Image Selection For Improved Multi-View Stereo

Alexander Hornung Boyi Zeng Leif Kobbelt
RWTH Aachen University
http://www.graphics.rwth-aachen.de

Abstract

The Middlebury Multi-View Stereo evaluation [18] clearly shows that the quality and speed of most multi-view stereo algorithms depends significantly on the number and selection of input images. In general, not all input images contribute equally to the quality of the output model, since several images may often contain similar and hence overly redundant visual information. This leads to unnecessarily increased processing times. On the other hand, a certain degree of redundancy can help to improve the reconstruction in more "difficult" regions of a model.

In this paper we propose an image selection scheme for multi-view stereo which results in improved reconstruction quality compared to uniformly distributed views. Our method is tuned towards the typical requirements of current multi-view stereo algorithms, and is based on the idea of incrementally selecting images so that the overall coverage of a simultaneously generated proxy is guaranteed without adding too much redundant information. Critical regions such as cavities are detected by an estimate of the local photo-consistency and are improved by adding additional views. Our method is highly efficient, since most computations can be out-sourced to the GPU. We evaluate our method with four different methods participating in the Middlebury benchmark and show that in each case reconstructions based on our selected images yield an improved output quality while at the same time reducing the processing time considerably.

1. Introduction

Recent evaluations by Seitz *et al*. [18, 25] of several methods for multi-view stereo (MVS) reconstruction have shown that this field is developing into a promising alternative to other methods for object digitization such as range imaging. However, for all types of image-based methods, the performance in terms of quality and efficiency generally depends significantly on the input data. On the one hand one needs enough measurements for a faithful reconstruction of the 3D object. On the other hand, however, it is as well desirable to minimize the amount of data, since processing overly redundant input increases the overall computing time without improving (or even decreasing) the reconstruction quality. Moreover, special care has to be taken to resolve difficulties inherent to image-based reconstruction methods such as occlusions, or complex surface materials.

These requirements pose a difficult challenge in the 3D reconstruction pipeline, often making a manual intervention by a human operator inevitable. Hence, especially in the field of active range imaging, there has been a lot of effort to automate this process, generally known by the term "Next Best View Planning" (NBV). But although there have been many advances in this field, a lot of practically relevant problems are still considered unsolved (Scott *et al*. [23]).

For MVS the dependency of the reconstruction on the input data is probably even more critical, and many of the involved problems are inherently different to previous work on NBV. For instance, we cannot assume that one measurement, *i.e*., one input image, already generates a sufficiently good partial reconstruction. Instead, a fundamental requirement is that we need at least two input images already for one single reconstruction step. In order to improve the robustness of the reconstruction process with respect to problems such as calibration errors, illumination changes, or image noise and blur, it is often necessary to increase the number of images. Finally, a particular challenge is to sufficiently capture details and features like deep concavities or fine, topologically relevant structures such as holes.

In practice there are two common ways for acquiring input images for MVS reconstruction. The first possibility is to manually control the image acquisition by a human expert, who can identify problematic regions of the object and hence choose the camera positions accordingly. Current automatic acquisition setups are generally using a turn-table or a robot based system, and simply acquire several images of an object on regularly spaced camera positions. Neither of these approaches can guarantee that all relevant parts of the object's surface are captured in a sufficient quality. Hence one often captures an unnecessarily redundant set of input images (up to several hundred). However, while the acquisition of large amounts of images is easy, many recent al-

gorithms cannot process such a high number of images efficiently. At the time of writing of this paper, only 9 out of 25 methods participating in the Middlebury Multi-view Stereo Evaluation [18] provide results for the dense image data sets. Most of them need several hours to compute, or even show a decreasing reconstruction quality. The impact of a proper view selection in terms of quality and speed for MVS from large community photo collections has been recently shown by Goesele *et al*. [5]. This indicates that the question of image selection is a so far mostly untapped resource for optimization of MVS.

The contribution of our work is an analysis of the typical requirements of MVS algorithms and the efficient, GPU-based implementation of a corresponding image selection scheme for improved reconstruction quality and speed. Our particular aims are a guaranteed visibility or *coverage* of each surface region and an adaptive focus on problematic surface regions. We present an algorithm based on an iterative process which evaluates certain quality criteria on the surface of an incrementally updated object proxy. New views are selected such that each image maximizes the quality gain with respect to attributes like surface visibility or standard photo-consistency measures. We employ a stereo-based proxy generation which does not require segmented input images and which ensures a reliable convergence to a faithful geometric approximation of the true object already from a small number of images, since the proxy generation and image selection are tightly coupled into a single optimization process. This leads to a qualitatively superior and more efficient proxy generation than, *e.g*., computing the visual hull from a larger set of segmented images. The proposed method supports a stand-alone implementation as an image *pre-selection* procedure prior to the actual MVS reconstruction as well as an *online* next best view estimation integrated into the MVS reconstruction pipeline. Our quantitative results show that our image selection scheme consistently improves the reconstruction quality and processing time for different classes of MVS techniques based on feature matching and patch expansion [4], surface growing [6], deformable models [7], and volumetric graph-cuts [10].

