Turning Mobile Phones into 3D Scanners

Kalin Kolev, Petri Tanskanen, Pablo Speciale and Marc Pollefeys

Department of Computer Science
ETH Zurich, Switzerland

Abstract

In this paper, we propose an efficient and accurate scheme for the integration of multiple stereo-based depth measurements. For each provided depth map a confidence-based weight is assigned to each depth estimate by evaluating local geometry orientation, underlying camera setting and photometric evidence. Subsequently, all hypotheses are fused together into a compact and consistent 3D model. Thereby, visibility conflicts are identified and resolved, and fitting measurements are averaged with regard to their confidence scores. The individual stages of the proposed approach are validated by comparing it to two alternative techniques which rely on a conceptually different fusion scheme and a different confidence inference, respectively. Pursuing live 3D reconstruction on mobile devices as a primary goal, we demonstrate that the developed method can easily be integrated into a system for monocular interactive 3D modeling by substantially improving its accuracy while adding a negligible overhead to its performance and retaining its interactive potential.

1. Introduction

There is a growing demand for easy and reliable generation of 3D models of real-world objects and environments. Vision-based techniques offer a promising accessible alternative to active laser scanning technologies with competitive quality. While the acquisition of photographs is trivial and does not require expertise, the generation of an image set, which ensures the desired accuracy of the subsequently obtained 3D model, is a more challenging task. Camera sensor noise, occlusions and complex reflectance of the scene often lead to failure in the reconstruction process but their appearance is difficult to predict in advance. This problem is addressed by monocular real-time capable systems which can provide useful feedback to the user in the course of the reconstruction process and assist him in planning his movements. Interactive systems based on video cameras [6] and depth sensors [14, 7] have been demonstrated. Unfortunately, their usability is limited to desktop computers and high-end laptops as they rely on massive processing resources like multi-core CPUs and powerful GPUs. This precludes applications of casual capture of 3D models but also reduces the user's benefit from the provided visual feedback since his attention should steadily be redirected from the capturing device to the display and back.

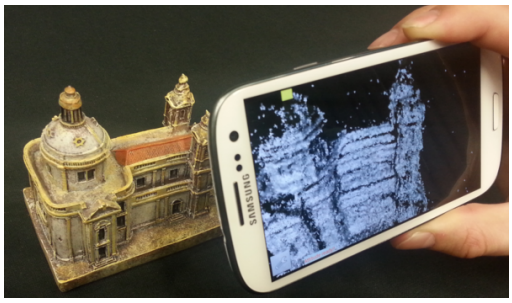

Figure 1. This paper deals with the problem of live 3D reconstruction on mobile phones. The proposed approach allows to obtain 3D models of pleasing quality interactively and entirely on-device.

Modern smartphones and tablet computers offer improved mobility and interactivity, and open up new possibilities for live 3D modeling. While recent mobile devices are equipped with a substantial computational power like multi-core processors and graphics processing cores, their capabilities are still far from those of desktop computers. To a great extent, these restrictions render most of the currently known approaches inapplicable on mobile devices, giving room to research in the direction of specially designed, efficient on-line algorithms to tackle all the limitations of embedded hardware architectures. While first notable attempts for interactive 3D reconstruction on smartphones have already been presented [8, 13, 20], an application able to produce high-quality 3D models of real-world objects and environments is still illusive.

This paper can be regarded as an effort towards closing the gap between the capabilities of current systems for live 3D reconstruction on mobile devices and the accuracy of similar interactive systems designed for high-end systems

(see Fig. 1). Its main contribution is the development of an efficient and accurate scheme for integrating multiple stereo-based depth hypotheses into a compact and consistent 3D model. Thereby, various criteria based on local geometry orientation, underlying camera setting and photometric evidence are evaluated to judge the reliability of each measurement. Based on that, the proposed fusion technique justifies the integrity of the depth estimates and resolves visibility conflicts. We demonstrate the performance of the developed method within a framework for real-time 3D reconstruction on a mobile phone and show that the accuracy of the system can be improved while retaining its interactive rate.

2. Related Work

As the current paper deals with the problem of depth map fusion, which is a classical problem in multi-view 3D reconstruction, it is related to a myriad of works on binocular and multi-view stereo. We refer to the benchmarks in [16], [17] and [18] for a representative list. However, most of those methods are not applicable to our particular scenario as they are not incremental in nature or don't meet the efficiency requirements of embedded systems. In the following, we will focus only on approaches which are conceptually related to ours.

