Multi-Resolution Range Data Fusion for Multi-View Stereo Reconstruction

Andreas Kuhn[1,2], Heiko Hirschmüller[2], Helmut Mayer[1]

[1] Institute for Applied Computer Science, Bundeswehr University Munich
[2] Institute of Robotics and Mechatronics, German Aerospace Center

Abstract. In this paper we present a probabilistic algorithm for multi-view reconstruction from calibrated images. The algorithm is based on multi-resolution volumetric range image integration and is highly separable as it only employs local optimization. Dense depth maps are transformed in an octree data structure with variable voxel sizes. This allows for an efficient modeling of point clouds with very variable density. A probability function constructed in discrete space is built locally with a Bayesian approach. Compared to other algorithms we can deal with extremely big scenes and complex camera configurations in a limited amount of time, as the solution can be split in arbitrarily many parts and computed in parallel. The algorithm has been applied to lab and outdoor benchmark data as well as to large image sets of urban regions taken by cameras on Unmanned Aerial Vehicles (UAVs) and from the ground, demonstrating high surface quality and good runtime performance.

1 Introduction

In spite of all impressive progress, 3D reconstruction from sets of calibrated real world images is still a challenging problem. This was recently demonstrated once again by Vu et al. [21]. While modeling of landscapes is often done in 2.5D, there is a need for detailed 3D modeling particularly for urban regions. Unfortunately, algorithms for 2.5D reconstruction cannot be extended easily. 3D reconstruction algorithms using n-Layer heightmaps expand 2.5D reconstruction algorithms and achieve impressive results for urban regions [7] with one dominant direction.

Recently, multi-view-stereo (MVS) algorithms for 3D modeling made considerable progress concerning accuracy and runtime performance. They yield surfaces of impressive quality, e.g., for the Middlebury multi-view benchmark [16]. Nevertheless, only few of the algorithms can deal with real world data sets such as introduced by Strecha et al. [18]. Too the best of our knowledge there are very few methods which are scalable in a way that they can process hundreds or even thousands of high-resolution images, i.e., with tens of Megapixels. Especially, there are almost no methods which can deal well with image configurations with very different distances to the surfaces, which occur, e.g., when combining images from UAVs and from the ground. Thus, this paper focuses on the fusion of an arbitrary number of 2.5D models, which are in our case range images, into one large model including all 3D details.

2 Previous Work

Algorithms for 3D surface reconstruction are often posed as variational problems minimizing an error function. Among them are algorithms based on global optimization that either do not scale well concerning computational and memory requirements, or have a need for additional information such as the visual hull of the object. The visual hull is often impossible to estimate robustly for cluttered scenes. In particular, global volumetric algorithms based on a regular decomposition of the reconstructed volume, i.e., voxels, can become unfeasibly complex if they are using graph cuts [2, 15] or total variation [23, 24].

There are only a few algorithms which were designed for cluttered outdoor scenes. A current detailed overview is given by Vu et al. [21]. Furukawa et al. [5] presented one of the first algorithm with the potential to handle large scenes without constraints, such as dominant directions [6]. It generates a semi-dense set of patches, which is filtered and optimized based on photometric consistency. Unfortunately, the transformation of the filtered point cloud into triangle meshes does not scale well for big scenes. Arguably the best results for the full 3D reconstruction of large outdoor scenes are obtained at the moment by Vu et al. [21]. Their algorithm starts from semi dense point clouds and derives sets of tetrahedra for visibility checks. After a transformation into triangle meshes these are optimized using graph cuts and variational refinement, restricting the scalability with respect to the runtime efficiency. A scalable algorithm was proposed by Jancosek et al [13]. Filtering on a limited number of images at a time makes their algorithm suitable for large scenes in spite of using global optimization.

Nevertheless, volumetric algorithms using local optimization like level-set methods [8] motivated by Curless and Levoy [3], or EM-based approaches [17] have potential. However, they need several improvements to be scalable for large datasets with challenging geometric configurations.

Local probabilistic optimization is commonly used by algorithms for online processing. To avoid filtering outliers as postprocessing step, these algorithms consider the outlier probability in modelling mixture functions, like the sum of Gaussian and uniform functions [20], or graph-based mixture functions [22, 9]. Those algorithms do not reach the quality of offline-processing. Additionally, they do not take varying distances from the camera into account either.

