

Plane-sweeping stereo is relevant to our work because it can achieve high throughput, as shown by Gallup et al. [10] who leveraged the parallelization capabilities of GPUs. Gallup et al. used the sparse points reconstructed during pose estimation to identify dominant plane orientations in the scene and reduce the effects of fronto-parallel bias. An extension to fisheye and non-pinhole cameras was published by Häne et al. [36]. Several deep learning MVS algorithms also rely on plane sweeping to associate image patches and then apply learned photoconsistency functions [37], [38], [39], [40], [41], [42], [43], [12]. Our work bears some similarity to methods specifically designed for small camera motion [44], [45], but our goals are different. Most of the above methods, however, similarly to video-based reconstruction research, have not been evaluated quantitatively.

## III. Benchmark Creation

In this section, we describe how video clips with dense ground truth depth maps were created from the training data of the Tanks and Temples benchmark [4]. (We could not apply our technique to the test data of the TnT bechmark, even if we were granted access to it, because this would partially release the ground truth of the test set and corrupt the benchmark.) As inputs we use the following data, which are available on the TnT website:

- the video sequence of each scene, captured at 29.97 frames per second (fps), from which we extract the *dense sequence* for each scene,
- the *sparse sequences* which have been sampled from the videos at 1 fps,
- poses for the sparse sequences estimated using COLMAP [46],
- rough estimates of camera intrinsics,
- ground truth meshes obtained by merging multiple range scans of each scene.

The ground truth meshes contain only the object of interest without background or transient objects. For example, the ground truth for Barn contains only the barn itself without the ground or the surrounding trees.

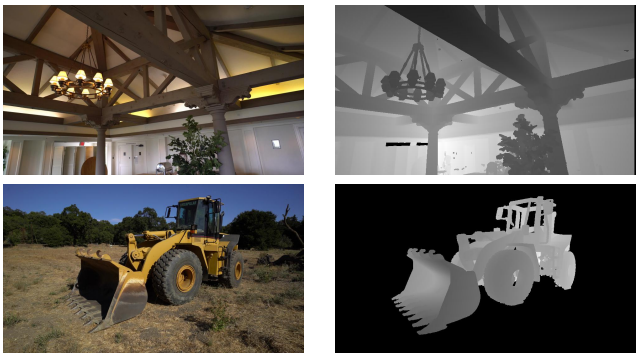


Fig. 2. Example reference frames and rendered ground truth depth maps from the Meetingroom, and Caterpillar scenes

**Rendering Ground Truth Depth Maps.** The first step is to render depth maps for the frames of the sparse sequence, those with poses estimated using COLMAP. The alignment of the ground truth model and the COLMAP coordinate system is provided at the TnT website. The rendered depth maps are dense on the objects of interest but contain no depths for other objects or the ground in some scenes. Figures 1 and 2 show examples of reference frames and ground truth depth maps. We determine the depth range for matching by extending the actual depth range of the ground truth by 5% towards and away from the camera. The depth range will be provided for both the training and the test data.

**Refinement of Intrinsics.** TnT aims at evaluating complete 3D reconstruction systems including the estimation of intrinsics. We, on the other hand, assume that users have access to the cameras and are able to calibrate them before deployment. Therefore, we refined the provided intrinsics by manually clicking corresponding points in a few images and the corresponding rendered depth maps.

**Pose Estimation for Dense Sequences.** Poses for the sparse sequences are provided with the dataset, but we must estimate poses for the dense sequences as well. We applied ORB-SLAM2 [47] on each dense sequence, we then split the sequences into short clips (see below) and refined the poses using *bundle adjustment on each clip separately.*

**Sequence Alignment.** A critical step is the alignment of the sparse and dense sequence of each scene. Since some dense sequences contain over 10,000 frames, computing a single global aligning transformation with the corresponding sparse sequence is bound to be inaccurate locally. As a result the ground truth depth maps would not align well with the images, limiting the usefulness of our dataset. We opted for estimating a large number of local transformations that ensure precise alignment of the ground truth depth maps with the corresponding reference frames. We accomplished this by first splitting the dense sequences into clips. Each clip contains three frames that appear in both the sparse and dense sequence: its first and last frames and a middle frame,

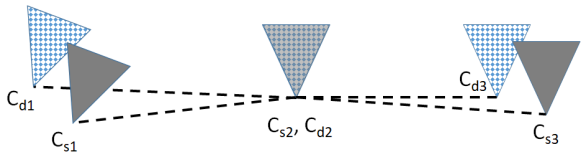



Fig. 3. Illustration of the alignment process. We first perfectly match the poses of the reference camera of the sparse and dense sequences denoted by $C_{s2}$ and $C_{d2}$, respectively. We then use the other two corresponding frames of each sequence to set the scale. The distance from the camera center of $C_{d1}$ to that of $C_{d3}$ is scaled to match the distance between $C_{s1}$ and $C_{s3}$. The clip is rejected if the angle of the rotation that would align the orientation of $C_{s1}$ and $C_{d1}$ is over $3^o$ or if the distance between these two cameras is more than 5% of the length of the clip, that is the distance between $C_{s1}$ and $C_{s3}$. The same criteria are then applied to $C_{s3}$ and $C_{d3}$.

which serves as reference and is associated with a rendered ground truth depth map.