Building upon pioneering work on reconstruction with a hand-held camera [10], Pollefeys *et al*. [11] presented a complete pipeline for real-time video-based 3D acquisition. The system was developed with focus on capturing large-scale urban scenes by means of multiple video cameras mounted on a driving vehicle. Yet, despite its real-time performance, the applicability of the system on a live scenario is not straightforward. Nevertheless, we drew some inspiration from the utilized depth map fusion scheme, originally published in [4]. The first methods for real-time interactive 3D reconstruction were proposed by Newcombe *et al*. [5] and Stuehmer *et al*. [19]. In both works, a 3D representation of the scene is obtained by estimating depth maps from multiple views and converting them to triangle meshes based on the respective neighborhood connectivity. Even though these techniques cover our context, they are designed for high-end computers and are not functional on mobile devices due to some time-consuming optimization operations. Another approach for live video-based 3D reconstruction, which is conceptually similar to ours, was proposed by Vogiatzis and Hernandez [21]. Here, the captured scene is represented by a point cloud where each generated 3D point is obtained as a probabilistic depth estimate by fusing measurements from different views. Similar to the already discussed methods, this one also requires substantial computational resources. Another key difference to our framework is the utilization of a marker to estimate camera poses, which entails considerable limitations in terms of usability. Recently, the work of Pradeep *et al*. [12] appeared. It presents another pipeline for real-time 3D reconstruction from monocular video input based on volumetric depth-map fusion. Again, those techniques are developed for high-end computers and have never been demonstrated on embedded systems.

Probably the most similar method to ours was proposed in [22] and subsequently generalized in [3, 1]. Therein, a system for interactive in-hand scanning of objects was demonstrated. Similar to the approach, presented in this paper, it relies on a surfel representation of the modeled 3D object. However, the developed fusion scheme is designed for measurements stemming from active sensors, which are considerably more accurate than stereo-based ones. Therefore, the employed confidence estimation is quite different from this proposed in the current paper.

Recently, the first works on live 3D reconstruction on mobile devices appeared. Wendel *et al*. [23] rely on a distributed framework with a variant of [2] on a micro air vehicle. A tablet computer is barely used for visualization while all demanding computations are performed on a separate server machine. Sankar *et al*. [15] proposed a system for interactively creating and navigating through visual tours. Thereby, an approximate geometry of indoor environments is generated based on strong planar priors and some user interaction. Pan *et al*. [8] demonstrated an automatic system for 3D reconstruction capable of operating entirely on a mobile phone. However, the generated 3D models are not very precise due to the sparse nature of the approach. Prisacariu *et al*. [13] presented a shape-from-silhouette framework running in real time on a mobile phone. Despite the impressive performance, the method suffers from the known weaknesses of silhouette-based techniques, e. g. the inability to capture concavities. Tanskanen *et al*. [20] developed a dense stereo-based system for 3D reconstruction capable of interactive rates on a mobile phone. We use a similar system as a starting point and show that considerable accuracy improvements can be achieved by integrating the proposed approach without affecting its interactive potential.

3. Multi-Resolution Depth Map Computation

In the first stage of the 3D modeling pipeline depth maps are created from a set of keyframes, and corresponding calibration information and camera poses. Here, we adopt the methodology proposed in [20]. Apart from being efficient and accurate, it is particularly appealing due to the potential of the utilized multi-resolution depth map computation scheme for implementation on mobile GPUs. In the following, we outline the procedure for the sake of completeness. More details can be found in [20].

A camera motion tracking system produces a series of keyframes and associated camera poses which are provided to a dense modeling module. As abrupt jumps in the cam-

era motion cannot be expected, a straightforward strategy is to maintain a sliding window containing the most recent keyframes and use them for stereo matching but also to check consistency between different depth maps. Pursuing an interactive framework on mobile devices, binocular stereo instead of multi-view stereo is applied to minimize the memory access overhead. In particular, a newly arrived keyframe is used as a reference image and is matched to an appropriate image in the current buffer. Thereby, a multi-resolution scheme for the depth map computation is employed to reduce the computational time and to avoid local maxima of the photoconsistency score along the considered epipolar segments. When moving from one resolution level to the next, the depth range is restricted based on the depth estimates at neighboring pixels. Additionally, computations are limited to pixels exhibiting sufficient local image texturedness within regions where the current 3D model has not reached the desired degree of maturity. The result is a depth map possibly corrupted by noise due to motion blur, occlusions, lack of texture, presence of slanted surfaces etc. A very efficient and effective filtering procedure is applied to remove the outliers. Thereby, the consistency of each depth measurement is tested on agreement with the other depth maps within the sliding window. If a sufficient number of confirmations is reached, the measurement is retained, otherwise it is discarded as an outlier. Subsequently, the depth map is smoothed by applying bilateral filtering to improve the precision of the depth values.