Like most volumetric approaches, our algorithm is based on range image integration, as proposed by Curless and Levoy [3]. In their algorithm an iso-surface is extracted. A high surface quality is obtained because the algorithm is optimal in the least squares sense. A combination of cumulative weighted signed distance functions is the basis for the extraction of the surface minimizing the least squares distance to all depth maps. A simple, but robust algorithm for multi-view reconstruction based on this algorithm was presented by Goessele et al. [8]. Unfortunately, the least-squares minimization is not suitable for varying surface-qualities. It is essential to consider the quality of the surfaces when dealing with surfaces imaged from strongly differing distances.

We extend these algorithms in three aspects: Firstly, using multi-resolution voxels with dynamic sizes. Like introduced by Fuhrmann et al. [4], our algo-

rithm can model surfaces acquired in configurations with variable distances. Secondly, by using the Gaussian Cumulative Distribution Function (CDF), we also a employ a cumulative weighted distance function, even with a sound statistical background. Thirdly, by adding an additional postprocessing step that filters conflicting outliers, we achieve an improved completeness rates. The outlier detection is visibility-based, like presented by Merell et al. [14].

3 Reconstruction Pipeline

The reconstruction pipeline of our algorithm consists of the following steps:

1. Estimation of disparity maps by Semi-Global-Matching (SGM) [11, 12].
2. Propagation of discrete 1D probability functions on the lines-of-sight.
3. Optimization of points on the surface based on the probability function.
4. Filtering by visibility checks.
5. Triangulation of the resulting point cloud.

For the estimation of disparity maps we use SGM, as it maintains small details due to pixelwise matching and has low processing time and memory requirements for large images. Expressing the disparities as probabilistic functions needs further discussion of the geometric error model, as described in Section 4.

Step 1 can be performed for all suitable image pairs separately. To allow for parallelization of the next steps, the volumetric space is divided and merged at the end. This allows processing on systems with limited main memory and offers scalability for very large scenes. Furthermore, it makes computing faster, for example on clusters with hundreds or thousands of cores. For dividing space, the algorithm runs in a preprocessing step through all depth maps and divides the total volume in subvolumes with the size depending on model resolution, memory size and number of cameras in a subarea. For merging the subvolumes the overlap of neighboring subvolumes has to be big enough so that meshes are equivalent in the merged volumes. More precisely, the overlap has to be at least twice of the local neighborhood used for meshing. This allows for a very easy fusion process as the meshes are equivalent in the inner half of the overlapping area. Triangles in the outer half are simply not considered. Steps 2 to 4 are complex and thus described in Sections 4.2 to 4.4. For step 5 we use a local triangulation for building the final triangle mesh incrementally [1].

4 Volumetric Modelling

We represent depth as a random variable. Because it comprises the most important part, at the moment we only consider the error in the direction of the line-of-sight. This error depends on four geometric parameters [10]:

$$\Delta P_z = \Delta p \frac{P_z^2}{ft} \sqrt{2} \quad , \tag{1}$$

with the length of the baseline t, the focal length of the camera f, the depth P_z and the expected error of the disparity Δp. Besides the last, all other parameters are known for a calibrated image set.

Basically, there is no information about the error of the disparity. Setting it to a constant of half a pixel was empirically found to be a good approximation. We consider a Gaussian $\mathcal{N}(\mu, \sigma)$ with expected value $\mu = d_i^x$, where d_i^x is the depth derived by SGM, and standard deviation $\sigma = \Delta P_z$ with $\Delta p = 0.5$.

In summary, the function for the expected noise of a depth value can be expressed by

$$p(d_i^x) = \mathcal{N}(d_i^x, 0.5\frac{(d_i^x)^2}{ft}\sqrt{2}), \tag{2}$$

4.1 Choice of Voxel Size

An important step to efficiently handle disparity maps of varying density is the choice of the voxel size v_s in the octree. In our case the octree cubes correspond to the voxels. Because the fusion of data is only reasonable for related data, the algorithm chooses the voxelsize for all disparities individually. The idea is that data are fused (cf. Section 4.2) with others having at minimum half and at maximum double the quality. This is due to the fact, that the voxelsize in an octree is rising by a factor of 2. Hence, the voxel is chosen, which has a sidelength of $\sigma < av_s < 2\sigma$, where a is a smoothness term. For practical applications $a \in [2, 3]$ was found to be suitable to maintain the details, but to avoid pitted surfaces. Our algorithm estimates a depending on the number of cameras in the neighborhood. The range is from $a = 2$ for a small number of cameras, like the TempleSparseRing from the Middlebury multi-view benchmark [16], to $a = 3$ for the complete Temple sequence. To avoid quantization artifacts, the probabilistic function is established also in a second, neighboring voxelsize.