Related Work Related to our work are methods for next best view planning for active range imaging. In this field, many early methods, *e.g*., Maver and Bajcsy [17], primarily focus on identifying occluded surface regions. Pito [19] describes a method for automatically reconstructing an unknown object. Banta *et al*. [2] incorporate a-priori model knowledge. Klein and Sequeira [12] present ideas for range image quality evaluation on the GPU. For an in-depth survey please refer to Scott *et al*. [23]. However, as argued above, the requirements for NBV in range imaging differ significantly from our problem setting. Furthermore, an unsolved problem emphasized by Scott *et al*. is the lacking efficiency of many methods, which significantly reduces their applicability in practice.

In the context of passive image based reconstruction and MVS, most techniques follow relatively simple image selection heuristics, *e.g*., using k-nearest images. However, a number of authors have developed dedicated selection schemes. Farid *et al*. [3] presented a first set of view selection strategies for multi-view stereo. Kutulakos and Dyer [14] describe a scheme for view point selection based on contours. Marchand and Chaumette [16] present perceptual strategies for scene reconstruction based on structure from motion. A recent example for optimal view point selection based on Kalman filtering has been presented by Wenhardt *et al*. [28]. View planning for a combination of shape from silhouette and shape from structured light has been presented by Sablatnig *et al*. [22]. Rusinkiewicz *et al*. [21] circumvent the necessity for computing views by transferring this task to the user. Vázquez *et al*. [26] consider the problem of automatic view selection for image based rendering. Lensch *et al*. [15] present a GPU accelerated approach for sampling spatially varying BRDFs.

Recently, Goesele *et al*. [5] showed that view selection schemes are almost inevitable when dealing with large image databases. Their method efficiently selects subsets of compatible views by evaluating shared image features. They also point out that view selection combined with widely applied standard metrics is a very effective tool for dealing with a variety of input modalities.

All these above methods solve important problems occurring in image based reconstruction. However, the diverse foci of each of these methods reflect also the inherent problem of selecting optimal input data with respect to the requirements of a specific technique. Our work is motivated by the recent evaluations of numerous MVS methods, such as [4, 5, 6, 7, 20, 27] just to mention a few. Please refer to [18] for a more complete list. To our knowledge, this is the first work on automatic and efficient input optimization addressing the specific requirements of such MVS techniques.

2. Conceptual Overview

In the following we describe the motivation and main concepts behind our approach. Given a (possibly very large) set of calibrated input images of an object, we aim at selecting a subset of images, which sufficiently capture all relevant features without adding too much redundancy.

Finding an optimal image subset is a complex combinatorial optimization problem. Hence, a common approach in the field of NBV planning for range imaging is to use iterative greedy procedures, which are generally based on the following generic work cycle. First the algorithm takes one or more initial measurements (scans), and generates a corresponding geometric proxy. This proxy is then iteratively refined by the optimization procedure. At the beginning

of each iteration, different quality measures are computed over the current surface approximation. These measures describe, for example, the current coverage or measurement certainty. Based on this information the algorithm selects new views for which one expects a maximal quality gain, and updates the surface proxy with this new information. These steps are repeated until a termination criterion with respect to the quality measures is met.

Since we aim at an algorithm supporting MVS, we have to identify the corresponding specific requirements. For recent algorithms we find that the reconstruction accuracy and efficiency depends largely on the following three criteria:

Initial Surface Proxy: Most algorithms either require or iteratively generate an initial proxy of the object as an initialization, *e.g*., for a proper topology and visibility estimation. Furthermore, a faithful proxy minimizes the computation time, since methods based on deformable models evolve the initial proxy to the actual object surface [7]. Volumetric approaches on the other hand perform better, the more voxels are carved away, which are not part of the true surface [10]. With a few exceptions [8], this geometric prior is generally based on segmented input images, *i.e*., the visual hull, since efficiently computing a more accurate stereo-based proxy, in particular from a large set of input images, is a non-trivial problem in practice. Hence, the first important goal of our method is the selection of a small subset of input images, which allow for an efficient generation of a stereo-based proxy that is a good approximation of the unknown object surface, and which is sufficiently covered by the selected images.

Surface Visibility: For MVS reconstruction, every surface point has to be visible in at least two images. In general, however, the reconstruction quality is strongly affected by texture, image noise and blur, calibration errors, as well as illumination effects which are not handled by the employed photo-consistency metric. These problems can be alleviated by capturing redundant data. On the other hand, unnecessary redundancy increases the processing time. Hence an algorithm for image selection should ensure that each surface point is visible for a certain number of cameras with a guaranteed minimum viewing angle, without including unnecessary images. Although this criterion seems similar to the above, the essential difference is that a visibility optimization without a proper initial proxy would potentially lead to a suboptimal selection of images, which focus on incorrectly approximated or even nonexistent surface parts.

Adaptivity: While the above steps guarantee a minimum viewing quality for every surface point on the proxy, they do not ensure a good reconstruction performance in particularly difficult surface regions, where the proxy is only a suboptimal approximation to the real object surface. Typically, this can happen for deep concavities or thin holes through the object, which are difficult to detect and capture properly. As a consequence, methods requiring an initially correct topology of the proxy fail. However, because of the distance of these proxy regions to the true object surface, they can often be characterized by having bad photo-consistency values in the input images. Hence, our algorithm should adapt to the surface reliability by selecting additional images focusing on photo-inconsistent regions.