The alignment of a clip begins by rotating and translating the reference camera in the dense sequence so that it perfectly matches the reference camera of the sparse sequence. The latter is chosen as the target since it is represented in a coordinate system with correct absolute scale. We then estimate a scaling factor that makes the distance from the first to the last frame of the dense sequence equal to the distance between the corresponding frames of the sparse sequence.

If trajectory estimation was perfect for both sequences, the three corresponding frames would now be perfectly aligned. This is not the case, however, due to noise. A clip is rejected if either the maximum rotation error between corresponding frames is larger than 3 degrees, or the maximum translation error is larger than 5% of the distance between the first and last frame. See Fig. 3 for an illustration of the alignment process. In practice, the clips are well calibrated especially near the reference image.

**Dataset Description.** The above steps are applied to all seven scenes of the TnT training set. The effectiveness of ORB-SLAM2 depends heavily on scene characteristics such as the lack of texture and the presence of specularities. Tracking failures and drift cause errors in sequence alignment and the rejection of the affected clips. We also manually reject clips in which the ground truth is heavily occluded by unmodeled objects. This phenomenon is particulary severe in the Courthouse data, in which the courthouse is occluded by trees, tents and other objects leading to the rejection of the entire scene. We also reject clips of the Barn in which the barn is occluded by trees that are not included in the ground truth mesh. The other scenes do not contain unmodeled objects, but many clips in Church and Meetingroom are rejected due to imprecise alignment with the sparse trajectory.

We then split the data into non-overlapping training and test sets. We decided against having different frames from the same scene in both sets to avoid overfitting. The scenes in the training set match some characteristics of scenes in the test set: each set contains an interior scene and each contains a vehicle. Finally, we assigned the scenes with the largest number of clips to the training set and obtained a 2:1 ratio of training to test data. The training set comprises Truck, Ignatius and Church with 180, 97 and 13 clips, respectively, while the test set comprises Caterpillar, Barn and Meetingroom with 72, 61 and 10 clips, respectively.

Each clip includes three frames of the sparse sequence. Since the dense sequence is sampled at 30 fps and the sparse sequence is sampled at 1 fps, each clip contains at least 61 frames and one ground truth depth map for the reference frame. The average baseline between the reference frame and the frames immediately adjacent to it ranges from 1.1 cm in Meetingroom to 2.3 cm in Church. The average baseline from the reference frame to the $30^{th}$ frame before or after it ranges from 31 to 67 cm, also for Meetingroom and Church respectively. The maximum depth varies between 3.87 m for Ignatius and 20.39 m for Barn. Triangulation angles near the maximum depth using images separated by approximately 10 baselines are in the order of $2^o$.

**Evaluation Protocol.** Depth maps are an effective representation for our purposes, due to the small baseline and short duration of the clips. We require that authors submit dense depth maps, the accuracy of which will be evaluated on a per-pixel basis on all pixels with ground truth. This protocol bypasses the complications arising from missing data in point cloud based evaluations, such as those of ETH3D [3] and DTU [2]. In these benchmarks, regions of 3D space had to be carefully designated as unobserved and reconstructed points in them are ignored, leading to unpenalized gross outliers. Moreover, as point density varies with distance from the camera, additional provisions are required to ensure fair evaluation. Measuring errors on depth maps does not suffer from these limitations.

To jointly evaluate accuracy and processing speed, we plan to use the harmonic mean (HAR) of a measure of each, as in the f-score which is widely used in classification. Specifically, we choose the *fraction of pixels with relative depth error under 5%* to measure accuracy and the *ratio of the frame rate over a nominal frame rate of 15 fps* for speed. Relative depth errors are defined per pixel as: $r(x, y) = |d(x, y) - g(x, y)|/g(x, y)$, where $d(x, y)$ is the depth estimate of a pixel and $g(x, y)$ is the ground truth depth. We refer to this measure as *FI* (Fraction of Inliers). Using a relative error measure allows averaging across scenes, in contrast to absolute depth errors that depend on the depth range. Speed and accuracy measures will be averaged over scenes, not images, to avoid bias towards scenes with more clips. (We also report other measures of accuracy, such as the mean and median absolute error per pixel.)

Authors will self-report the run time of their algorithm after reading the inputs and before writing the output as well as the specifications, such as CPU, GPU and RAM, of their system. As is often the case, we will rely on the integrity of our fellow researchers for this aspect of the evaluation. We envision grouping algorithms according to the capacity of their processing platforms. Submission of results from the same algorithm will be allowed every 10 days. This practice is currently accepted [48], [3] as a good trade-off between providing some feedback to researchers and overfitting.

## IV. Superpixel-based Plane Sweeping Stereo

We use two fast multi-view stereo (MVS) algorithms to estimate depth maps from the above clips. The first algorithm performs plane sweeping stereo similar to [10], while the second is a superpixel-based plane sweeping algorithm. We will refer to the algorithms as PS and SBPS, respectively. Plane sweeping stereo is well-suited to our problem because it is fast, it estimates a depth map for the reference frame using an arbitrary number of target frames and it does not require the images to be rectified.