4.2 Propagation into Probabilistic Space

We allocate a probability to those voxels in discrete space, which lie on the line-of-sight of our depth value in an area d_a around the estimated depth, whose size corresponds to a couple of voxels. This area can be seen as an $\mathcal{L}^\infty$ norm, which reduces the influence of outliers on the surface quality. For our results, d_a was set to a size of eight voxels.

Along the line-of-sight we estimate the probability, that the voxel v_i lies behind the detected surface (Fig. 1). We use $p(v_i^0)$ and $p(v_i^1)$ for the probability that a voxel lies in front or behind the surface, respectively.

As the probability $p(v_i^1)$ is the integral of the Gaussian from $-\infty$ to the distance a_i of the camera center to the intercept point of the line-of-sight and the voxel v_i, one can take the Gaussian CDF instead of the Probability Density Function (PDF) to estimate it immediately:

$$p(v_i^1) = \int_0^{a_i} \mathcal{N}_{pdf}(x)dx = \mathcal{N}_{cdf}(a_i) \tag{3}$$

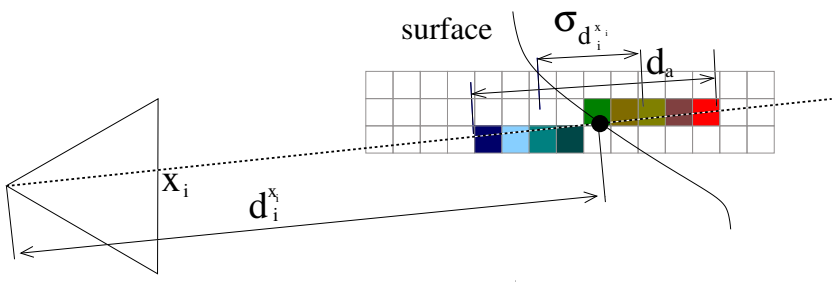

Fig. 1. Discrete probability of surface – Point with pixel coordinate x_i and expected distance $d_i^{x_i}$. $\sigma_{d_i^{x_i}}$ is the standard deviation of the 3D point position along the line-of-sight. The colored voxels represent the probability that a voxel lies behind the surface.

The Gaussian CDF is numerically estimated with the Gauss error function.

Probabilities on different rays from the same image that fall in the same voxel are averaged. Because for several occupancy grid approaches Bayes theorem proved to be appropriate, we employ a statistical framework for fusion of data from different images. We write the Bayesian theorem assuming independent measurements as:

$$p(v_i^1|D=d) \propto p(v_i^1) \prod_{j \in 1,\ldots,n} p(D_j = d_j|v_i^1). \tag{4}$$

Since the probabilistic state of the surface is binary, i.e., a voxel is either occupied or not, we use the Binary Bayes Filter for probabilistic fusion [19].

$$l_i = log\frac{p(v_i^1)}{p(v_i^0)} = log\frac{p(v_i^1)}{1-p(v_i^1)} = \sum_j log\frac{p(v_{ij}^1)}{1-p(v_{ij}^1)} \tag{5}$$

4.3 Optimization of Point Positions on the Surface

The surface is characterized by neighboring voxels for which the probability that one is in front of the surface and the other is behind the surface is maximum. To achieve a better accuracy than the estimated distances d_x^i, we make use of the probabilistic voxel grid described in the last Section. We shift the estimated point along the line-of-sight for twice the standard deviation in both directions. We then consider all voxels I in the area and take the neighboring two, for which the product of probabilities that one is in front of and the other is behind the surface is maximum:

$$\arg\max_{i \in I}(p(v_i^0)p(v_{i+1}^1)) \ . \tag{6}$$

To obtain the position with subvoxel accuracy, we fit a Gaussian to the neighboring voxels of v_i and v_{i+1}. For this we use Maximum Likelihood estimation:

$$d_n = \frac{1}{4}\sum_{j=i-1}^{i+2} d_j \frac{e^{l_j}}{1+e^{l_j}} \tag{7}$$

4.4 Filtering of Point Space by Visibility Checks

Octrees allow for efficient local consistency-checking based on raytracing. Even though raytracing is a global method, we use it locally because outliers have a detrimental influence on the surface quality particularly if they appear nearby. Solitary outliers are ignored in the later step of mesh generation. The idea of local raytracing is to filter conflicting points having relative lower quality as they occlude others with a better quality. In our case the quality is described by the maximum probability of the voxel surface estimated by function (6).