These criteria exhibit a natural successive order, since each of them relies on the respective previous criterion. So instead of simultaneously optimizing all criteria, these dependencies allow for an efficient iterative optimization procedure consisting of three corresponding phases.

3. Image Selection and Proxy Generation

According to the above criteria, the three main phases of our algorithm are the following: Phase 1 aims at choosing views that support a fast convergence towards an initial geometric proxy. Phase 2 then ensures a sufficient coverage of each point on the proxy surface in at least 2 images, and phase 3 adds additional images focusing on proxy regions with locally bad photo-consistency values.

In each phase, a corresponding quality criterion has to be evaluated on every point of the current surface approximation. Based on this evaluation, the algorithm selects a new image, which is expected to maximally improve this quality criterion. To simplify the problem setting, we use a voxel grid V as a discrete volumetric geometry representation for the proxy. A voxel $v \in V$ can be either full or empty. Initially, we start with all voxels full. In phase 1, we classify all voxels as empty, which can be identified as not being part of the object. The remaining full voxels $S_i \subset \cdots \subset S_0 = V$ represent the iteratively improved object proxy. Full voxels with empty neighbors lie on the current proxy surface and are denoted by ∂S_i. For these voxels we evaluate the quality criteria corresponding to each phase.

In our experiments we found that a medium grid resolution of 128^3 provides the best tradeoff in terms of accuracy and efficiency. Furthermore, we do not store the complete voxel grid, but we rather build an adaptive octree which allows us to prune large empty or full regions. Surface voxels are always enforced to be from the finest resolution level. For these voxels we can then easily compute an estimated normal vector by fitting a regression plane to the local neighborhood of surface voxels.

Depending on the image acquisition setup, the available images can be distributed in the whole embedding space or in some arbitrary sub-region, *e.g*., with viewpoints constrained to lie on a sphere around the object. Our method is not limited to any specific configuration, but can handle arbitrary viewpoints and -directions. The set of camera positions corresponding to the set of images is denoted as C.

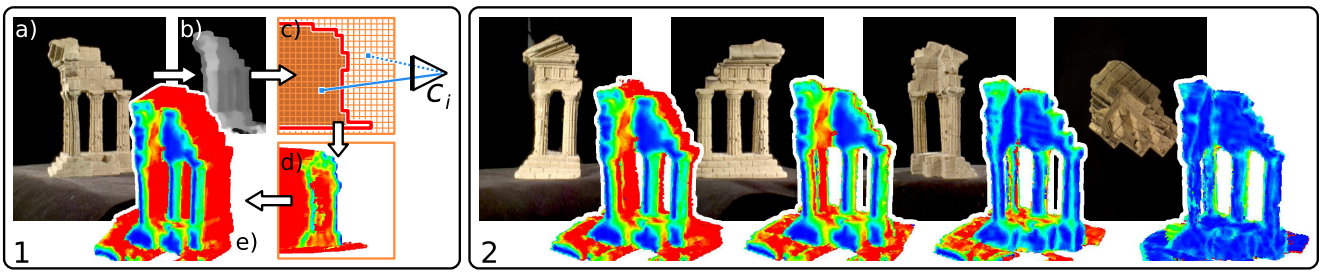

Figure 1. Illustration of phase 1. (1) For each image a) the algorithm computes a depth map b) using a small-baseline stereo method, and carves away all voxels lying in front of the depth map c). For the remaining voxels it then estimates a quality value depending, *e.g.*, on their visibility in the images selected so far d). The color coding visualizes the current state of the proxy (shown from a different viewpoint in e) for illustration purposes). Visible voxels are marked blue, while voxels currently not visible in any image are marked red. (2) With each iteration, the algorithm selects a new image, which maximizes the number of visible voxels, and then iterates steps b)-d).

3.1. Selection Procedure

In the ith step of our algorithm, we have constructed a proxy ∂S_{i-1} from the views $I_1, \ldots, I_{i-1}$ seen from viewpoints $c_1, \ldots, c_{i-1} \in C$. Our goal is to pick a new viewpoint c_i such that adding the corresponding view I_i leads to a maximally improved proxy ∂S_i with respect to the quality criterion of the current phase.

The major problem with this approach is that we cannot predict ∂S_i without actually knowing the new view I_i and integrating it into the current reconstruction. However, for a large number of views, such a tentative integration and evaluation of all possible images I_i is computationally infeasible. Hence, the best we can do is to rate the improvement that the new view would have on the old proxy ∂S_{i-1}. With this approach, all quality criteria can be formulated in terms of the viewpoints c_i only. This allows us to estimate the quality gain for a given image I_i efficiently by rendering the current proxy ∂S_{i-1} as seen from the respective viewpoint c_i, without having to recompute and evaluate the proxy for every image. The next best view is the image which maximizes the number of visible low quality voxels.

Each phase continues as long as the quality gain per iteration stays above a certain threshold. Otherwise we switch to the next phase by changing the quality criterion. The following describes the phases and criteria in detail. Sect. 4 then shows how these criteria can be evaluated efficiently.