The final output of this stage is a series of partial depth maps. We build upon this scheme and additionally compute a normal vector to each depth measurement by applying a local plane fitting procedure. Isolated points with insufficient support within the neighborhood are discarded. In the next stage, all those measurements are merged into a unified 3D model of the scene.

4. Confidence-Based Depth Map Fusion

A central issue in the design of a depth map fusion approach is the representation of the modeled scene. While triangle meshes exhibit a common geometric representation, they do not seem well-suited for interactive applications running in real time since considerable efforts are needed to guarantee the integrity and consistency of the mesh topology after adding, updating or removing any vertices. Note that the user is expected to make use of the live visual feedback and recapture certain parts of the scene until the desired surface quality is reached. For that reason, we rely on a *surfel* representation [9]. A surfel s_j consists of a position p_j, normal vector N_j, color C_j and a confidence score c_j which is defined as the difference between a cumulative inlier and outlier weight, i. e. $c_j = W_j^{(in)} - W_j^{(out)}$. Additional attributes like local patch radius or visibility information could be maintained if needed. The utilized surfel representation offers the required resilience since the unstructured set of surfels can easily be kept consistent throughout any modifications.

The proposed depth map fusion approach relies on the following scheme: When a new depth map becomes available, a weight is assigned to each pixel measurement reflecting its expected accuracy. Based on this input, the surfel model is modified by adding new surfels, updating or removing existing ones. In the following, these steps are explained in more detail.

4.1. Confidence-Based Weighting

The accuracy of a depth measurement, obtained from stereo matching, depends on many factors, e. g. inherent scene texture, geometry orientation, camera noise, distance between the scene and the camera device etc. In an effort to capture all those aspects we assign different weights to each estimate and combine them subsequently to obtain a final weighting score that expresses our confidence in the particular depth value.

Geometry-Based Weights. The accuracy of a depth measurement depends on the local surface orientation at that point. The depth measurement is more accurate when the observed geometry is fronto-parallel and less accurate at grazing viewing angles. As a local normal vector is computed to each depth estimate, those cases can be identified by considering the scalar product between the normal and the respective viewing direction of the camera. If $n_x \in S^2$ denotes the normal vector and $v_x \in S^2$ stands for the normalized reverted viewing direction of the camera for a pixel $x \in \Omega \subset \mathbb{Z}^2$ within the image domain, we define a geometry-based weight to x as

$$w_g(x) = \begin{cases} \dfrac{\langle n_x, v_x \rangle - \cos(\alpha_{max})}{1 - \cos(\alpha_{max})}, & \text{if } \sphericalangle(n_x, v_x) \leq \alpha_{max} \\ 0, & \text{otherwise,} \end{cases} \tag{1}$$

where α_{max} is a critical angle at which the measurements are considered unreliable and is set to $80°$ throughout all experiments. The weight defined in (1) takes on values within $[0, 1]$. Note that it does not directly depend on the depth estimates. However, there is an indirect relation as the computation of the normal vectors relies on them.

Camera-Based Weights. The accuracy of a depth measurement, obtained from binocular stereo, depends on the utilized camera setting. For example, short baselines implicate high depth imprecision as larger changes of the depth along the visual rays result in small projection footprints on the image plane of the non-reference camera. Analogously, increasing the image resolution or moving the camera closer to the scene leads to more accurate depth estimates. Based on these observations, a camera-based

weight could be defined by measuring the depth deviation corresponding to a certain shift (for example one pixel) along the respective epipolar line. Yet, this cannot be realized efficiently since it involves an additional triangulation operation. Further complications pose the discrepance between viewing ray traversal and pixel sampling. Instead, we revert the inference and measure the pixel shift δ that a certain offset along the ray produces. More concretely, the offset along the visual rays is set to $1/600$ of the depth range. Then, a camera-based weight to a pixel x is defined as

$$w_c(x) = 1 - e^{-\lambda\delta}, \tag{2}$$

where $\lambda \in \mathbb{R}$ is a parameter specifying the penalizing behavior of the term and is set to 5.0 throughout all experiments, and δ is measured in pixel coordinates. Note that $w_c \in [0, 1]$ is inversely proportional to the estimated depths, i. e. larger depths get lower weights and smaller depths get higher weights. This corresponds to the intuition that parts of the scene closer to the camera are expected to be reconstructed more accurately than parts further away from the camera. Moreover, the length of the baseline is also taken into account by the formulation in (2). In particular, depth maps, obtained from short baselines, will generally be weighted lower.