For all voxels v, which have been classified as occupied above, consistency is checked. Rays are cast to those cameras the voxel has been seen from. If there is another occupied voxel on the ray, a conflict is detected. For all voxels the maximum quality of conflicting voxels i on all rays is saved. Those voxels are filtered, whose qualities are worse than the highest quality for all conflicting voxels. If a conflict occurs for voxels on different octree levels, the voxels on the lower resolutions are filtered immediately.

5 Results

We present results for different data sets ranging from laboratory data to large area outdoor models. The given runtimes relate to parallel processing on a cluster with hundreds of CPU cores. A 32 bit architecture is used, so no more than 4 GB of RAM is used per core. All data sets are processed with the same parameter settings. The multi-resolution capability is demonstrated by Figs. 2 and 4.

Fig. 2. Textured and shaded multi-resolution 3D model from 54 images (left column). The size of the largest voxels is about 200 times larger than the size of the smallest.

5.1 Compact Objects

Our algorithm is designed for big cluttered data sets of outdoor scenes. When evaluating it on the Middlebury data sets [16], the untextured laboratory back-

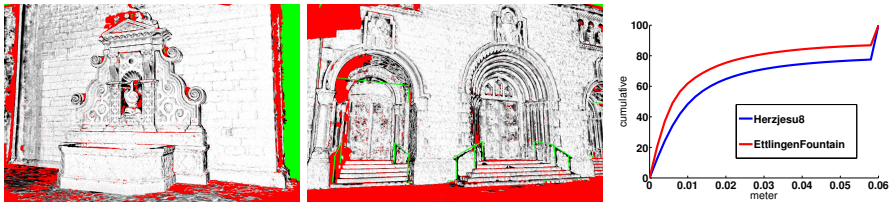

Fig. 3. Evaluation for the EttlingenFountain and Herzjesu8 sequences [18]. In the left and centre images the *absolute error* ranges from $0.002m$ (white) to $0.06m$ (black). The right figure shows the cumulated error function of the *absolute error* in meters.

ground causes spurious responses around the silhouettes. As this does not occur in cluttered scenes, we filter all disparities of the dark background to obtain results which can be compared to other algorithms. With this constraint we get state-of-the art results for all data sets from sparse to full sequences.

Because our algorithm assumes very accurate calibration of the image sets and does not consider texture in the optimization step, our results do not keep up with the best global methods. However, for the full sequences, where the redundancy of the images compensates for the calibration inaccuracy, the probabilistic optimization leads to results comparable to the best in the Dino sequence which is very sparsely textured. Particularly on those data sets with a lack of texture, algorithms using global optimization tend to overfit. In the Middlebury ranking [16] we obtain on average only a place in the middle of all algorithms, but we are able to handle all datasets, from sparse to dense sequences. However, by optimizing locally we are able to process the full sequences in parallel in a couple of minutes, setting our algorithm apart from others. In comparison with Goesele et al. [8] we achieve on average a similar surface quality (0.58mm [8] versus 0.615mm), but much better completeness rates (67% [8] versus 90%). This is a considerable improvement, even considering different densities of the underlying stereo techniques. It should also be noted that in contrast to [8] we do not use additional information like the slant to the surface.

5.2 Large Buildings

A benchmark test set was provided by Strecha et al. [18]. Unfortunately, the evaluation is not provided any more. LIDAR data but without accuracy information is available as ground truth for the EttlingenFountain and the Herzjesu8 sequences. Thus, we could conduct only an evaluation measuring the *absolute* instead of the relative error (Fig. 3) and, therefore, a numerical comparison to other algorithms is not possible. Visually, our results almost reach the quality of the algorithms using global optimization like Furukawa et al. [5]. Particularly, scalable algorithms like Jancosek et al. [13] achieve a much lower quality. Additionally, we obtained results that contain even tiny details of the scene as shown in Fig. 4. Our algorithm is one of the first working on dense depth maps