Phase 1: Initial Surface Proxy This initial phase aims at guaranteeing that every voxel $v \in \partial S_{i-1}$ of the current proxy is visible in at least one image I_j from an acute viewing angle $\leq \phi$ (Fig. 1) in order to reliably classify it as being inside or outside of the object's photo-hull. The viewing angle is the angle between the surface normal $\mathbf{n}$ of a voxel v and the vector $\mathbf{d}_j = (\mathbf{c}_j - \mathbf{v})/\|\mathbf{c}_j - \mathbf{v}\|$ pointing from v to a camera center c_j. This requirement can be formalized as

$$\forall v \in \partial S_{i-1} \;\; \exists j \in [1,i] : P_\phi(v, c_j) \text{ with}$$
$$P_\phi(v, c_j) : \text{visible}(v, c_j) \wedge \mathbf{d}_j \cdot \mathbf{n} \geq \cos\phi \;\; . \quad (1)$$

The sets of low quality voxels $\widetilde{\partial S}_{i-1}$, $\widetilde{\partial S}'_{i-1} \subseteq \partial S_{i-1}$, which violate this condition before and after the integration of view c_i are defined as

$$\widetilde{\partial S}_{i-1} = \{v \in \partial S_{i-1} : \forall j \in [1, i-1] : \neg P_\phi(v, c_j)\} \text{ and } \quad (2)$$
$$\widetilde{\partial S}'_{i-1} = \{v \in \partial S_{i-1} : \forall j \in [1, i] : \neg P_\phi(v, c_j)\} \;\; . \quad (3)$$

The quality gain g_i in the ith iteration can then be defined as the relative improvement of low quality voxels, *i.e.*,

$$g_i(c_i) = (\#\widetilde{\partial S}_{i-1} - \#\widetilde{\partial S}'_{i-1})/\#\widetilde{\partial S}_{i-1} \;\; , \quad (4)$$

The free parameter to maximize $g_i(c_i)$ (*i.e.*, to minimize $\#\widetilde{\partial S}'_{i-1}$) is the next view c_i among all the candidates $c \in C$.

Maximizing this expression directly would correspond to counting the number of improved voxels, without taking the actual degree of improvement for each voxel into account. Hence we use the weighted improvement of the viewing direction with respect to all previous views $(\mathbf{d}_i \cdot \mathbf{n} - \max_{j \leq i-1}(\mathbf{d}_j \cdot \mathbf{n}))$ in order to increase the robustness and sensitivity of the algorithm. This allows the image selection to focus on proxy regions with the lowest quality first. Furthermore, by taking the minimum of $\mathbf{d}_i \cdot \mathbf{n}$ and $\cos\phi$, this weighted approach does not reward improvements beyond the angle threshold ϕ, which implicitly promotes a sufficient parallax between the input images. The complete functional $g'_i(c_i)$ for the quality gain then is

$$g'_i(c_i) = \sum_{v \in \widetilde{\partial S}_{i-1}} g'_i(v, c_i), \;\; \text{with} \quad (5)$$

$$g'_i(v, c_i) = \begin{cases} \min(\mathbf{d}_i \cdot \mathbf{n}, \cos\phi) - \max\limits_{j \leq i-1}(\mathbf{d}_j \cdot \mathbf{n}) & \text{if } P_\phi(v, c_i) \\ 0 & \text{else} \end{cases} . \quad (6)$$

If the viewpoint maximizing $g'_i(c_i)$ actually leads to an effective improvement $g_i(c_i)$, which is above a prescribed threshold δ, the algorithm adds I_i to the set of images.

We then have to update the proxy $S_{i-1} \rightarrow S_i$ by identifying all voxels, which are outside of the photo-hull as seen from this new view. We achieve this by computing a depth map for image I_i with a variant of the method by Yang and

Pollefeys [29] for real-time, small baseline stereo. As comparison images, our algorithm simply selects 2 images from the whole set of available images, which are closest to I_i and which have the most similar viewing direction. The proxy is then updated by carving away all voxels lying in front of the depth map. Instead of simple carving one could of course apply more sophisticated voting schemes, but in our experiments this approach worked sufficiently well. Silhouettes can optionally be exploited.

Phase 1 terminates if either the percentage of low quality voxels drops under a threshold $\#\widetilde{\partial S}_{i-1}/\#\partial S_{i-1} < \epsilon$, or if the improvement of low quality voxels measured by $g_i(c_i)$ is less than δ. Otherwise, the algorithm continues with iteration $i+1$. In all our experiments the parameters ϵ and δ were fixed to $\epsilon = 0.05$ and $\delta = 0.02$ for all 3 phases.

Please note that there might be low quality surface voxels left in $\widetilde{\partial S}_i$ at the end of phase 1 which have never been visible in any of the images, *e.g.*, the bottom of the Middlebury Dino model. These voxels are excluded from further processing. The result of this phase is a sequence of images, which supports a fast convergence towards a sufficiently covered, faithful approximation of the unknown object. The resulting proxy can optionally serve as an input for subsequent MVS based on, *e.g.*, deformable models.

