SyB3R: A Realistic Synthetic Benchmark for 3D Reconstruction from Images

Andreas Ley(✉), Ronny Hänsch, and Olaf Hellwich

Computer Vision and Remote Sensing Group,
Technische Universität Berlin, Berlin, Germany
{andreas.ley,r.haensch,olaf.hellwich}@tu-berlin.de

Abstract. Benchmark datasets are the foundation of experimental evaluation in almost all vision problems. In the context of 3D reconstruction these datasets are rather difficult to produce. The field is mainly divided into datasets created from real photos with difficult experimental setups and simple synthetic datasets which are easy to produce, but lack many of the real world characteristics. In this work, we seek to find a middle ground by introducing a framework for the synthetic creation of realistic datasets and their ground truths. We show the benefits of such a purely synthetic approach over real world datasets and discuss its limitations.

1 Introduction and Related Work

The reconstruction of digital 3D models from images includes various tasks ranging from camera calibration, over the determination of camera positions (structure from motion) and dense reconstruction, to surface generation and interpretation (e.g. segmentation). Over the last years, a still rising number of algorithms have been proposed that are able to obtain high-quality 3D reconstructions in several application scenarios, including those where other approaches are not easily applicable. The state of the art of this field is still improving rapidly. An overview about recent advances in structure from motion methods can be found in [1], while [2–4] offer reviews of multi-view stereo algorithms.

The need to objectively compare such algorithms and to investigate their intrinsic properties has led to the proposal of many benchmark datasets, which provide reference data (i.e. measured by other sensors) or ground truth (based on synthetic models). Both types of datasets have complementary benefits and limitations. Datasets that are based on real measurements have the advantage that all the effects that can occur during data acquisition are (at least potentially) included as they actually happen during the acquisition. This property of real datasets is of course only theoretical, since the concrete, practical experimental setup is limiting the effects that can be covered. These datasets mostly contain a few example and often simplified scenarios, where images are obtained under

Electronic supplementary material The online version of this chapter (doi:10.1007/978-3-319-46478-7_15) contains supplementary material, which is available to authorized users.

B. Leibe et al. (Eds.): ECCV 2016, Part VII, LNCS 9911, pp. 236–251, 2016.
DOI: 10.1007/978-3-319-46478-7_15

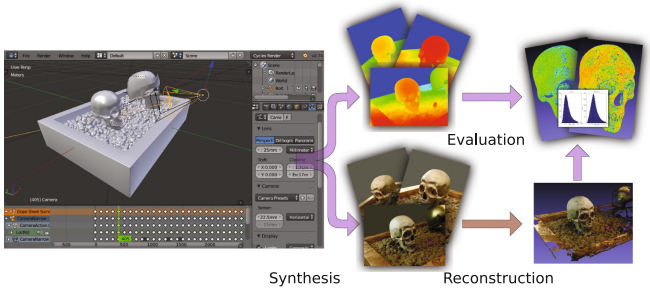

Fig. 1. We present a framework for the synthetic generation of realistic 3D reconstruction datasets with ground truth data which allows the evaluation of 3D reconstruction methods in fully controlled and possibly non-standard application scenarios.

fixed conditions (e.g. same lighting, same camera, a certain baseline, etc.). Furthermore, these datasets cannot provide ground truth but only reference data, which is acquired by a sensor (mostly structured light or laser scanning) that is assumed to have an accuracy superior to the system under investigation.

In [5] a database of images with reference data based on structured light is provided for the computation of a dense depth map from a pair of images. The last extension of this database provides 33 datasets of high-resolution images and subpixel-accurate reference data [6]. Several multi-view stereo reconstruction algorithms are evaluated in [3] on the basis of images and laser scans of a plaster reproduction of two objects. A spherical gantry is used to move the camera to predefined positions on a hemisphere. To remove shadows casted by the gantry the hemisphere had to be covered two times with different configurations leading to 790 views in total. The images are corrected for radial distortion. Calibration accuracy is in the order of a pixel corresponding to roughly 0.25 mm on the object. Since the measurement of completeness and correctness of the estimated mesh with respect to the reference mesh is problematic if either contains holes, a hole-filled version of the reference mesh is used and estimated points close to hole-filled regions are discarded. In [7] a robotic arm is used to position the camera and a structured light sensor. Camera positions are not known due to low position accuracy of the robot (despite high repeatability) but are estimated based on a calibration object that is included in the scene. One of the probably best known benchmarks is introduced in [8] and provides six different datasets with 8–30 images mainly showing different architectural objects (such as facades, fountains, building entrances). The images have been corrected for radial distortion and reference data is provided for camera calibration and depth measured by a laser scanner with an accuracy of less than 1 cm. The authors stress the role of reference data within a benchmark and report the variance of the laser scans. In [9] an autonomous driving platform is used to compile challenging real-world benchmarks for stereo, optical flow, visual odometry, 3D object detection, and 3D tracking. The data consists of nearly 200 scenes of a static environment and was extended by another 400 scenes in 2015 [10]. Reference data is produced by

a laser scanner and GPS localization. Their results illustrate the disadvantage of data captured in controlled environments by reporting below average performance of methods that achieve high rankings on other established benchmarks. Despite the advantages of real datasets for benchmarking, the examples above illustrate their downsides: The requirements on hardware as well as software are tremendous and any evaluation is limited to the objects and data acquisition circumstances covered by a specific benchmark as well as to methods which are at least an order of magnitude less accurate than the provided reference data.