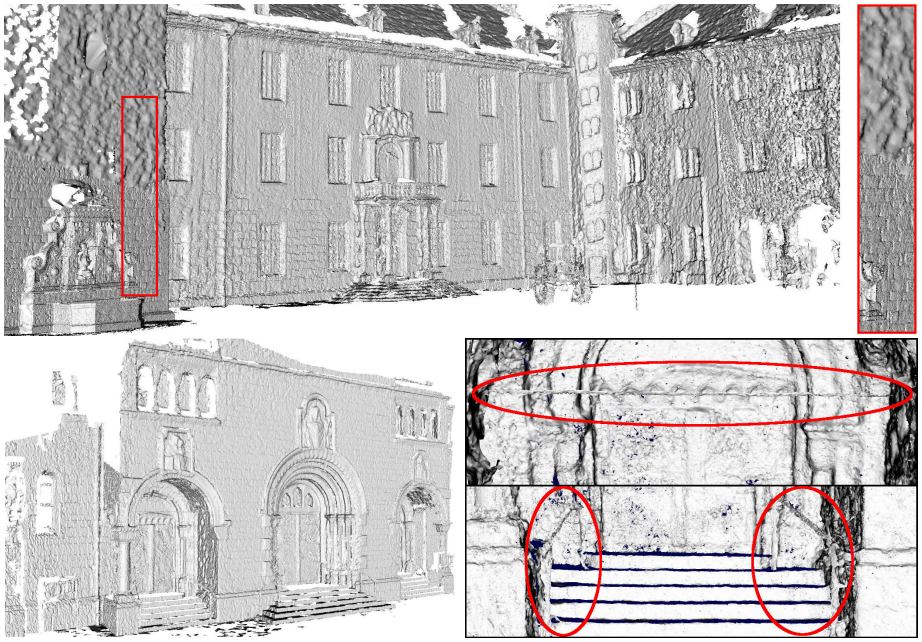

Fig. 4. Result for Strecha's data sets. The upper image shows a combination of the Ettlingen10, Ettlingen30 and the EttlingenFountain images in one model. The region to the right of the fountain shows shows a transition between very different resolutions (cf. also in detail in the right – top: low versus bottom: high). The lower image shows Herzjesu25 with small details as the metal bar (top) and the stair railings.

and reaching state of the art quality. I.e., it has no tendency to generate ghost surfaces in empty areas like algorithms working on sparse point clouds.

We measured the runtime for the largest sequence of Ettlingen30. 3D space was divided in 100 subvolumes which were processed in parallel. The total runtime amounts to one hour, mainly split in depth estimation ($\approx$ 20 min), modeling ($\approx$ 30 min) and meshing ($\approx$ 10 min). The runtime could be further reduced by using more cores and more subvolumes.

5.3 Large Area Models

Our method makes it possible to process a nearly arbitrary number of calibrated high-resolution images. For demonstration we processed a data set acquired with ten Megapixel cameras on different UAVs. It shows a village with a large number of buildings in more then 600 images.

The model, for which one view with details is shown in Fig. 5 was computed, based on over thousand submodels with a total runtime of about five hours.

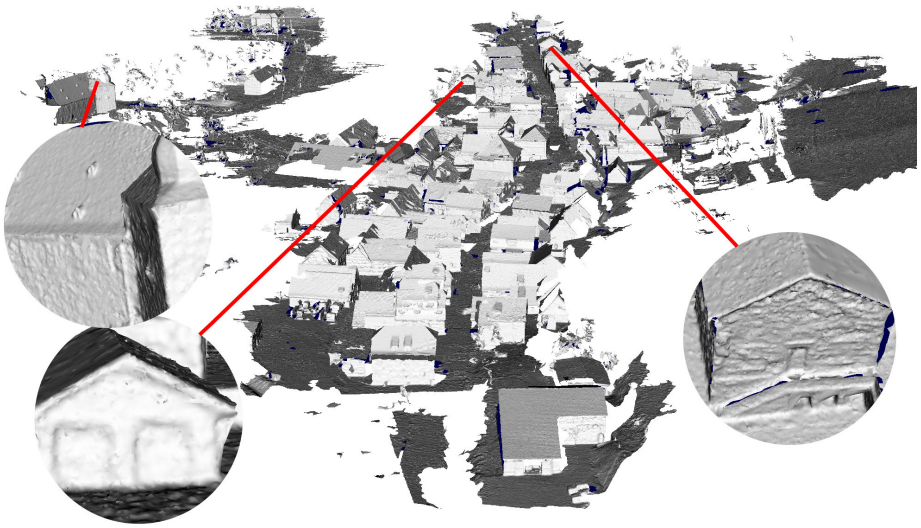

Fig. 5. 3D model of a village from 603 images made with UAVs showing small details.

6 Conclusions and Outlook

A flexible multi-resolution approach for multi-view stereo reconstruction has been presented. Because it uses local optimization, it allows for the 3D reconstruction of scenes of nearly unlimited size with complex imaging configurations. Furthermore, the images are processed in parallel at any time. In spite of this, we obtain a quality that is state of the art for ambitious image sequences.