Photoconsistency-Based Weights. Probably the most straightforward criterion to judge the accuracy of a depth measurement is its photoconsistency score. However, this is also the least discriminative criterion since the provided depth maps are already checked for consistency and filtered, thus, the respective matching scores are expected to be high. The easiest way to obtain the photoconsistency value to a depth estimate is to use the one delivered by the stereo module. Yet, as normal information is available at that point, a more accurate measure can be employed. Here, we adopt normalized cross-correlations (NCC) over 5×5 patches where the provided normal vectors are leveraged to warp the patches from the reference image to the second view. Then, for a pixel x we specify

$$w_{ph}(x) = \begin{cases} NCC(x), & \text{if } NCC(x) \geq thr \\ 0, & \text{otherwise} \end{cases} \tag{3}$$

as the photoconsistency-based weight. Thereby, thr is a threshold parameter set to 0.65 throughout all experiments, and $NCC(x)$ denotes the NCC score for the depth and the normal at x. Again, we have $w_{ph} \in [0, 1]$. It should be noted that the computation of the photoconsistency-based weights is more time-consuming than that of the geometry-based and the camera-based ones while having the least contribution to the final weighting values. For this reason, it could be omitted when more efficiency is required.

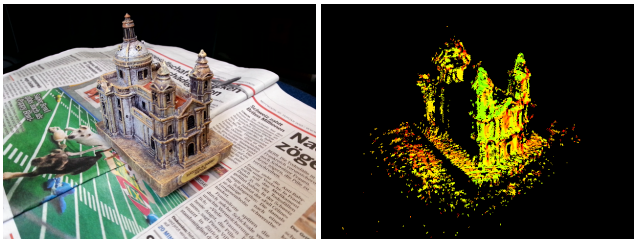

Figure 2. Confidence-based weighting of depth measurements. The reference image of a stereo pair and corresponding color-coded weights to the computed depth estimates. Green represents high weighting, red represents low weighting. Note that pixels, where the local normal vector points away from the camera, get small weights. Also, more distant measurements tend to be weighted low.

The last step is to combine all weight estimates and to provide a final overall weight to each depth measurement in the provided depth map. To this end, for each x we set

$$w(x) = w_g(x) \cdot w_c(x) \cdot w_{ph}(x). \tag{4}$$

The overall weight lies in $[0, 1]$ and will be high only when all three weights, the geometry-based one, the camera-based one and the photoconsistency-based one, are high. In other words, a measurement is considered as accurate if it is accurate from geometric, stereoscopic and photometric point of view.

Fig. 2 shows an example of the estimated weighting for a depth map capturing a small church figurine. For all depth measurements the corresponding weights are computed according to (4). Note that the effects from applying the geometry and the camera term are clearly visible. Indeed, pixels, where the local normal vector points away from the camera, get small weights. Also, more distant measurements tend to be weighted low. The effect from applying the photoconsistency term is less noticeable.

4.2. Measurement Integration

When a new depth map becomes available and confidence weights are assigned to all measurements, the provided data is used to update the current surfel cloud. This is done using three basic operations: *surfel addition*, *surfel update* and *surfel removal*. New surfels are created for parts of the depth map that are not explained by the current model. Surfels that are in correspondence with the input depth map are updated by integrating the respective depth and normal estimates. Surfels with confidence value below a certain threshold are removed from the cloud. In the following, these operations are explained in more detail.

Surfel addition. Surfels are added in those parts where the depth map is not covered by model surfels. Of course,