Synthetic benchmarks are complementary to real data: Since image acquisition only consists of rendering images on a computer, it is fast, cheap, and allows full control of scene content and properties (e.g. lighting) as well as changing camera parameters or image characteristics. Since only software and data are needed, the whole process of image production can be shared (instead of only image and reference data), which increases the repeatability of the experiment by others. Instead of reference data with measurement errors by itself, the actual ground truth is known. The disadvantage is that it is often unclear how realistic the produced data is and whether the evaluated methods react to real data in the same way as to synthetic data. Furthermore, the creation of the synthetic 3D models is often complex and requires (besides a good understanding of the properties of cameras) a certain artistic skill. That is why often either simplified 3D scenes are used or models that had been created for other purposes.

One of the more realistic synthetic datasets ("*Tsukuba*") is proposed in [11,12]. It provides 1800 stereo image pairs of an office scene under four different illuminations along with the corresponding true disparity maps. The scene has been created and rendered photo-realistically by use of Pixologic ZBrush and Autodesk Maya. A synthetic dataset for benchmarking optical flow algorithms is proposed in [13]. It is based on the open source 3D animated short film "*Sintel*" and rendered with Blender using its internal ray tracer. The ground truth includes camera intrinsics and extrinsics as well as the true depth for each pixel. The authors prepare several variations of this dataset for different tasks including depth, camera motion, stereo, and disparity estimation.

A common disadvantage of such benchmarks is that they are restricted to only one scene, although this scene is composed of complex objects/subscenes and can be rendered under various conditions. While the *Tsukuba* dataset is designed for benchmarking computer vision methods, the *Sintel* dataset is originally intended to be visually pleasing. Many parts of the scene that seem to contain 3D structures are actually flat. The visible structure exists only in their texture and normal maps but is not existent in the actual 3D mesh (and consequently also not in the ground truth depth maps). The image synthesis stops at the image formation process of a pinhole camera, while other effects of a real camera (such as tone mapping, noise, motion blur) are neglected or tweaked for artistic purposes.

This paper proposes an evaluation pipeline (see Fig. 1) that stands between real benchmarks on the one hand with all the challenges of real data but high costs for reference data of limited accuracy and on the other hand synthetic

datasets that provide accurate ground truth but only simplified 3D models and image formation. The proposed framework uses realistically rendered images (as opposed to artistic/stylized images as in *Sintel*), where "realistic" means not only photo-realistic in the sense that images look real but also in a more "physical" sense. We use Blender with path tracing instead of simple ray tracing to be able to simulate more complex light-surface interactions. Many real world effects of image acquisition (such as motion blur and noise) are simulated during image rendering or post-processing. All camera parameters are known as well as the 3D structure of the scene.

Our ultimate goal is not to replace but complement benchmarks based on real data by novel datasets i.e. scenes that have been rendered with varying properties. This not only allows to evaluate and compare different methods, but also to investigate the influence of camera or scene properties on the reconstruction, to prototype, design, and test experiments before realizing them in the real world, and to generate training data for learning-based approaches. These experiments open the possibility to analyze when and why methods fail and consequently to suggest potentials for future work. We provide the means to produce datasets in addition to the datasets introduced in Sect. 3.

The contributions of this work are four-fold: We provide (1) an automatic synthesis framework with full control about the scene and the image formation, (2) several datasets with a variety of challenging characteristics, (3) results of a few example experiments illustrating the benefits of synthetic but realistic benchmarks, and (4) a flexible and open-source C++ implementation of the proposed framework. Our datasets and evaluation framework are publicly available [14] and open to the general community.

It should be stressed, that the focus of this paper is the framework itself, i.e. explaining its general workflow, discussing its potentials and limitations, as well as illustrating possible applications and experiments. The experiments in Sect. 3 serve the only purpose to illustrate the general potential of SyB3R and are not meant as a thorough study of corresponding methods. The publication of the methodology and software of SyB3R in its current state will help to steer future work into directions most needed by the scientific community and to include scene and image characteristics, error measurements, and datasets that are specifically designed to answer currently open questions.

2 SyB3R

The image formation process in digital cameras is more complicated than a simple projection plus digitization and quantization. After a complicated setup of lens elements the light hits the image sensor. The measurements of the image sensor are subject to noise, some from the sensor itself, and some from the random but quantized number of photons that comprise a certain amount of light. Each sensor element (pixel) measures only one color channel which is enforced by color filters that block photons of other wavelengths. This has two crucial consequences: First, color information has to be locally shared and interpolated.