Phase 2: Surface Visibility After the initial proxy generation in phase 1 is accomplished, we change our quality criterion and add additional images such that each voxel now becomes visible in a user specified number $\kappa \geq 2$ of images with a guaranteed maximal viewing angle $\leq \phi$. The possibility of enforcing visibility of each voxel in more than 2 images generally helps to increase the robustness of the subsequent MVS reconstruction process. The corresponding requirement can be expressed similar to Eq. (1) as

$$\forall v \in \partial S_{i-1} : Q(v), \text{ with } Q(v) : \#\{j \in [1,i] : P_\phi(v, c_j)\} \geq \kappa \ , \quad (7)$$

with the low quality voxels $\widetilde{\partial S}_{i-1}$ analogously defined as

$$\widetilde{\partial S}_{i-1} = \{v \in \partial S_{i-1} : \neg Q(v)\} \ . \quad (8)$$

The computation of the quality gain $g_i(c_i)$ and the termination criterion is identical to phase 1. To promote a more uniform distribution of viewpoints, we start phase 2 with a variable $\kappa' = 2$. Each time the termination criterion is met, κ' is increased by one until $\kappa' = \kappa$.

For the selection of the next view c_i, we again adopt a weighted approach similar to phase 1, which takes the visibility improvement for each surface voxel into account, *i.e.*, a voxel which is visible in only one other view $I_1, \ldots, I_{i-1}$ counts more than a voxel which is already visible in several other views. The corresponding functional is defined analogous to Eq. (5), with a different quality gain $g'_i(v, c_i)$

$$g'_i(v, c_i) = \begin{cases} 1 - \frac{m(v)}{\kappa} & \text{if } P_\phi(v, c_i) \\ 0 & \text{else} \end{cases} \ , \quad (9)$$

where $m(v) = \#\{j \in [1, i-1] : P_\phi(v, c_j)\}$ is the number of views among $I_1, \ldots, I_{i-1}$ in which v is sufficiently visible.

Phase 3: Adaptivity The final phase supports the reconstruction of problematic or topologically important surface regions such as concavities or holes. These regions can often be identified by their bad photo-consistency because of a significant deviation from the true surface, or because of deficiencies of the consistency metric. We found that the reconstruction quality can be considerably improved by integrating additional images focusing on these regions.

Hence, we compute for each voxel $v \in \partial S_{i-1}$ a consistency value $\rho(v)$ using a standard metric based on, *e.g.*, normalized cross-correlation or color variances. These metrics are widely used among recent MVS methods and therefore are a reasonable choice for addressing consistency problems [5]. In our implementation we employ the robust consistency estimation based on voxel supersampling proposed in [11]. Large values of $\rho(v)$ correspond to a high color variance and hence represent a bad photo-consistency. So for each voxel having a value $\rho(v)$ larger than a consistency threshold ψ, the algorithm should guarantee τ additional views from a viewing angle $\theta < \phi$:

$$\forall v \in \partial S_{i-1} : R(v), \text{ with } R(v) : \rho(v) < \psi \vee \#\{j \in [1,i] : P_\theta(v, c_j)\} \geq \tau \ . \quad (10)$$

Please note that the number of additional views τ and the angle θ have an equivalent meaning to κ and ϕ in phase 2. However, our experiments showed that τ can be chosen smaller than κ since one generally needs only a few extra images to improve the reconstruction in problematic regions. ψ obviously depends on the method for measuring photo-consistency. In our implementation we found these parameters to work quite stable for different data sets, so that we could simply keep them constantly set to $\tau = 2, \theta = 30$, and $\psi = 0.7$. Low quality voxels $\widetilde{\partial S}_{i-1}$ are defined as before

$$\widetilde{\partial S}_{i-1} = \{v \in \partial S_{i-1} : \neg R(v)\} \ , \quad (11)$$

and the quality gain $g_i(c_i)$ and termination criteria are again analogous to phase 1. As in the previous phases, we compute a weighted estimate, based on the photo-consistency values, for selecting the image

$$g'_i(v, c_i) = \begin{cases} \rho(v) & \text{if } P_\theta(v, c_i) \\ 0 & \text{else} \end{cases} \ . \quad (12)$$

4. GPU-based Implementation

Efficient GPU-based implementations for small baseline stereo or photo-consistency estimation have already been presented in previous work [29, 11]. The remaining, most time consuming part of our algorithm is the evaluation of the quality criteria in all 3 phases. Practically useful computation times can only be achieved if we manage to evaluate each input image in just a few milliseconds. Remember

that a quality estimate for each single image requires the following steps: (1) check every surface voxel for visibility, (2) estimate the quality gain per voxel (based on the viewing angle or photo-consistency), and (3) accumulate the quality gain over all voxels to estimate the total gain. In order to achieve the required efficiency, we transfer the computation of these steps to the GPU.

Our GPU implementation consists of two main passes. First, we have to determine the visibility of all surface voxels for a given image I_i. This is an operation which can be performed most efficiently on a modern GPU by exploiting the z-buffer. The idea is to simply render all surface voxels as seen from the corresponding viewpoint. This can be easily achieved by setting the projection of the rendering system according to the calibration data of I_i. The GPU takes care that only the nearest (and hence visible) ones will be stored in the frame buffer. For maximum performance we use a splat-based rendering approach [13] by replacing each voxel with a screen-aligned quad located at the center of the voxel. This is achieved by sending one point primitive per voxel to the GPU. In order to render a closed surface, the projected screen-size of each rendered primitive is computed in a vertex shader [24], conforming to the size of its corresponding voxel in the volumetric grid .