The PS algorithm is a straightforward implementation of the algorithm of Gallup et al. [10] *on the CPU*. Depth is estimated for each pixel of the reference frame in winner-take-all (WTA) fashion by estimating the photoconsistency of the pixel at multiple depths, defined by a family of planes that are swept through the scene. Photoconsistency is estimated by defining a window centered at the pixel in the reference frame, projecting the window to each of the target frames via the current plane and computing a matching function, such as the sum of absolute differences (SAD) or normalized cross-correlation (NCC). The process is repeated for all depth values (planes) for each pixel. At the end the maximally photoconsistent depth is assigned to each pixel.

SBPS is inspired by the limitations of PS and attempts to address them while reducing the computational cost at the same time. The two key ideas are that piecewise planar reconstructions are typically effective in terms of accuracy and visual quality and that only a small number of 3D points are required to fit a plane. After the reference image has been segmented into superpixels using SLIC [49], we randomly select a fraction of the pixels and depth is estimated only for these pixels using plane sweeping over all possible depths (planes) as described above. Then, a plane is fitted to each superpixel using RANSAC on pixels with depth.

SBPS has the following advantages: First, superpixel-wise depth estimation leads to substantially fewer outliers than pixel-wise depth estimation. Second, computation is accelerated by estimating photoconsistency for only a small fraction of the pixels of each superpixel. Estimating the photoconsistency of only 5-10% of all pixels leads to negligible loss in accuracy compared fitting planes on fully dense depth maps using RANSAC. Third, fitting planes to the superpixels almost entirely eliminates the fronto-parallel bias, even though planes are swept in only one direction, and obtains depth estimates with sub-plane precision.

## V. Experimental Results

We have performed a comprehensive evaluation of all critical configuration parameters using both plane sweeping (PS) and the superpixel-based plane sweeping (SBPS) algorithm. Our tests shed light on the effects of parameters of the camera configuration such as the number of target frames and the baseline between the reference and target frames; and parameters of the matching function such as the choice of the function itself and of the window size. For SBPS only, we also considered segmentation parameters such as the size of the superpixels and the degree of regularization [49]; plane fitting parameters including the fraction of pixels for which depth is estimated, the threshold used by RANSAC and the number of iterations. Due to lack of space we present a subset of the results in this section and will provide additional results in a technical report.

All depth maps contain depth estimates for all pixels, but evaluation is limited to pixels with ground truth. Images are processed at quarter resolution ($960\times540$) compared to the TnT originals. We used gSLICr [50] on an NVIDIA TitanX GPU, followed by dense or sparse plane sweeping parallelized using OpenMP on an Intel Core i7-5820K at 3.3 GHz. Plane fitting was also run on the CPU. The number of RANSAC iterations was set to 100 and the threshold was set equal to the distance between consecutive planes.

**Baseline.** Tables I and II show mean absolute error (MAE) per pixel and FI, respectively, using SAD in $11\times11$ windows as the matching function between the reference and *one target* frame. We vary the number of frames separating the target and reference frame and refer to this gap as the *baseline*. A baseline of 1 means that the two frames are adjacent in the video. The most important observations are: (i) if the baseline is too small, accuracy suffers, (ii) SBPS is better in MAE because it prevents gross outliers, and (iii) PS typically obtains higher FI values because it estimates individual depths per pixel without planar approximations.

**Number of Images.** We also investigated the effect of the number of target images. We performed trinocular matching using two target frames at the same baseline before and after the reference frame, using the same baselines as in Tables I and II. The additional target frame led to a reduction of MAE of 10.3% for PS and 7.1% for SBS and increases in FI of 12% and 10.2%, respectively. These come at the cost of essentially doubling the runtime of plane sweeping, which is linear in the number of target views. Extending the configuration to include three target views on each side of the reference, MAE drops approximately by 16.7% for PS and 7.5% for SBPS. (We compare binocular and 7-view configurations based on their widest baseline.)

**Matching Function.** We conducted similar experiments using NCC as the matching function. Comparing Tables I and II with MAE and FI results using NCC as the matching function, we observe a reduction of MAE of 4.9% for PS and 3.1% for SBS and increases in FI of 4.1% and 3.2%, respectively. These differences are small, while runtimes are also similar with our implementations.

**Effects of Bundle Adjustment.** Performing bundle adjustment (BA) on each clip substantially improves accuracy. We repeated the experiments in Table I on the same clips, but without BA. BA substantially reduces the MAE as the baseline grows. When the baseline is 1, both PS and SBPS are within 2% in MAE regardless of BA. PS improves by 19% and 26% when the baseline is 11 and 21, respectively, while the same figures for SBPS are 18% and 27%. This is expected since wider baselines require more precise calibration to keep epipolar errors acceptable for matching.