Second, a change of the color model has to be performed, since the color filters are not necessarily focused at the RGB primary wavelengths of the sRGB model. The mapping of radiometric intensities to sRGB values is nonlinear and camera dependent. Finally, the JPG compression adds additional image artifacts. A more in-depth discussion on camera models is beyond the scope of this paper but can e.g. be found in [15,16]. Scene and camera properties lead to many effects such as specularity and reflections, image noise, blur caused by camera and object motion, chromatic aberrations, radial distortions, depth of field, etc. which are often ignored or insufficiently modelled in other synthetic datasets.

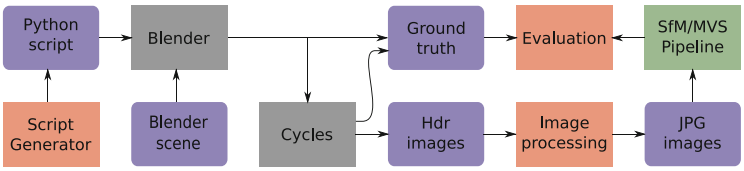

Fig. 2. Pipeline overview.

This section presents the framework of our Synthetic Benchmark for 3D Reconstruction (SyB3R), which is supposed to be a step towards closing the gap between existing real and synthetic datasets. In its current state, it is not a traditional benchmark per se in the sense that we provide a wide range of datasets accompanied with ground truth and perform a thorough evaluation of existing methods. Instead, SyB3R is the implementation of a modular framework to create such datasets. We strive for simulating the above mentioned real-world effects as realistically as possible while remaining flexible for future improvements as well as easy to use. To this end, we have to limit the image synthesis to a simplified model that does not encompass the entire physical process. We split the image generation into two parts: The actual rendering that projects the scene into a 2D image (Sect. 2.1) and a post-processing that implements remaining effects in image space (Sect. 2.2). An overview of SyB3R is shown in Fig. 2, while the following subsections explain the individual steps in more detail. All steps are implemented in a highly modular manner which allows to toggle individual modules on/off, change their relative order, or exchange them with different versions to create images optimized for specific purposes.

2.1 Image Rendering

Similar to [13] we use Blender to compose the virtual scene. The primary benefit of Blender over other alternatives (such as Autodesk Maya used by [11,12]) is that Blender is open source and can be acquired and used free of charge. This is of vital importance since we wish to release not only the final datasets, but also the software and tools that created them. Blender has an extensive animation and scripting system, allowing virtually every property to be animated or controlled

via Python scripts. This allows our framework to use small scripts which control the rendering process and perform automated modifications of key parameters of the scene. Another benefit is the infrastructure and community that comes with such a popular tool providing for example high-quality models under Creative Commons licenses (e.g. at [17]). There exist even render farms for Blender, where rendering time can be rented if sufficient compute power is locally unavailable.

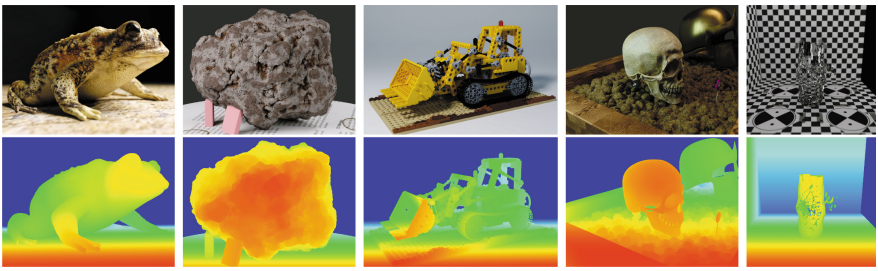

Fig. 3. Example images from our synthetic datasets. Top: Rendered color images. Bottom: Ground truth depth. (Color figure online)

For the image rendering we use *Cycles*, a Monte-Carlo path tracer that accurately simulates the propagation of light through the scene including refraction, reflection, and caustics. *Cycles* is distributed as part of Blender and has backends for CPUs and Cuda-GPUs. The produced images are stored as HDR images to retain the full floating-point precision of all intensity values for further processing. All scene properties (e.g. lighting and surface texture), object motion and large camera motion, as well as camera properties such as focal length, principal point, resolution, depth of field (DoF), and field of view are handled during the rendering process by *Cycles* unless they can be implemented as a post-processing step. For example, DoF is not added as a post-processing effect but is implemented in *Cycles* by offsetting the origin of the view rays. The view rays don't originate from a single focal point but from an area/volume that is shaped by the aperture (bokeh) and, if enabled, camera motion.

Even though cameras usually do not capture directly the three primary colors of the sRGB model, we render those radiometric RGB intensities since it allows the reuse of community models and textures.

2.2 Image Post-Processing

While *Cycles* provides some interesting image formation effects such as DoF, other aspects of the image formation process have to be realized as post-processing steps. We implement the image post-processing as a chain of individual modules (see Fig. 4) that can be exchanged and combined in different ways.

Fig. 4. Overview of the provided post-processing chain.

Camera Rotation Motion Blur. Since object motion blur and depth of field are affected by lighting and the 3D structure of the scene, we apply them during the rendering process. For short exposure times, however, camera motion blur stems from small rotations of the camera. This can be implemented as a post-processing effect which allows experimentation with different exposure times without having to rerender the images.