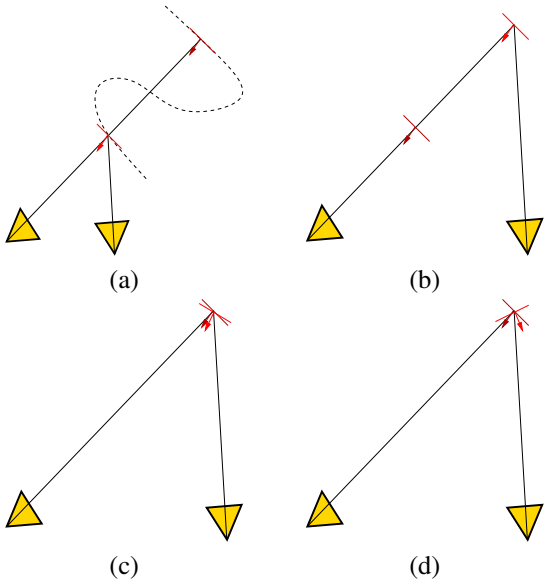

Figure 3. Different cases for a surfel update. Red denotes the incoming measurement and dark red - the surfel. (a) Measurement is in front of the observed surfel. There is no visibility conflict. (b) Measurement is behind the observed surfel. There is a visibility conflict. (c) Measurement and observed surfel match. (d) Depths of the measurement and the observed surfel match but not their normals. There is a visibility conflict. See text for more details.

for the initial depth map all measurements will create new surfels. For each newly created surfel the position and normal vector are set according to the depth and normal estimate of the measurement. The color is set to the color of the respective image pixel. The cumulative inlier weight is initialized with the weight of the depth measurement and the cumulative outlier weight - with zero.

Surfel update. If the projection of a surfel coincides with a provided depth measurement, the surfel is updated. Let $s_j = (p_j, N_j, C_j, W_j^{(in)}, W_j^{(out)}, c_j)$ be the surfel of interest. If there are multiple surfels along the same visual ray, we take the one closest to the camera center that is expected to be visible. Additionally, we maintain a state vector $X_j = (p_1, p_2, p_3, \theta, \phi) \in \mathbb{R}^5$ encoding its current position and normal. Thereby, the normal is represented by means of a polar angle θ and an azimuth angle ϕ. When a new surfel is created, a spherical coordinate system is generated with the provided normal estimate as the first base vector. Let $x = \Pi(p_j)$ be the projection of the surfel onto the image plane of the current frame and let $d(p_j)$ be its depth with respect to the camera center. At x the given depth map provides a depth measurement d_x and a normal measurement n_x. In addition to that, we get a weight $w(x)$ reflecting the accuracy of the estimates.

Now, we have to update the surfel based on this input. There are four different update cases (see Fig. 3):
(1) $d(p_j) \gg d_x$: The depth measurement occludes the model surfel. By itself this is not a visibility conflict since the depth map could capture a different part of the surface. The dashed line in Fig. 3(a) shows a potential visibility configuration. In fact, this is the most delicate case as both the surfel and the measurement could be outliers. Here, we just ignore the depth measurement and do not perform any surfel update. Note that this could cause problems when parts of the surface are acquired which are in the line of sight of already reconstructed ones (with the same orientation). However, this is unlikely to occur in practice as the user usually captures more accessible parts first before moving to locations that are more difficult to reach.
(2) $d(p_j) \ll d_x$: The depth measurement is behind the model surfel. This is a clear visibility conflict. In this case we add the measurement's weight to the cumulative outlier weight of the surfel, i. e.

$$W_j^{(out)} \leftarrow W_j^{(out)} + w(x). \tag{5}$$

(3) $\frac{|d(p_j)-d_x|}{d(p_j)} < \epsilon$ and $\sphericalangle(N_j, n_x) \leq 45°$: The measurement and the model surfel match, both in terms of depth and normal orientation. Then, the surfel position and normal are updated accordingly. In particular, we compute a running weighted average

$$\begin{aligned} X_j &\leftarrow \frac{W_j^{(in)} X_j + w(x) X_x}{W_j^{(in)} + w(x)} \\ W_j^{(in)} &\leftarrow W_j^{(in)} + w(x), \end{aligned} \tag{6}$$

where the pixel's depth d_x and normal n_x are converted into a state vector X_x.
(4) $\frac{|d(p_j)-d_x|}{d(p_j)} < \epsilon$ and $\sphericalangle(N_j, n_x) > 45°$: The measurement and the model surfel match in terms of depth but the orientations of their normals deviate from each other. We consider this as a visibility conflict and increment the cumulative outlier weight according to (5).

Recall that there are two additional attributes to each surfel - a color C_j and a confidence score c_j. The color is set to the color of the pixel with the largest weight $w(x)$ used in the fusion process for the surfel. The confidence measure is defined as the difference between cumulative inlier weight and cumulative outlier weight, i. e. $c_j = W_j^{(in)} - W_j^{(out)}$, and has to be updated each time one of those values is modified.

Surfel removal. Surfels are removed from the cloud during the acquisition process if their confidence falls below a threshold. We set this threshold to -0.5 throughout all conducted experiments. Note that the removal of surfels opens up gaps that can be filled by new more accurate surfels.