TABLE I

MEAN ABSOLUTE DEPTH ERROR (MAE) PER PIXEL IN METERS, AVERAGED OVER REFERENCE IMAGES, USING ONE TARGET IMAGE AND SAD IN 11×11 WINDOWS. BASELINE REFERS TO THE SEPARATION BETWEEN THE REFERENCE AND TARGET FRAMES IN NUMBER OF FRAMES IN THE SEQUENCE. ONLY **5%** OF THE PIXELS HAVE BEEN SAMPLED FOR SBPS, WHILE DEPTH FOR ALL PIXELS IS ESTIMATED BY PS. SBPS OPERATES AT 1.85 FPS, WHILE PS AT 1.10 FPS.

| Baseline | 1 | 3 | 5 | 7 | 9 | 11 | 13 | 15 | 17 | 19 | 21 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Truck | | | | | | | | | | | |
| PS | 1.200 | 0.701 | 0.550 | 0.498 | 0.483 | 0.480 | 0.489 | 0.503 | 0.522 | 0.541 | 0.554 |
| SBPS | 1.150 | 0.640 | 0.476 | 0.418 | 0.395 | 0.382 | 0.378 | 0.383 | 0.392 | 0.405 | 0.413 |
| Ignatius | | | | | | | | | | | |
| PS | 0.531 | 0.311 | 0.233 | 0.198 | 0.184 | 0.175 | 0.173 | 0.174 | 0.175 | 0.177 | 0.179 |
| SBPS | 0.575 | 0.339 | 0.252 | 0.217 | 0.199 | 0.184 | 0.179 | 0.180 | 0.180 | 0.180 | 0.179 |
| Church | | | | | | | | | | | |
| PS | 5.708 | 3.921 | 3.365 | 3.039 | 2.890 | 2.840 | 2.812 | 2.791 | 2.804 | 2.800 | 2.813 |
| SBPS | 5.915 | 3.962 | 3.280 | 2.926 | 2.752 | 2.620 | 2.603 | 2.526 | 2.521 | 2.503 | 2.472 |
| Caterpillar | | | | | | | | | | | |
| PS | 1.721 | 1.173 | 0.974 | 0.896 | 0.837 | 0.812 | 0.796 | 0.785 | 0.774 | 0.796 | 0.780 |
| SBPS | 1.727 | 1.177 | 0.972 | 0.877 | 0.800 | 0.780 | 0.750 | 0.731 | 0.723 | 0.744 | 0.711 |
| Barn | | | | | | | | | | | |
| PS | 4.160 | 3.573 | 2.930 | 2.705 | 2.613 | 2.596 | 2.497 | 2.386 | 2.376 | 2.344 | 2.335 |
| SBPS | 4.087 | 3.515 | 2.806 | 2.589 | 2.424 | 2.318 | 2.154 | 2.004 | 2.014 | 1.943 | 1.982 |
| Meetingroom | | | | | | | | | | | |
| PS | 3.812 | 2.796 | 2.380 | 2.165 | 2.056 | 2.000 | 1.965 | 1.907 | 1.916 | 1.907 | 1.872 |
| SBPS | 3.827 | 2.634 | 2.236 | 1.936 | 1.875 | 1.795 | 1.742 | 1.698 | 1.712 | 1.723 | 1.674 |

TABLE II

FRACTION OF PIXELS WITH RELATIVE DEPTH ERROR UNDER 5% (FI) FOR THE BINOCULAR CONFIGURATION OF TABLE I. LARGER VALUES ARE BETTER HERE.

| Baseline | 1 | 3 | 5 | 7 | 9 | 11 | 13 | 15 | 17 | 19 | 21 |
|---|---|---|---|---|---|---|---|---|---|---|---|
| Truck | | | | | | | | | | | |
| PS | 0.103 | 0.259 | 0.370 | 0.436 | 0.472 | 0.493 | 0.500 | 0.501 | 0.498 | 0.493 | 0.489 |
| SBPS | 0.098 | 0.275 | 0.401 | 0.477 | 0.516 | 0.543 | 0.554 | 0.557 | 0.553 | 0.548 | 0.545 |
| Ignatius | | | | | | | | | | | |
| PS | 0.125 | 0.287 | 0.417 | 0.502 | 0.551 | 0.585 | 0.597 | 0.602 | 0.608 | 0.609 | 0.612 |
| SBPS | 0.088 | 0.254 | 0.401 | 0.474 | 0.534 | 0.576 | 0.587 | 0.585 | 0.595 | 0.594 | 0.599 |
| Church | | | | | | | | | | | |
| PS | 0.047 | 0.100 | 0.134 | 0.171 | 0.202 | 0.215 | 0.230 | 0.238 | 0.245 | 0.259 | 0.261 |
| SBPS | 0.037 | 0.089 | 0.127 | 0.168 | 0.204 | 0.224 | 0.237 | 0.245 | 0.256 | 0.268 | 0.274 |
| Caterpillar | | | | | | | | | | | |
| PS | 0.078 | 0.184 | 0.267 | 0.316 | 0.353 | 0.375 | 0.387 | 0.399 | 0.408 | 0.408 | 0.412 |
| SBPS | 0.065 | 0.180 | 0.268 | 0.319 | 0.363 | 0.385 | 0.394 | 0.409 | 0.413 | 0.413 | 0.418 |
| Barn | | | | | | | | | | | |
| PS | 0.060 | 0.117 | 0.166 | 0.202 | 0.228 | 0.244 | 0.263 | 0.282 | 0.283 | 0.290 | 0.284 |
| SBPS | 0.050 | 0.109 | 0.161 | 0.196 | 0.231 | 0.247 | 0.281 | 0.300 | 0.305 | 0.316 | 0.311 |
| Meetingroom | | | | | | | | | | | |
| PS | 0.037 | 0.084 | 0.124 | 0.150 | 0.176 | 0.187 | 0.198 | 0.212 | 0.216 | 0.225 | 0.231 |
| SBPS | 0.034 | 0.086 | 0.119 | 0.148 | 0.168 | 0.183 | 0.194 | 0.210 | 0.216 | 0.215 | 0.229 |