Camera motion blur is applied in image space on the basis of the HDR images. Instead of using hand-crafted blurring kernels, we took images of bright dots and measured the length of the glow trails in these images. The final blur kernel is a linear blur with random orientation, where the length of the blur is drawn from a gamma distribution that was fitted to the measured image blur (see Fig. 3 in the supplemental material).

Radial Distortion and Chromatic Aberration. To simulate the effects of radial distortion, we provide a post-processing step that resamples the image according to the commonly employed polynomial distortion model

$$\mathbf{x} = \mathbf{c} + \mathbf{d} \cdot \left(1 + |\mathbf{d}|^2 \cdot \kappa_2 + |\mathbf{d}|^3 \cdot \kappa_3 + |\mathbf{d}|^4 \cdot \kappa_4\right) \text{ with } \mathbf{d} = \mathbf{y} - \mathbf{c} \qquad (1)$$

where $\mathbf{x}$ and $\mathbf{y}$ are the source and destination pixel coordinates, respectively, $\mathbf{c}$ is the projection center, and $\kappa_{\{2,3,4\}}$ are polynomial coefficients that can be estimated by SFM as part of the internal calibration. Chromatic aberration is simulated by using separate sets of $\kappa_{\{2,3,4\}}$ coefficients for each color channel.

Automatic Exposure Control. Most cameras automatically adapt the exposure of the sensor to make full use of the limited numerical range of the final image. We implement this by scaling the color channels based on the average brightness of the central image region. While in reality the exposure control has an effect on exposure time, aperture size, and artificial amplification, we currently do not adapt the strength of motion blur, DoF, or sensor noise.

Sensor Noise. In reality the camera signal is corrupted with noise at the very beginning of the image formation process, but gets transformed by subsequent processing steps (such as color interpolation due to the Bayer pattern and changing the color model). The result is intensity-dependent noise, that is correlated spatially and across color channels. The corresponding relationship between signal and noise is highly complex and very different from the often assumed iid additive Gaussian noise [15]. This is illustrated in Fig. 5, which shows real noise

(on the left) in an image obtained with ISO 1600 as well as synthetic Gaussian iid noise (on the right), which is added to an image where real noise was reduced beforehand by averaging 30 pictures. The Gaussian noise has the same variance as noise that was found in test images of a medium gray tone.

There are two principle ways to derive more realistic noise: Modeling it through a full reproduction of the image formation process including demosaicing and color matrix multiplication or as a post-processing step by fitting the statistics of the rendered image to those of a real camera. The first approach has the advantage of providing better control about potential effects during the image formation process. However, many educated guesses concerning the choice of demosaicing filter and absence or nature of camera-internal denoising filters have to be made [16]. The latter approach, while being considerably simpler, allows to fit the image data closer to that of a specific real camera. The current version of SyB3R follows the second approach, while the first method has been deferred to future work as an additional module.

We acquired several test images of various colors and intensities with a Canon EOS 400D. We model the variance $\boldsymbol{\sigma}_{r,g,b}$ in the three color channels in linear space as a function of the color intensities **rgb** in linear space such that $\boldsymbol{\sigma}_{r,g,b} = \mathbf{A} \cdot \mathbf{rgb} + \mathbf{b}$. We assume a linear relationship between variance and intensity via **A**, since most of the image noise is shot noise from the Poisson distributed photon count. Intensity-independent noise is represented by the constant offset **b**. The matrix **A** and the vector **b** are fitted to the noise observed in the test images. The spatial correlation of the noise is modeled by blurring it by a Gaussian kernel with a standard deviation of 0.75 pixels which is normalized to retain the noise energy (i.e. the squared kernel values sum to one). The middle of Fig. 5 shows the resulting synthetic noise on the same input image as for the iid Gaussian noise. Although minor discrepancies remain between the real noise and our synthetic noise, it is clearly visible, that the similarity is much higher than for the iid Gaussian noise.

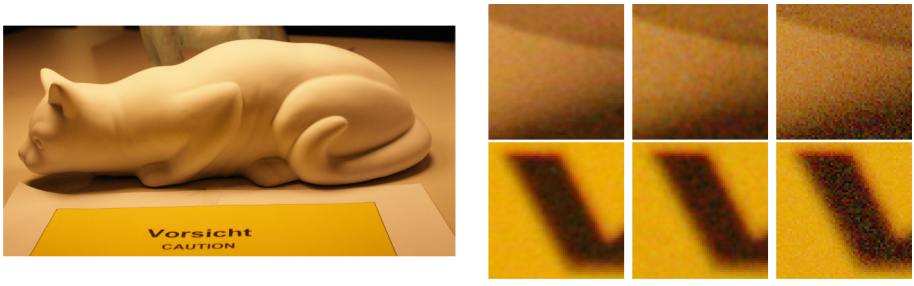

Fig. 5. Two different details of the image at the left taken with ISO 1600. From left to right: Original image; synthetic noise: our approach; synthetic noise: iid additive Gaussian with the noise variance equal to that of an ISO 1600 image of a 50 % gray tone.

Tonemapping and Compression. Instead of using predefined operators, the tone mapping is modeled after actual measurements. We measured the response curves (see Fig. 2 within the supplemental material) of a Canon EOS 400D as proposed in [18] which are used as a nonlinear mapping from radiometric intensities to 8-bit sRGB values.