Next, we transfer the local quality gain estimation to the GPU by evaluating the corresponding equations (*e.g.*, Eq. (6)) in a fragment shader [24], and encoding the result of the computation in the rendered splat color. There are efficient techniques well-known in the point-based rendering community that allow us to render more than 30 million splats per second including visibility and additional calculations [13], and with the support of recent graphics processors for floating point output buffers we achieve the same computational accuracy as in a CPU based implementation.

Unfortunately, summing up frame buffer pixels for accumulating color-encoded quality gain values and counting the number of improved voxels is not an efficient operation on today's GPUs. We can, however, exploit the color blending functionality to perform the quality gain accumulation. Instead of rendering each voxel to its projected 2D position in the input image as above, we define a frame buffer of size 1×1, and render the required values of each visible voxel into this single pixel. By configuring the rendering pipeline to perform additive blending of the output colors, we achieve the desired accumulation over the proxy surface.

5. Results

In this section we show that our technique for image selection can significantly improve the quality and efficiency for four different classes of MVS techniques.

Table 1 shows results for several experiments on synthetic and real input data with the MVS approach described in [10], into which we integrated our technique as an online procedure for image selection and proxy generation. For each data set we created a reference model from all available images, and then compared reconstructions with different parameter settings from images selected by our algorithm (SEL) vs. uniformly distributed images (UNI).

Model	ϕ	Images	Error UNI	Error SEL	Rel. Improv.
Mouse	45	27	0.35 (4.46)	0.24 (2.82)	30% (37%)
	30	44	0.33 (4.58)	0.24 (2.72)	27% (41%)
CAD	60	23	0.99 (4.81)	0.44 (3.77)	55% (22%)
Scarecrow	45	24	0.62 (5.93)	0.35 (3.84)	44% (35%)
Bahkauv	45	19	1.65 (7.52)	0.75 (5.89)	55% (22%)
	30	26	0.92 (6.33)	0.67 (5.95)	27% (6%)
Temple	60	21	0.60 (3.55)	0.52 (3.20)	13% (10%)
	45	50	0.50 (4.28)	0.42 (2.58)	16% (40%)
Dino	45	41	0.56 (4.37)	0.47 (3.73)	16% (15%)
	30	50	0.53 (4.52)	0.45 (3.28)	15% (27%)

Table 1. Evaluation with the reconstruction technique in [10] for several data sets and parameter settings, showing the RMS & (MAX) Hausdorff distance to the respective reference model.

Thresholds	Matching [4]	Growing [6]	Deformable [7]	Graph-Cuts [10]
80% (mm)	0.43 / 0.41	0.52 / 0.49	0.36 / 0.33	0.64 / 0.59
90% (mm)	0.60 / 0.56	0.90 / 0.66	0.50 / 0.45	1.00 / 0.88
99% (mm)	1.36 / 1.31	1.38 / 1.27	1.11 / 0.83	2.86 / 2.08
0.75 mm (%)	92.1 / 93.2	81.5 / 85.2	95.5 / 97.4	79.5 / 82.9
1.25 mm (%)	97.8 / 97.8	92.3 / 94.2	99.0 / 99.4	90.2 / 93.0
1.75 mm (%)	99.2 / 99.3	95.8 / 97.3	99.8 / 99.6	94.3 / 96.9

Table 2. Middlebury evaluation [18] for four different MVS approaches with 41 uniform / selected ($\phi = 45$) images of the Dino.

In the synthetic experiments (Mouse, CAD) we investigated the performance of the algorithm for different types of features, such as concavities or thin holes. We generated 800 images uniformly distributed around a laser scanned 3D mesh. Since the photo-consistency metric (Phase 3) is based on color variances, we simulated a non-trivial consistency estimation by rendering each model with a white, textureless surface illuminated by a few light sources (Fig. 2), and set parameter $\kappa = 2$ due to the perfect image calibration and noiseless images.

Our experiments with real data were performed with the Middlebury Temple and Dino [25] data set (>300 images each), with 150 images of a Scarecrow model captured using a turntable, and with 290 images of the Bahkauv statue captured with a hand-held camera. Again, the reference reconstructions were generated using all available images. To compensate for calibration errors and other problems like image noise we set $\kappa = 3$ for these experiments.