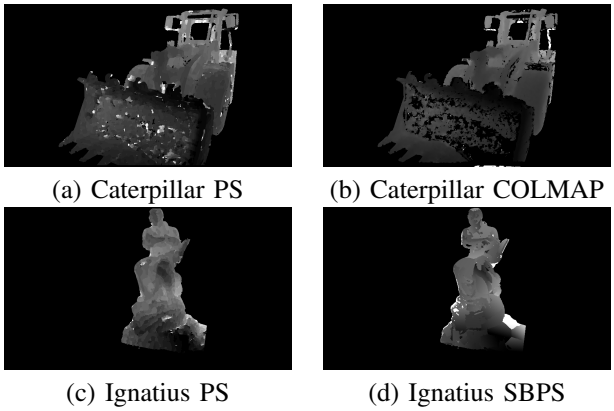
**Depth Density in SBPS.** Table III shows MAE and FI for SBPS, using 11×11 SAD on one target frame with the baseline set at 7, as we vary the percentage of pixels for which depth is estimated before superpixel fitting. The loss of accuracy is negligible even when depth is estimated for only 5% of all pixels, while, as expected, the speed of photoconsistency estimation and plane fitting increase as the density decreases. SBPS becomes slower than PS at high density because it is tailored for sparse depth estimates and the implementation does not take advantage of the regularity of fully dense depth maps.

**COLMAP and MVSNet.** We tested COLMAP [11] and MVSNet [12] on our data to gain additional insights, even though both have been tailored for offline 3D reconstruction. Results can be seen in Table IV and Fig. 4.

After verifying that the triangulation angles at the far range are large enough, we provided camera poses, depth range and a total of *three* images (the reference and the seven frames before and after it) as input to COLMAP to estimate the depth map of the reference image. Since COLMAP does not strictly respect the depth range, we clipped the output depth values to enforce it. As expected, COLMAP does not produce depths for textureless regions, or pixels of low confidence in general. On the other hand, it produces depths with lower MAE for reasonably textured scenes, like Ignatius and Caterpillar. COLMAP takes approximately 67 seconds to estimate depth using a total of three input images. Increasing the baseline from 7 did not lead to improved accuracy – increasing the number of images does, but processing times are longer.

The pre-trained model of MVSNet was applied on the same image triplets (with a baseline of 7) as COLMAP

and depths were clipped to the specified range. MVSNet takes 17.4 seconds for each output, after the model has been loaded. It achieves the best FI on the harder scenes and overall, but it does not perform as well in terms of MAE. We hypothesize that this is due to the downsampling that takes place in the network. MVSNet is faster than COLMAP, but much slower than PS and SBPS.



(a) Caterpillar PS (b) Caterpillar COLMAP

(c) Ignatius PS (d) Ignatius SBPS

Fig. 4. Depth maps of Caterpillar and Ignatius. PS and SBPS results are binocular, while COLMAP results are trinocular.

Table IV shows statistics of representative runs of all algorithms. PS, SBPS, PS-NCC and SBPS-NCC use one target view, while COLMAP and MVSNet use two target views. PS-7, SBPS-7, and COLMAP-7 use 7 views: the reference and target views 7, 14 and 21 frames away in both directions. All SBPS results refer to a 5% sampling density. Averages are taken over the six scenes and fps values are divided by 15 before computing the harmonic mean (HAR). Binocular SBPS provides the best balance between speed and accuracy. The vectorized implementation of NCC makes PS-NCC also competitive. COLMAP performs well in terms of FI, but is slow. MVSNet is faster but less accurate.

A broad conclusion from our experiments is that the baseline is indeed the most critical factor that affects 3D reconstruction. Other factors, such the matching function or the window size had small impact on this dataset.

## VI. Conclusions

We have created a benchmark that we hope will aid progress in video-based 3D reconstruction. This is an area of geometric vision with crucial applications, such as in autonomous driving, UAV perception and augmented reality, that suffers from the lack of data with ground truth. We are optimistic that our dataset can accelerate development by providing the means to objectively measure performance, assess algorithmic design choices and also serve as a reference for comparing different algorithms. We plan to release the data and the ground truth of the training set and create a website that will accept and score depth maps submitted by the research community and host the leaderboard. The source code of PS and SBPS will also be released.