The last processing step is the JPG compression of the obtained LDR images. The top of Fig. 3 shows some examples.

2.3 Ground Truth Generation

The extrinsic camera matrix is given by Blender as the world matrix of the camera object. The intrinsic camera matrix is computed using focal length, sensor size, principal point, and image size as given by Blender. The f-number of the camera is computed from focal length and aperture size but is only used for reference in the current work. All of these parameters are automatically extracted via a Python script and stored in an XML file.

The depth values (as distance to the focal point) are returned from *Cycles* in an additional "Z-buffer" pass in which motion blur and DoF are disabled. The second row of Fig. 3 shows some examples. Finally, a 3D mask of the object of interest is provided.

2.4 3D Reconstruction

A typical pipeline for 3D reconstruction from images consists of multiple parts such as calibration (to determine internal camera parameters), structure from motion (SfM, to determine external camera parameters), multi-view stereo (MVS, to obtain densely distributed samples on the surface), and eventually meshing (to determine the surface). Depending on the focus of their work, researchers evaluate either the whole pipeline from the start up to a certain point, or they concentrate on individual components. Currently the proposed benchmark evaluates SfM (i.e. camera position) and dense reconstruction. It allows (a) to run SfM only and evaluate the calibration, (b) to run MVS with the ground truth calibration only to evaluate the dense reconstruction, and (c) to run both to evaluate the influence of calibration errors by SfM.

2.5 Evaluation

To evaluate SfM, we transform the estimated camera positions into the coordinate system of the ground truth via a least squares Helmert transformation. The Euclidean distance between estimated and ground truth camera positions is computed as the error metric and is expressed in meters by supplying a metric scale (set inside Blender).

MVS methods are evaluated on the basis of the dense point clouds. The ground truth point cloud is synthesized (similar to laser/structured light scans in real datasets) by projecting each pixel back into 3D space using the ground

truth camera calibration and depth maps. If MVS was not run with the ground truth camera positions but the estimates of SfM, the estimated point cloud is transformed into the ground truth coordinate system by the Helmert transformation based on the camera correspondences.

Our performance metrics are similar to those applied by [3,7] on real datasets: The *precision* is the average distance of each estimated point to the closest ground truth point, while the *completeness* is the average distance of each ground truth point to the closest estimated point. Usually our datasets contain one object of interest and background. A reconstruction should neither be penalized for providing a good estimation of the object while ignoring the background (completeness), nor favored for providing a good reconstruction of the background while being less accurate on the object (precision). Based on the "object of interest"-mask rendered in *Cycles*, we label the ground truth points as foreground or background. The evaluation of completeness is then restricted to the foreground points. For the precision, only those estimated points are considered whose closest ground truth point is labeled as foreground.

Additionally we create ply-files with corresponding vertex qualities to visually represent completeness and precision. This provides a visual summary as well as information about the spatial distribution of errors over the point cloud.

3 Experiments

The focus of this paper is the proposal of SyB3R to synthesize images with their ground truth as well as illustrating its potential to evaluate SfM and MVS pipelines. An exhaustive comparison of modern 3D reconstruction methods as well as an in-depth study of the influence of all parameters are therefore both beyond the scope of this paper. Nevertheless, we provide several analysis examples on isolated parameters using VSFM [1,19,20] and a custom SfM pipeline as well as PMVS2 [21]. The following subsections showcase a few experiments, which would have been difficult to achieve with real benchmarks such as shift of principal point, change of surface texture, and different signal-to-noise ratios. Two more example experiments illustrating the influence of DoF and motion blur can be found in the supplemental material.

The datasets have been selected to contain objects being published with compatible licenses. The different objects have different surface properties such as sharp as well as smooth features, complex surface structures, strong concavities, and strongly as well as weakly textured regions. All images are rendered with 2000×1500 pixel resolution. All datasets are small in metric scale to showcase the influence of depth of field. A case where it would have been impossible to create real reference data is shown on the right side of Fig. 3 and described in the supplemental material.

3.1 General Performance Evaluation

This section illustrates an example of a general performance evaluation using SyB3R based on the *Toad* dataset [22]. An example image with its depth map is

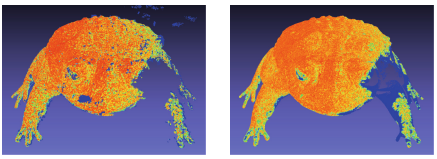

Fig. 6. Qualitative results for the *Toad* dataset after being processed by VSFM and PMVS2. Distances are color coded from zero (red) to 0.5 mm (blue). Left: Precision (distance to the closest ground truth point). Right: Completeness (distance to the closest estimated point). (Color figure online)

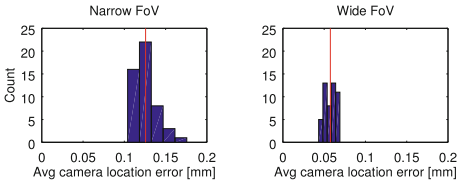

Fig. 7. Camera position error for narrow FoV (80 mm lens on 22.5 mm sensor) and wide FoV (25 mm lens on 22.5 mm sensor). Notice the significantly increased error for the narrow FoV.

shown on the left side of Fig. 3. The object has strong texture but a rather simple geometry with small depth complexity. Nevertheless, the bumps of the skin are modeled in 3D and not only simulated by texture and normal maps. A few parts of the surface (especially on the eyes) show specularity and are thus challenging for most reconstruction methods. Camera pose and position are estimated by VSFM, while the dense reconstruction is carried out with PMVS2.