We made three main contributions. First, a local multi-resolution approach for optimization of range images suitable for images taken from very different distances and thus varying detail. Second, a probabilistic function of a 3D point using the Gaussian CDF providing a better statistical background for fusion of different qualities. Third, a postprocessing step for filtering occlusions having a lower quality in terms of the probabilistic function.

Concerning future work, several issues regarding error modeling of points need to be analyzed in more detail. Currently, points are represented as random variables with a Gaussian probability function of the disparity error that is assumed to be constant. Furthermore, also other attributes such as the strength of the texture and the angle with the normal vector of the surface, e.g., derived from nearest neighbors, probably have an influence on the disparity error. At the moment no 3D regularization is considered. There is a potential to get a better completeness using a local regularization term. Finally, the probabilistic representation allows us to take additional information like the covariance of the projection matrices from calibration into account.

References

1. Bodenmüller, T.: Streaming Surface Reconstruction from Real Time 3D Measurements. Ph.D. thesis, Technical University Munich (2009)
2. Campbell, N.D., Vogiatzis, G., Hernández, C., Cipolla, R.: Using multiple hypotheses to improve depth-maps for multi-view stereo. In: CVPR (2008)
3. Curless, B., Levoy, M.: A volumetric method for building complex models from range images. In: SIGGRAPH (1996)
4. Fuhrmann, S., Goesele, M.: Fusion of depth maps with multiple scales. In: Proceedings of the 2011 SIGGRAPH Asia Conference (2011)
5. Furukawa, Y., Ponce, J.: Accurate, dense, and robust multiview stereopsis. PAMI (2010)
6. Gallup, D., Frahm, J.M., Mordohai, P., Yang, Q., Pollefeys, M.: Real-time plane-sweeping stereo with multiple sweeping directions. In: CVPR (2007)
7. Gallup, D., Pollefeys, M., Frahm, J.M.: 3d reconstruction using an n-layer heightmap. In: DAGM (2010)
8. Goesele, M., Curless, B., Seitz, S.M.: Multi-view stereo revisited. In: CVPR (2006)
9. Guan, L., Franco, J.S., Pollefeys, M.: 3D Object Reconstruction with Heterogeneous Sensor Data. In: 3DPVT (2008)
10. Hirschmüller, H.: Stereo Vision based mapping and immediate virtual walkthroughs. Ph.D. thesis, De Montfort University (2003)
11. Hirschmüller, H.: Stereo processing by semiglobal matching and mutual information. PAMI (2008)
12. Hirschmüller, H., Scharstein, D.: Evaluation of stereo matching costs on images with radiometric differences. PAMI (2009)
13. Jancosek, M., Shekhovtsov, A., Pajdla, T.: Scalable multi-view stereo. In: 3DIM09 (2009)
14. Merrell, P., Akbarzadeh, A., Wang, L., Mordohai, P., Frahm, J.M., Yang, R., Nistér, D., Pollefeys, M.: Real-time visibility-based fusion of depth maps. In: CVPR (2007)
15. Mücke, P., Klowsky, R., Goesele, M.: Surface reconstruction from multi-resolution sample points. In: VMV (2011)
16. Seitz, S.M., Curless, B., Diebel, J., Scharstein, D., Szeliski, R.: A comparison and evaluation of multi-view stereo reconstruction algorithms. In: CVPR (2006)
17. Strecha, C., Fransens, R., Gool, L.J.V.: Combined depth and outlier estimation in multi-view stereo. In: CVPR (2006)
18. Strecha, C., von Hansen, W., Gool, L.J.V., Fua, P., Thoennessen, U.: On benchmarking camera calibration and multi-view stereo for high resolution imagery. In: CVPR (2008)
19. Thrun, S., Burgard, W., Fox, D.: Probabilistic Robotics (Intelligent Robotics and Autonomous Agents). The MIT Press (2005)
20. Vogiatzis, G., Hernández, C.: Video-based, real-time multi-view stereo. Image Vision Comput. (2011)
21. Vu, H.H., Labatut, P., Pons, J.P., Keriven, R.: High accuracy and visibility-consistent dense multiview stereo. PAMI (2012)
22. Woodford, O.J., Vogiatzis, G.: A generative model for online depth fusion. In: ECCV (2012)
23. Zach, C.: Fast and high quality fusion of depth maps. 3DPVT (2008)
24. Zach, C., Pock, T., Bischof, H.: A globally optimal algorithm for robust tv-l1 range image integration. In: ICCV (2007)