TABLE III

Effects of varying the density of pixels with depth estimates in SBPS on mean absolute error (MAE) and fraction of inliers (FI)

| | Density | MAE | FI | fps |
|---|---|---|---|---|
| Truck | 100% | 0.410 | 0.491 | 0.137 |
| | 50% | 0.412 | 0.489 | 0.266 |
| | 10% | 0.411 | 0.483 | 1.181 |
| | 5% | 0.417 | 0.478 | 2.052 |
| Caterpillar | 100% | 0.892 | 0.322 | 0.137 |
| | 50% | 0.888 | 0.322 | 0.267 |
| | 10% | 0.886 | 0.320 | 1.156 |
| | 5% | 0.878 | 0.320 | 1.731 |

TABLE IV

Statistics of representative runs of all algorithms. See text for details.

| | Algorithm | MAE | FI | fps | HAR |
|---|---|---|---|---|---|
| Truck | PS | 0.498 | 0.436 | 1.120 | 0.127 |
| | SBPS | 0.417 | 0.477 | 1.902 | 0.200 |
| | PS-7 | 0.438 | 0.579 | 0.174 | 0.023 |
| | SBPS-7 | 0.435 | 0.594 | 0.481 | 0.061 |
| | PS-NCC | 0.460 | 0.497 | 1.157 | 0.134 |
| | SBPS-NCC | 0.348 | 0.558 | 1.037 | 0.123 |
| | COLMAP | 0.752 | 0.248 | 0.014 | 0.002 |
| | COLMAP-7 | 0.682 | 0.306 | 0.006 | 0.001 |
| | MVSNet | 1.811 | 0.288 | 0.057 | 0.008 |
| Ignatius | PS | 0.198 | 0.502 | 1.128 | 0.131 |
| | SBPS | 0.215 | 0.474 | 2.026 | 0.210 |
| | PS-7 | 0.146 | 0.672 | 0.180 | 0.024 |
| | SBPS-7 | 0.172 | 0.640 | 0.516 | 0.065 |
| | PS-NCC | 0.174 | 0.570 | 1.257 | 0.146 |
| | SBPS-NCC | 0.188 | 0.548 | 1.033 | 0.122 |
| | COLMAP | 0.176 | 0.585 | 0.014 | 0.002 |
| | COLMAP-7 | 0.155 | 0.632 | 0.006 | 0.001 |
| | MVSNet | 0.614 | 0.140 | 0.057 | 0.007 |
| Church | PS | 3.039 | 0.171 | 1.109 | 0.103 |
| | SBPS | 2.933 | 0.168 | 1.881 | 0.144 |
| | PS-7 | 2.277 | 0.299 | 0.182 | 0.023 |
| | SBPS-7 | 2.208 | 0.294 | 0.531 | 0.063 |
| | PS-NCC | 2.456 | 0.222 | 1.219 | 0.119 |
| | SBPS-NCC | 2.930 | 0.206 | 1.021 | 0.102 |
| | COLMAP | 3.758 | 0.177 | 0.015 | 0.002 |
| | COLMAP-7 | 2.982 | 0.214 | 0.006 | 0.001 |
| | MVSNet | 2.985 | 0.285 | 0.057 | 0.008 |
| Caterpillar | PS | 0.896 | 0.316 | 1.122 | 0.121 |
| | SBPS | 0.878 | 0.319 | 1.913 | 0.182 |
| | PS-7 | 0.553 | 0.502 | 0.175 | 0.023 |
| | SBPS-7 | 0.592 | 0.481 | 0.483 | 0.060 |
| | PS-NCC | 0.759 | 0.392 | 1.222 | 0.135 |
| | SBPS-NCC | 0.737 | 0.402 | 1.020 | 0.116 |
| | COLMAP | 0.584 | 0.446 | 0.015 | 0.002 |
| | COLMAP-7 | 0.484 | 0.500 | 0.006 | 0.001 |
| | MVSNet | 1.748 | 0.300 | 0.057 | 0.008 |
| Barn | PS | 2.705 | 0.202 | 1.014 | 0.101 |
| | SBPS | 2.521 | 0.196 | 1.603 | 0.138 |
| | PS-7 | 1.845 | 0.347 | 0.174 | 0.022 |
| | SBPS-7 | 1.802 | 0.347 | 0.460 | 0.056 |
| | PS-NCC | 2.674 | 0.225 | 1.240 | 0.121 |
| | SBPS-NCC | 2.412 | 0.228 | 1.011 | 0.104 |
| | COLMAP | 2.156 | 0.239 | 0.015 | 0.002 |
| | COLMAP-7 | 1.935 | 0.282 | 0.006 | 0.001 |
| | MVSNet | 3.199 | 0.331 | 0.057 | 0.008 |
| Meetingroom | PS | 2.165 | 0.150 | 1.133 | 0.100 |
| | SBPS | 1.937 | 0.148 | 1.962 | 0.139 |
| | PS-7 | 1.746 | 0.262 | 0.176 | 0.022 |
| | SBPS-7 | 1.627 | 0.239 | 0.494 | 0.058 |
| | PS-NCC | 2.087 | 0.173 | 1.232 | 0.111 |
| | SBPS-NCC | 1.852 | 0.172 | 1.035 | 0.098 |
| | COLMAP | 2.754 | 0.180 | 0.015 | 0.002 |
| | COLMAP-7 | 2.449 | 0.214 | 0.001 | 0.000 |
| | MVSNet | 1.579 | 0.239 | 0.057 | 0.007 |
| AVERAGE | PS | | 0.296 | 1.104 | 0.114 |
| | SBPS | | 0.297 | **1.881** | **0.169** |
| | PS-7 | | **0.444** | 0.177 | 0.023 |
| | SBPS-7 | | 0.433 | 0.494 | 0.061 |
| | PS-NCC | | 0.346 | 1.221 | 0.128 |
| | SBPS-NCC | | 0.352 | 1.026 | 0.111 |
| | COLMAP | | 0.313 | 0.015 | 0.002 |
| | COLMAP-7 | | 0.358 | 0.005 | 0.001 |
| | MVSNet | | 0.264 | 0.057 | 0.007 |

**Acknowledgment.** The authors are grateful to Johannes Schönberger for his help with COLMAP.

## REFERENCES


[1] 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, pp. 519–528.