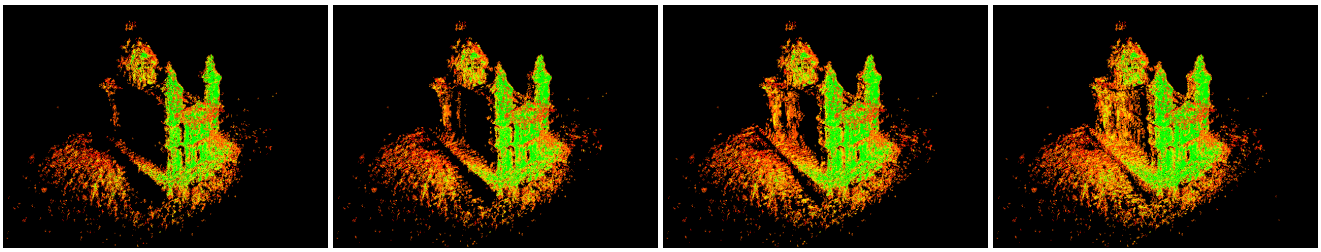

Figure 4. Confidence evolution during reconstruction. Visualized are the color-coded confidence scores of the generated surfels for consecutive frames of a real-world sequence. Green represents high confidence, red represents low confidence. An input image from the same viewpoint can be seen in Fig. 2. Note how the confidence values of surfels, seen from different directions, increase in the course of reconstruction.

One could wonder why the normals are integrated in the proposed depth map fusion scheme. In fact, they can be obtained in a post-processing step by considering the neighborhood of each point within the point cloud. There are two main reasons for this design decision. First, the normal information is useful as it captures the local geometric structure of each depth measurement and enables the identification of accidental matches like in the case depicted in Fig. 3(d). Second, the proposed scheme allows to leverage the neighborhood relation between different measurements, provided by the camera sensor. Moreover, note that the proposed depth map fusion procedure is incremental and lends itself to online applications. Also, it allows reconstructed parts of the scene to be recaptured by providing additional depth data and improving the accuracy of the respective subset of the surfel cloud.

Fig. 4 depicts the evolution of the confidence scores of the generated surfels for consecutive frames of a real-world sequence. Note that the confidence values are small for newly created surfels but increase in the course of the acquisition process if they are observed from other viewpoints.

5. Experimental Results

We validate the proposed confidence-based depth map fusion scheme by comparing it to two state-of-the-art real-time capable alternatives. Furthermore, we demonstrate its performance by integrating it into a system for live 3D reconstruction running on a mobile phone.

5.1. Comparison to Alternative Techniques

For the sake of comparison we implemented two alternative techniques meeting the efficiency requirements of the application at hand.

The first one is the merging method used in [20]. Thereby, the interconnection between the different input depth maps is exploited barely to identify inconsistencies and to filter out outliers. All consistent depth measurements are back-projected to 3D and merged into a unified point cloud. Moreover, a coverage mask based on photometric criteria is estimated in each step to reduce the generation of redundant points. See [20] for more details.

To evaluate the viability of the confidence-based weighting approach, we combined the developed fusion scheme with the weight computation proposed in [4]. The basic idea of this strategy is to judge the accuracy of each depth measurement by analyzing the photoconsistency distribution along the respective visual rays. Rays with a single sharp maximum are expected to provide more accurate estimates than those exhibiting a shallow maximum or several local maxima. More details can be found in [4].

Fig. 5 shows the reconstructions generated by applying all three techniques on a real-world image sequence. One of the input images can be seen in Fig. 2. Camera poses were obtained by applying a version of [2]. Note that the approach of [20] does not explicitly estimate normals to the generated point cloud. Therefore, for the purpose of rendering we assigned to each point a normal vector based on the depth map that was used to create it. For the other two approaches we used the normal estimates obtained online from the fusion process. It is evident that while all three methods achieve a high degree of completeness, the proposed one with confidence-based weighting outperforms the others in terms of accuracy. The technique in [20] produces an oversampling of the scene and is more sensitive to noise than the other two as each 3D point is based on a single depth measurement. This proves the importance of a depth map fusion scheme. Moreover, the reconstruction obtained with the proposed confidence-based weighting is significantly more accurate than the one relying on the weighting of [4], which validates the deployment of geometric and camera-based criteria in the depth integration process.