The average positioning error of the cameras is 41.6 µm. The obtained values for completeness and precision are 118 µm and 144 µm, respectively. For reference, the toad is assumed to be about 11 cm long in its depicted pose. Figure 6 shows which points contributed to these errors by color-coding the distance of each point from zero (red) to 0.5 mm (blue). It should be noted, that regions with high specularity (e.g. eyes, fingers) have big errors or are completely missing and thus appear blue in the precision and completeness images, respectively.

3.2 Focal Length

In the following experiment we used a self-made 3D model depicting a geological hand sample on a turntable (in the spirit of [23]) shown in the second column of Fig. 3. The lighting is fixed to the camera, i.e. it moves with respect to the rock. A common empirical observation is that SfM usually benefits from a wide field of view (FoV). We rendered two datasets by using cameras with two different focal lengths. The cameras are placed at different distances to the object to equalize the object coverage and overlap of their images. Since SfM contains random elements (e.g. RANSAC), we run VSFM 50 times. The histograms in Fig. 7 show the resulting average camera errors. Indeed, the average positioning error increased from 0.0575 mm for a wide FoV to 0.1257 mm for a narrow FoV.

Note, that the dense reconstruction (PMVS2) by itself is virtually unaffected by the change in focal length. When run with the ground truth calibration data, we observe no significant change in precision (Narrow: 55.6 µm; Wide: 56.1 µm) nor in completeness (Narrow: 110 µm; Wide: 115 µm).

Table 1. Camera position errors for shifts in the principal point. For reference, the impact of the calibration errors on the precision of the dense reconstruction is also shown. All numbers in µm.

Shift amount	0 %	0.82 %	1.1 %
SfM	position errors		
VSFM	93	405	527
In house SfM	160	183	171
SfM + MVS	precision		
VSFM + PMVS2	259	383	378
In house SfM + PMVS2	242	241	244

Table 2. Precision and completeness for the full amount and quarter amount of texture and various noise amounts. All numbers in µm.

Texture	100 %		25 %	
Error	prec.	compl.	prec.	compl.
No noise	171	525	211	751
JPG 80	189	551	270	771
ISO 1600	212	751	398	14741

3.3 Principal Point

This experiment as well as those in the following Sect. 3.4 are carried out based on a self-composed dataset that consists of multiple community models: A skull [24], a helmet [25], and stone pebbles in a wooden box [26]. An example image is shown in the fourth column of Fig. 3. This dataset is inspired from the reconstruction of fossils and skeletons in natural history museums. These situations are often prone to insufficient lighting and weak texture.

This experiment investigates the influence of estimating the principal point on the position accuracy of SfM. We measured the principal point of a Canon EOS 400D to be offset from the image center by about 39 pixels (1 % of the image width). In [8] an offset of about 23 pixels (0.74 % of width) is reported, while [7] states an offset of about 57 pixels (3.56 % of width) for the left camera and about 25 pixels (1.56 % of width) for the right camera. Although a shift of 1 % of the image width seems realistic, several SfM pipelines (e.g. VSFM) do not allow an estimation of the principal point.

We rendered datasets with no shift, a shift of 16.4 pixels (0.82 % of width), and 21.3 pixels (1.1 % of width). The results in Table 1 show that the impact is severe, especially considering that only camera position and not rotation is evaluated: The positioning error increased from 0.093 mm to 0.527 mm for VSFM but stayed nearly constant for the custom pipeline which does estimate the principal point.

3.4 Surface Texture vs. JPG Compression or Noise

MVS methods require surface texture for an accurate reconstruction. This texture, however, can be weak or hidden by noise or compression artifacts [27]. In the following experiment, we modify the amount of texture on the skull and pitch the 100 % texture and the 25 % texture against JPG compression artifacts at 80 % quality as well as synthetic ISO 1600 sensor noise. The impact of JPG

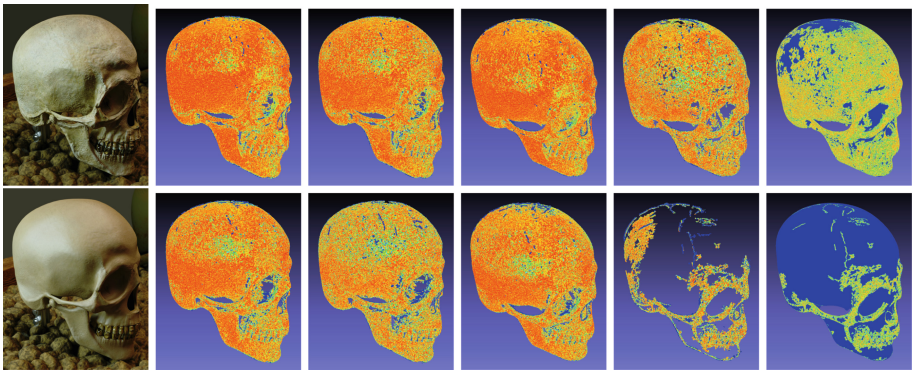

Fig. 8. Reduction of texture on a surface. Top row: Full texture. Bottom: 25 % texture. From left to right: Image at 100 % JPG quality, Precision at 100 % and 80 % JPG quality, respectively, Precision without noise, Precision with synthetic ISO 1600 noise, Completeness with synthetic ISO 1600 noise.