[2] H. Aanæs, R. R. Jensen, G. Vogiatzis, E. Tola, and A. B. Dahl, "Large-scale data for multiple-view stereopsis," *IJCV*, 2016.

[3] T. Schöps, J. L. Schonberger, S. Galliani, T. Sattler, K. Schindler, M. Pollefeys, and A. Geiger, "A multi-view stereo benchmark with high-resolution images and multi-camera videos," in *CVPR*, 2017.

[4] A. Knapitsch, J. Park, Q.-Y. Zhou, and V. Koltun, "Tanks and Temples: Benchmarking Large-Scale Scene Reconstruction," *ACM Transactions on Graphics*, vol. 36, no. 4, 2017.

[5] M. Burri, J. Nikolic, P. Gohl, T. Schneider, J. Rehder, S. Omari, M. W. Achtelik, and R. Siegwart, "The EuRoC micro aerial vehicle datasets," *The International Journal of Robotics Research*, 2016.

[6] "Waymo open dataset: An autonomous driving dataset," 2019.

[7] M.-F. Chang, J. W. Lambert, P. Sangkloy, J. Singh, S. Bak, A. Hartnett, D. Wang, P. Carr, S. Lucey, D. Ramanan, and J. Hays, "Argoverse: 3d tracking and forecasting with rich maps," in *CVPR*, 2019.

[8] X. Huang, X. Cheng, Q. Geng, B. Cao, D. Zhou, P. Wang, Y. Lin, and R. Yang, "The Apolloscape Dataset for Autonomous Driving," in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition Workshops*, 2018, pp. 954–960.

[9] R. Kesten, M. Usman, J. Houston, T. Pandya, K. Nadhamuni, A. Ferreira, M. Yuan, B. Low, A. Jain, P. Ondruska, S. Omari, S. Shah, A. Kulkarni, A. Kazakova, C. Tao, L. Platinsky, W. Jiang, and V. Shet, "Lyft level 5 av dataset 2019," urlhttps://level5.lyft.com/dataset/, 2019.

[10] D. Gallup, J.-M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys, "Real-time plane-sweeping stereo with multiple sweeping directions," in *CVPR*, 2007.

[11] J. L. Schönberger, E. Zheng, J.-M. Frahm, and M. Pollefeys, "Pixelwise view selection for unstructured multi-view stereo," in *ECCV*, 2016, pp. 501–518.

[12] Y. Yao, Z. Luo, S. Li, T. Fang, and L. Quan, "MVSNet: depth inference for unstructured multi-view stereo," in *ECCV*, 2018.

[13] Y. Furukawa and C. Hernández, "Multi-view stereo: A tutorial," *Foundations and Trends in Computer Graphics and Vision*, 2015.

[14] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen, "On benchmarking camera calibration and multi-view stereo for high resolution imagery," in *CVPR*, 2008.

[15] M. Pollefeys, D. Nistér, J. M. Frahm, A. Akbarzadeh, P. Mordohai, B. Clipp, C. Engels, D. Gallup, S. J. Kim, P. Merrell, C. Salmi, S. Sinha, B. Talton, L. Wang, Q. Yang, H. Stewénius, R. Yang, G. Welch, and H. Towles, "Detailed real-time urban 3D reconstruction from video," *IJCV*, vol. 78, no. 2-3, pp. 143–167, 2008.

[16] J. Stühmer, S. Gumhold, and D. Cremers, "Real-time dense geometry from a handheld camera," in *DAGM*, 2010, pp. 11–20.

[17] R. Newcombe and A. Davison, "DTAM: Dense tracking and mapping in real-time," in *ICCV*, 2011.

[18] A. Wendel, M. Maurer, G. Graber, T. Pock, and H. Bischof, "Dense reconstruction on-the-fly," in *CVPR*, 2012, pp. 1450–1457.

[19] V. Pradeep, C. Rhemann, S. Izadi, C. Zach, M. Bleyer, and S. Bathiche, "Monofusion: Real-time 3d reconstruction of small scenes with a single web camera," in *International Symposium on Mixed and Augmented Reality (ISMAR)*, 2013, pp. 83–88.

[20] M. Pizzoli, C. Forster, and D. Scaramuzza, "Remode: Probabilistic, monocular dense reconstruction in real time," in *ICRA*, 2014, pp. 2609–2616.

[21] J. Zienkiewicz, A. Tsiotsios, A. Davison, and S. Leutenegger, "Monocular, real-time surface reconstruction using dynamic level of detail," in *International Conference on 3D Vision (3DV)*, 2016, pp. 37–46.

[22] T. Schöps, T. Sattler, C. Häne, and M. Pollefeys, "Large-scale outdoor 3D reconstruction on a mobile device," *CVIU*, vol. 157, pp. 151–166, 2017.

[23] L. Teixeira and M. Chli, "Real-time local 3D reconstruction for aerial inspection using superpixel expansion," in *ICRA*, 2017, pp. 4560–4567.