5.2. Live 3D Reconstruction on a Mobile Phone

Pursuing a system for live 3D reconstruction running on mobile phones as a primary goal, we integrated the proposed method into the framework of [20]. This substantially improved its accuracy while adding a negligible overhead of less than a second per processed image. In the follow-

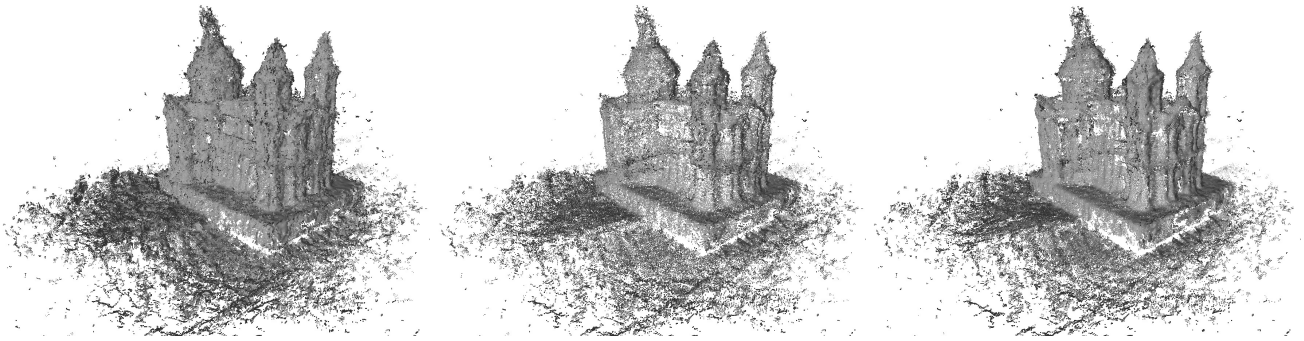

Figure 5. Comparison to alternative techniques. *From left to right:* Reconstructions with the depth map merging technique in [20], the developed fusion scheme with the weighting suggested in [4] and the complete approach proposed in this paper. One of the images in the input sequence can be seen in Fig. 2. The reconstructions contain 311135, 161647 and 181077 points, respectively. While all three methods achieve a high degree of completeness, the proposed approach with confidence-based weighting outperforms the other two in terms of accuracy.

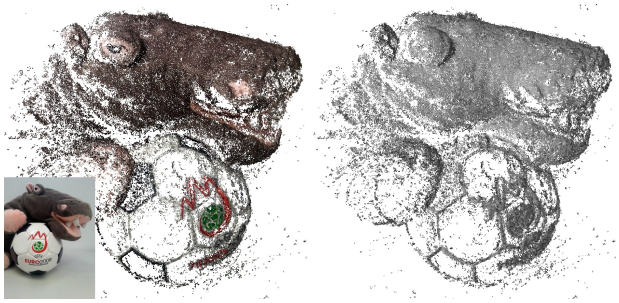

Figure 6. Hippopotamus. Rendering of the reconstructed surfel cloud with colors and shading, and a reference image of the object. Note the accurate reconstruction of the head.

Figure 7. Relief. Rendering of the reconstructed surfel cloud with colors and shading, and a reference image of the object. The model was captured outdoors.

ing, multiple reconstructions of real-world objects, generated interactively on a Samsung Galaxy SIII and a Samsung Galaxy Note 3, are depicted.

Fig. 6 depicts the reconstruction of a fabric toy of a hippopotamus. Expectedly, homogeneous regions (e. g. on the ball) lead to holes in the 3D model. However, the well-textured head of the hippopotamus is reconstructed at high geometric precision.

Figure 8. Buddha statue. Rendering of the reconstructed surfel cloud with colors and shading, and a reference image of the object. Note the accurately captured small-scale details.

Fig. 7 shows the reconstruction of a relief on a decoration vase. The model was captured outdoors under sunlight conditions. Note that this is a known failure case for many active sensors.

The capabilities of current mobile devices for in-hand scanning are further demonstrated in Fig. 8. The reconstruction of a Buddha statue in a museum is visualized. Even though the generated point cloud exhibits a substantial amount of high-frequency noise, many small-scale details like the wrinkles of the clothing or the face features are captured in the reconstruction.

6. Conclusion

We presented an efficient and accurate method for confidence-based depth map fusion. At its core is a two-

stage approach where confidence-based weights, that reflect the expected accuracy, are first assigned to each depth measurement and subsequently integrated into a unified and consistent 3D model. Thereby, the maintained 3D representation in form of a surfel cloud is updated dynamically so as to resolve visibility conflicts and ensure the integrity of the reconstruction. The advantages of the proposed approach in terms of accuracy improvements are highlighted by a comparison to alternative techniques which meet the underlying efficiency requirements. Additionally, the potential of the developed method is emphasized by integrating it into a state-of-the-art system for live 3D reconstruction running on a mobile phone and demonstrating its performance on multiple real-world objects.