compression can be seen visually in the left half of Fig. 8. The distances are again color coded with blue corresponding to 1 mm. The right half of Fig. 8 shows the impact of ISO 1600 sensor noise with the same color coding. The average precision and completeness for each case is compiled into Table 2. It is noteworthy, that weak texture of its own is not truly a problem. Rather the relation between texture and noise, the "signal-to-noise-ratio", dictates the precision and completeness of the reconstruction [27]. The reduction of the texture to one quarter of its original strength results in a minor reduction of quality. Only in combination with strong compression or sensor noise does the quality decrease, for the latter quite significantly.

4 Conclusion and Future Work

This paper proposes SyB3R as a framework for the synthetic generation of realistic benchmarks for 3D reconstruction from images. We compose scenes from publicly available photo-realistic models and extend them with realistic effects of the image formation process of digital cameras. This approach not only allows to retrieve actual ground truth data. It also gives full control of all scene properties, including, but not restricted to, internal and external camera parameters, the light situation, object motion, and surface properties. Instead of only releasing a limited set of image sequences, we additionally make the whole processing chain publicly available. This enables researchers to quickly produce and prototype datasets for a wide range of applications. These datasets can include scenes with varying intrinsic properties such as surface texture (see Fig. 8) or datasets for non-standard computer vision problems (see right side of Fig. 3) for which an actual experimental setup would be too complicated or should only be attempted after an initial (synthetic) prototyping phase. Additionally to data and ground

truth generation, the framework provides an automatic qualitative and quantitative evaluation in a modular fashion to promote the direct inclusion into the test benches of research projects.

As common image acquisition artifacts, we leverage Blender and Cycle's abilities to model reflective and refractive surfaces, object motion blur, as well as depth of field and implement camera motion blur, radial distortion, chromatic aberrations, auto exposure, camera sensor noise, nonlinear tone mapping, and JPG compression as post processing. Highly situation and camera dependent effects such as camera motion blur, sensor noise, and tone mapping are modeled after empirical measurements. The corresponding tools are released as well, which allows to fit these models to new cameras. The automatic evaluation includes qualitative (visualization) as well as quantitative (position errors, precision, completeness) measurements.

The image synthesis is cheap compared to the creation of reference data for real datasets, but comes with a high computational load. Since a Monte-Carlo path tracer is used, a decrease of the inherent sampling noise must be paid for with quadratically increased rendering time. Render times can be multiple hours per image, depending on scene/material complexity and image resolution. However, distributing the rendering tasks to multiple computers is easily possible. Another point is that datasets have to be selected with a certain care. Models where fine geometry is only simulated by perturbing local surface normals would give bad results for reconstruction methods that use shading cues. Another common artistic trick are repeating textures, which will rise issues for most methods. The resolution of meshes and textures of all digital models is limited, which results in an upper limit for reasonable image sizes. Despite the advantages and potentials of synthetic data, real world datasets are still necessary to investigate which effects should be included into the synthesis and to confirm the findings based on synthetic benchmarks. However, for individual parameter tests and specialized, non-standard use-cases we believe synthetic datasets to be a valuable complement to real world datasets.

Future versions should extend the image formation in three major points: First, motion blur and sensor noise should be modelled as dependent on the exposure. Second, modelling the complete image formation process would allow a more sophisticated noise model. Third, depth of field can be modelled in image space with only minor quality degradation. This would lead to a smaller computational load since images do not have to be re-rendered. A next version of SyB3R will include more error measures capturing different aspects of the quality of 3D reconstructions. It is in particular possible to compute precision and completeness metrics on the underlying mesh exported from blender. These metrics might behave differently, since the ground truth/reference point cloud in our current approach as well as in laser/structured light scanning method have an implicit prioritization. Despite the focus of this work on benchmarking path estimation and dense reconstruction, the applications of SyB3R are by no means limited to those. Instead, it can be easily extended to other application areas such as keypoint matching, surface reconstruction, and optical flow estimation.

Last but not least, SyB3R will be used to create more benchmark datasets and to perform a thorough investigation of modern 3D reconstruction pipelines and their dependency on scene and camera properties. This evaluation will hopefully facilitate innovation by focusing attention on open challenges and modules that still contain large potential for improvement.

Acknowledgements. This paper was supported by a grant (HE 2459/21-1) from the Deutsche Forschungsgemeinschaft (DFG).