[24] M. Bleyer, C. Rhemann, and C. Rother, "PatchMatch stereo-stereo matching with slanted support windows." in *BMVC*, 2011.

[25] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz, "PMBP: Patchmatch belief propagation for correspondence field estimation," *IJCV*, vol. 110, no. 1, pp. 2–13, 2014.

[26] P. Heise, S. Klose, B. Jensen, and A. Knoll, "PM-Huber: PatchMatch with Huber regularization for stereo matching," in *ICCV*, 2013, pp. 2360–2367.

[27] C. Bailer, M. Finckh, and H. P. Lensch, "Scale robust multi view stereo," in *ECCV*, 2012, pp. 398–411.

[28] S. Shen, "Accurate multiple view 3D reconstruction using patch-based stereo for large-scale scenes," *IEEE Transactions on Image Processing*, vol. 22, no. 5, pp. 1901–1914, 2013.

[29] Y. Uh, Y. Matsushita, and H. Byun, "Efficient multiview stereo by random-search and propagation," in *International Conference on 3D Vision (3DV)*, 2014, pp. 393–400.

[30] J. Wei, B. Resch, and H. Lensch, "Multi-view depth map estimation with cross-view consistency," in *BMVC*, 2014.

[31] E. Zheng, E. Dunn, V. Jojic, and J.-M. Frahm, "Patchmatch based joint view selection and depthmap estimation," in *CVPR*, 2014, pp. 1510–1517.

[32] S. Galliani, K. Lasinger, and K. Schindler, "Massively parallel multiview stereopsis by surface normal diffusion," in *ICCV*, 2015, pp. 873–881.

[33] S. N. Sinha, D. Scharstein, and R. Szeliski, "Efficient high-resolution stereo matching using local plane sweeps," in *CVPR*, 2014, pp. 1582–1589.

[34] A. Geiger, M. Roser, and R. Urtasun, "Efficient large-scale stereo matching," in *ACCV*, 2010.

[35] C. LeGendre, K. Batsos, and P. Mordohai, "High-resolution stereo matching based on sampled photoconsistency computation," in *BMVC*, 2017.

[36] C. Häne, L. Heng, G. H. Lee, A. Sizov, and M. Pollefeys, "Real-time direct dense matching on fisheye images using plane-sweeping stereo," in *International Conference on 3D Vision (3DV)*, 2014, pp. 57–64.

[37] Y. Dai, Z. Zhu, Z. Rao, and B. Li, "MVS$^2$: Deep unsupervised multi-view stereo with multi-view symmetry," in *International Conference on 3D Vision (3DV)*, 2019.

[38] J. Flynn, I. Neulander, J. Philbin, and N. Snavely, "Deepstereo: Learning to predict new views from the world's imagery," in *CVPR*, 2016.

[39] W. Hartmann, S. Galliani, M. Havlena, L. Van Gool, and K. Schindler, "Learned multi-patch similarity," in *ICCV*, 2017.

[40] P.-H. Huang, K. Matzen, J. Kopf, N. Ahuja, and J.-B. Huang, "DeepMVS: learning multi-view stereopsis," in *CVPR*, 2018.

[41] S. Im, H.-G. Jeon, S. Lin, and I. S. Kweon, "DPSNet: end-to-end deep plane sweep stereo," in *ICLR*, 2019.

[42] V. Leroy, J.-S. Franco, and E. Boyer, "Shape reconstruction using volume sweeping and learned photoconsistency," in *ECCV*, 2018, pp. 781–796.

[43] K. Luo, T. Guan, L. Ju, H. Huang, and Y. Luo, "P-MVSNet: Learning patch-wise matching confidence aggregation for multi-view stereo," in *ICCV*, 2019, pp. 10 452–10 461.

[44] H. Ha, S. Im, J. Park, H.-G. Jeon, and I. So Kweon, "High-quality depth from uncalibrated small motion clip," in *CVPR*, 2016, pp. 5413–5421.

[45] F. Yu and D. Gallup, "3D reconstruction from accidental motion," in *CVPR*, 2014, pp. 3986–3993.

[46] J. L. Schonberger and J.-M. Frahm, "Structure-from-motion revisited," in *CVPR*, 2016, pp. 4104–4113.

[47] R. Mur-Artal, J. M. M. Montiel, and J. D. Tardós, "ORB-SLAM: a versatile and accurate monocular SLAM system," *IEEE Transactions on Robotics*, vol. 31, no. 5, pp. 1147–1163, 2015.

[48] A. Geiger, P. Lenz, C. Stiller, and R. Urtasun, "Vision meets robotics: The KITTI dataset," *International Journal of Robotics Research (IJRR)*, vol. 32, no. 11, pp. 1231–1237, 2013.

[49] R. Achanta, A. Shaji, K. Smith, A. Lucchi, P. Fua, and S. Susstrunk, "SLIC superpixels compared to state-of-the-art superpixel methods," *PAMI*, vol. 34, no. 11, pp. 2274–2282, 2012.

[50] C. Y. Ren, V. A. Prisacariu, and I. D. Reid, "gSLICr: SLIC superpixels at over 250Hz," *ArXiv e-prints*, Sept. 2015.