Acknowledgments

We thank Lorenz Meier for helping with the supplementary material. This work is funded by the ETH Zurich Postdoctoral Fellowship Program, the Marie Curie Actions for People COFUND Program and ERC grant no. 210806.

References

[1] M. Keller, D. Lefloch, M. Lambers, S. Izadi, T. Weyrich, and A. Kolb. Real-time 3D reconstruction in dynamic scenes using point-based fusion. In *3DV*, pages 1–8, 2013. 2

[2] G. Klein and D. Murray. Parallel tracking and mapping on a camera phone. *ISMAR*, pages 83–86, 2009. 2, 6

[3] M. Krainin, P. Henry, X. Ren, and D. Fox. Manipulator and object tracking for in-hand 3D object modeling. *Int. J. Rob. Res.*, 30(11):1311–1327, 2011. 2

[4] P. Merrell, A. Akbarzadeh, L. Wang, P. Mordohai, J.-M. Frahm, R. Yang, D. Nistr, and M. Pollefeys. Real-time visibility-based fusion of depth maps. In *IEEE International Conference on Computer Vision (ICCV)*, pages 1–8, 2007. 2, 6, 7

[5] R. A. Newcombe and A. J. Davison. Live dense reconstruction with a single moving camera. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2010. 2

[6] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. DTAM: Dense tracking and mapping in real-time. In *IEEE International Conference on Computer Vision (ICCV)*, pages 2320–2327, 2011. 1

[7] R. A. Newcombe et al. Kinectfusion: Real-time dense surface mapping and tracking. In *ISMAR*, pages 127–136, 2011. 1

[8] Q. Pan, C. Arth, E. Rosten, G. Reitmayr, and T. Drummond. Rapid scene reconstruction on mobile phones from panoramic images. In *ISMAR*, pages 55–64, 2011. 1, 2

[9] H. Pfister, M. Zwicker, J. van Baar, and M. Gross. Surfels: surface elements as rendering primitives. In *SIGGRAPH*, pages 335–342, 2000. 3

[10] M. Pollefeys, L. Van Gool, M. Vergauwen, F. Verbiest, K. Cornelis, J. Tops, and R. Koch. Visual modeling with a hand-held camera. *Int. J. Comput. Vision*, 59(3):207–232, 2004. 2

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

[12] 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 *ISMAR*, pages 83–88, 2013. 2

[13] V. A. Prisacariu, O. Kaehler, D. Murray, and I. Reid. Simultaneous 3D tracking and reconstruction on a mobile phone. In *ISMAR*, 2013. 1, 2

[14] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3D model acquisition. In *SIGGRAPH*, pages 438–446, New York, NY, USA, 2002. ACM. 1

[15] A. Sankar and S. Seitz. Capturing indoor scenes with smartphones. In *ACM Symposium on User Interface Software and Technology*, 2012. 2

[16] D. Scharstein and R. Szeliski. A taxonomy and evaluation of dense two-frame stereo correspondence algorithms. *Int. J. Comput. Vision*, 47(1-3):7–42, Apr. 2002. 2

[17] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 519–528, 2006. 2

[18] C. Strecha, W. von Hansen, L. V. Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Anchorage, AK, USA, 2008. 2

[19] J. Stuehmer, S. Gumhold, and D. Cremers. Real-time dense geometry from a handheld camera. In *Pattern Recognition (Proc. DAGM)*, pages 11–20, 2010. 2

[20] P. Tanskanen, K. Kolev, L. Meier, F. Camposeco, O. Saurer, and M. Pollefeys. Live metric 3D reconstruction on mobile phones. In *IEEE International Conference on Computer Vision (ICCV)*, 2013. 1, 2, 6, 7

[21] G. Vogiatzis and C. Hernandez. Video-based, real-time multi-view stereo. *Image Vision Comput.*, pages 434–441, 2011. 2

[22] T. Weise, T. Wismer, B. Leibe, , and L. V. Gool. In-hand scanning with online loop closure. In *IEEE International Workshop on 3-D Digital Imaging and Modeling*, 2009. 2

[23] A. Wendel, M. Maurer, G. Graber, T. Pock, and H. Bischof. Dense reconstruction on-the-fly. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 1450–1457, 2012. 2