References

1. Wu, C.: Towards linear-time incremental structure from motion. In: 2013 International Conference on 3D Vision, 3DV 2013, pp. 127–134 (2013)
2. Dyer, C.: Volumetric scene reconstruction from multiple views. In: Davis, L.S., (ed.) Foundations of Image Understanding, pp. 469–489. Kluwer (2001)
3. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: Conference on Computer Vision and Pattern Recognition (CVPR 2006), vol. 1, pp. 519–526 (2006)
4. Slabaugh, G., Culbertson, B., Malzbender, T., Shafer, R.: A survey of methods for volumetric scene reconstruction from photographs. In: International Workshop on Volume Graphics, pp. 81–101 (2001)
5. Scharstein, D., Szeliski, R.: A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. IJCV **47**, 7–42 (2002)
6. Scharstein, D., Hirschmüller, H., Kitajima, Y., Krathwohl, G., Nesic, N., Wang, X., Westling, P.: High-resolution stereo datasets with subpixel-accurate ground truth. In: German Conference on Pattern Recognition (GCPR 2014) (2014)
7. Jensen, R., Dahl, A., Vogiatzis, G., Tola, E., Aanaes, H.: Large scale multi-view stereopsis evaluation. In: Conference on Computer Vision and Pattern Recognition (CVPR 2014), pp. 406–413 (2014)
8. Strecha, C., von Hansen, W., Gool, L.V., Fua, P., Thoennessen, U.: On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: Conference on Computer Vision and Pattern Recognition (CVPR 2008), pp. 1–8 (2008)
9. Geiger, A., Lenz, P., Urtasun, R.: Are we ready for autonomous driving? the kitti vision benchmark suite. In: Conference on Computer Vision and Pattern Recognition (CVPR 2012), pp. 3354–3361 (2012)
10. Menze, M., Geiger, A.: Object scene flow for autonomous vehicles. In: Conference on Computer Vision and Pattern Recognition (CVPR 2015) (2015)
11. Martorell, M.P., Maki, A., Martull, S., Ohkawa, Y., Fukui, K.: Towards a simulation driven stereo vision system. In: ICPR2012, pp. 1038–1042 (2012)
12. Martull, S., Martorell, M.P., Fukui, K.: Realistic cg stereo image dataset with ground truth disparity maps. In: ICPR2012 Workshop TrakMark2012, pp. 40–42 (2012)
13. Butler, D.J., Wulff, J., Stanley, G.B., Black, M.J.: A naturalistic open source movie for optical flow evaluation. In: Fitzgibbon, A., Lazebnik, S., Perona, P., Sato, Y., Schmid, C. (eds.) ECCV 2012. LNCS, vol. 7577, pp. 611–625. Springer, Heidelberg (2012). doi:10.1007/978-3-642-33783-3_44
14. Ley, A., Hänsch, R., Hellwich, O.: Project Website. http://andreas-ley.com/projects/SyB3R/

15. Ramanath, R., Snyder, W.E., Yoo, Y., Drew, M.S.: Color image processing pipeline. IEEE Signal Process. Mag. **22**(1), 34–43 (2005)
16. Deever, A., Kumar, M., Pillman, B.: Digital camera image formation: processing and storage. In: Digital Image Forensics: There is More to a Picture than Meets the Eye, pp. 45–77. Springer, New York (2013)
17. Muldoon, M., Acosta, J.: Blend Swap. http://www.blendswap.com
18. Debevec, P.E., Malik, J.: Recovering high dynamic range radiance maps from photographs. In: Proceedings of the 24th Annual Conference on Computer Graphics and Interactive Techniques, SIGGRAPH 97, pp. 369–378 (1997)
19. Wu, C.: Siftgpu: a gpu implementation of scale invariant feature transform (sift) (2007). http://cs.unc.edu/ccwu/siftgpu
20. Wu, C., Agarwal, S., Curless, B., Seitz, S.M.: Multicore bundle adjustment. In: Conference on Computer Vision and Pattern Recognition (CVPR 2011), pp. 3057–3064. IEEE (2011)
21. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multi-view stereopsis. IEEE Trans. Pattern Anal. Mach. Intell. **32**(8), 1362–1376 (2010)
22. arenyart: Toad. http://www.blendswap.com/blends/view/74827. Released under CC-Zero
23. James, M.R., Robson, S.: Straightforward reconstruction of 3d surfaces and topography with a camera: accuracy and geoscience application. J. Geophys. Res. Earth Surf. **117**(F3) (2012)
24. ColeHarris: Skull. http://www.blendswap.com/blends/view/21995. Released under CC-Zero
25. matpiet: Spartan helmet. http://www.blendswap.com/blends/view/68806. Released under CC-Zero, slightly adapted for Cycles
26. wesvdes: Cycles wood material. http://blenderartists.org/forum/showthread.php?246113-A-fine-procedural-wood-material-for-Cycles
27. Ley, A., Hänsch, R., Hellwich, O.: Reconstructing white walls: multi-view, multi-shot 3D reconstruction of textureless surfaces. ISPRS Ann. Photogrammetry, Remote Sens. Spat. Inf. Sci. **III**(3), 91–98 (2016)