We then measured the RMS and maximal symmetric Hausdorff distance of the SEL and UNI models to the respective reference model using [1]. Table 1 shows that the reconstruction error is consistently lower for the selected images. Although the numerical improvement sometimes seems relatively small, the visual improvement of the overall shape is often significant (Fig. 2). Especially in cases with a relatively small number of images for complex surfaces, the results are significantly better, *e.g.*, the Scarecrow,

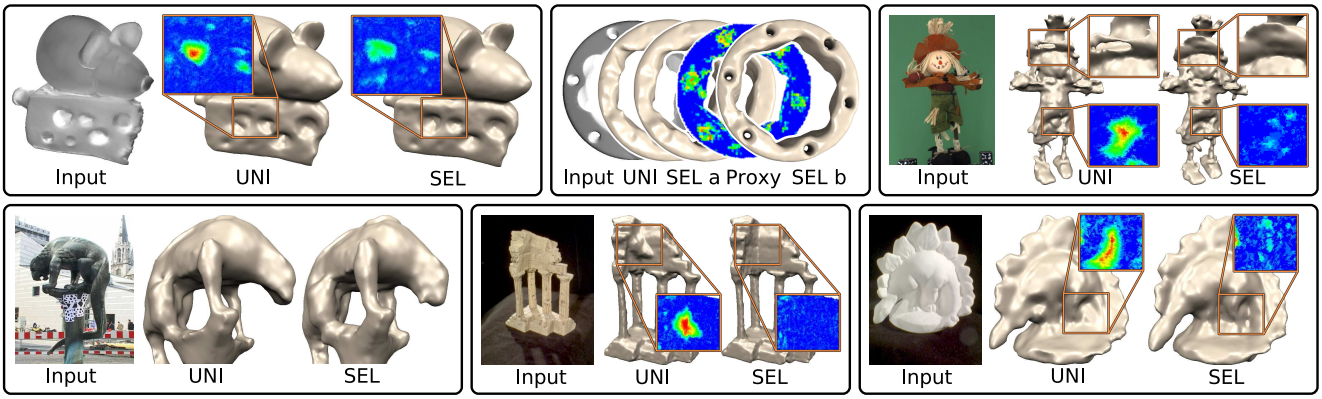

Figure 2. Visual comparison of reconstructed meshes from uniform (UNI) and selected (SEL) input. The color codings show the approximation error to the reference model. In the Mouse experiment, only the selected image set consistently reproduces the deep concavities in the cheese. For the CAD model, the uniform images as well as phases 1&2 (SEL a) fail to capture the thin holes. Phase 3 reveals inconsistent voxels in these proxy regions and selects corresponding images with a significantly improved result (SEL b). Our experiments with real data show consistent visual improvements as well. For instance, our algorithm selects mainly side views and only a small number of top views for the Dino model, and hence captures the head region and the concavities between the legs much better. Also the quite complex Scarecrow and the Bahkauv statue show significant improvements, in particular for difficult features like the Scarecrow's hat.

the Bahkauv, or the CAD model. Reconstructions using selected images even perform better than the uniform data with considerably more images, *e.g.*, UNI Dino, 50 images: 0.53 RMS / 4.52 MAX vs. SEL Dino, 41 images: 0.47 RMS / 3.73 MAX (see also Mouse, Temple, or Bahkauv).

Table 2 presents the Middlebury results for four MVS approaches participating in this evaluation, each representing a fundamentally different class of techniques: feature matching and patch expansion [4], surface growing [6], deformable models [7], and volumetric graph-cuts [10]. We used the 41 selected ($\phi = 45$) and uniform images of the Dino as input images for each method, since this model is generally considered a difficult example because of the missing texture. Please note that the parameter settings used by the corresponding authors were not identical to the ones used in their own Middlebury submission, so that the reconstruction quality might differ a bit [18]. However, the parameter settings used for each technique were identical for the selected and uniform images. The first three rows of Table 2 show the quantitative results of uniform vs. selected views (UNI / SEL) for different accuracy thresholds. The last three rows show the results in terms of completeness. Please see [25] for an in-depth explanation of these thresholds. Again, our selected images consistently produce better results for all methods and thresholds.

The processing time of our algorithm depends on the number of input images and iterations, the proxy resolution, and the number of low quality voxels. However, even for relatively high numbers of images as in the experiments with synthetic data (800), the algorithm needs only 1 to 15 seconds (4 seconds on average) for a single iteration. The quality gain for a single input image is evaluated in about 3 to 20 ms. For all experiments the overall processing time never took more than 1 to 4 minutes, which is negligible compared to the total runtime of most MVS reconstruction algorithms [18]. For instance, using our algorithm in combination with [10], the UNI Temple with 50 images took 15 min. to compute, while the SEL result with 21 images took only 7 min. (including the image selection), with similar reconstruction errors. All presented experiments and measurements were performed on a P4 2.8 GHz with a GeForce 6800 Ultra GPU. Our results (e.g., Table 1) show that reconstructions based on images selected by our algorithm usually have an even higher quality than reconstructions from non-optimized input with up to twice as many images. Considering the fact that the running-time of MVS algorithms is generally dominated by the number of input images, this property helps to considerably reduce processing times, while achieving the same or better output quality.

6. Conclusions

We presented a new image selection technique for MVS reconstruction. Our algorithm specifically addresses the requirements of recent MVS methods, and consistently shows improved performance for four different classes of MVS techniques. The central idea of our method is the definition of three subsequent phases, each of which optimizes specific aspects such as a fast convergence towards a full visual coverage of the unknown object for a fast generation of an initial proxy, guaranteed visibility of the surface with a sufficient quality, and an adaptive focus on uncertain or otherwise critical regions. Due to the integrated stereo based proxy generation, our algorithm does not require any preprocessing of the input images such as segmentation. Moreover, all computationally intensive steps can be executed ef-

ficiently on the GPU.

The numerical and visual evaluation shows that proper image selection is an important, yet currently insufficiently considered resource of optimization in MVS reconstruction. Similar observations have been made in [5]. Our automatic image selection is a step into this direction, and provides ideas for increasing the flexibility and automation of MVS, while at the same time improving the reconstruction quality and performance.

In future work we would like to investigate extensions to our method such as an explicit evaluation and handling of calibration errors, additional entropy based image quality measures, or view selection based on robust statistics [27]. Moreover, techniques based on photometric stereo [9] obviously have requirements different from the standard MVS setting. We believe that investigating image selection for a wider range of techniques is an interesting direction for future work as well.

Acknowledgements

We would like to thank Yasutaka Furukawa, Martin Habbecke, Carlos Hernández, and Daniel Scharstein very much for their support during the evaluation of this work. This project was funded by the DFG research cluster "Ultra High-Speed Mobile Information and Communiation" (UMIC), http://www.umic.rwth-aachen.de/.

References

[1] N. Aspert, D. Santa-Cruz, and T. Ebrahimi. Mesh: Measuring error between surfaces using the hausdorff distance. In *ICME*, volume 1, pages 705–708, 2002.

[2] J. E. Banta, L. R. Wong, C. Dumont, and M. A. Abidi. A next-best-view system for autonomous 3-d object reconstruction. *SMC*, 30(5):589–598, 2000.

[3] H. Farid, S. Lee, and R. Bajcsy. View selection strategies for multi-view, wide-base stereo. Technical Report MS-CIS-94-18, University of Pennsylvania, May 1994.

[4] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. In *CVPR*, 2007.

[5] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. Seitz. Multi-view stereo for community photo collections. In *ICCV*, 2007.

[6] M. Habbecke and L. Kobbelt. A surface-growing approach to multi-view stereo reconstruction. In *CVPR*, 2007.

[7] C. Hernández and F. Schmitt. Silhouette and stereo fusion for 3d object modeling. *CVIU*, 96(3):367–392, 2004.

[8] C. Hernández, G. Vogiatzis, and R. Cipolla. Probabilistic visibility for multi-view stereo. In *CVPR*, 2007.

[9] C. Hernández, G. Vogiatzis, and R. Cipolla. Multi-view photometric stereo. *to appear in PAMI*, 2008.

[10] A. Hornung and L. Kobbelt. Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding. In *CVPR*, volume 1, pages 503–510, 2006.

[11] A. Hornung and L. Kobbelt. Robust and efficient photo-consistency estimation for volumetric 3d reconstruction. In *ECCV*, pages 179–190, 2006.

[12] K. Klein and V. Sequeira. The view-cube: An efficient method of view planning for 3d modelling from range data. In *WACV*, pages 186–191, 2000.

[13] L. Kobbelt and M. Botsch. A survey of point-based techniques in computer graphics. *Computers & Graphics*, 28(6):801–814, 2004.

[14] K. Kutulakos and C. Dyer. Recovering shape by purposive viewpoint adjustment. *IJCV*, 12(2-3):113–136, 1994.

[15] H. Lensch, J. Lang, A. M. Sa, and H.-P. Seidel. Planned sampling of spatially varying BRDFs. *Computer Graphics Forum*, 22(3):473–482, 2003.

[16] E. Marchand and F. Chaumette. Active vision for complete scene reconstruction and exploration. *PAMI*, 21(1):65–72, 1999.

[17] J. Maver and R. Bajcsy. Occlusions as a guid for planning the next view. *PAMI*, 15(5):417–433, 1993.

[18] Middlebury evaluation. http://vision.middlebury.edu/mview/, Mar. 2008.

[19] R. Pito. A solution to the next best view problem for automated surface acquisition. *PAMI*, 21(10):1016–1030, 1999.

[20] J.-P. Pons, R. Keriven, and O. Faugeras. Modelling dynamic scenes by registering multi-view image sequences. In *CVPR*, volume 2, pages 822–827, 2005.

[21] S. Rusinkiewicz, O. Hall-Holt, and M. Levoy. Real-time 3d model acquisition. In *SIGGRAPH*, pages 438–446, 2002.

[22] R. Sablatnig, S. Tosovic, and M. Kampel. Next view planning for a combination of passive and active acquisition techniques. In *3DIM*, pages 62–69, 2003.

[23] W. R. Scott, G. Roth, and J.-F. Rivest. View planning for automated three-dimensional object reconstruction and inspection. *ACM Comput. Surv.*, 35(1):64–96, 2003.

[24] M. Segal and K. Akeley. The OpenGL Graphics System: A Specification (Version 2.1). http://www.opengl.org, 2006.

[25] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR*, volume 1, pages 519–526, 2006.

[26] P.-P. Vázquez, M. Feixas, M. Sbert, and W. Heidrich. Automatic view selection using viewpoint entropy and its application to image-based modelling. *Computer Graphics Forum*, 22(4):689–700, 2003.

[27] G. Vogiatzis, C. Hernández, P. H. S. Torr, and R. Cipolla. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *IEEE PAMI*, 29(12), 2007.

[28] S. Wenhardt, B. Deutsch, J. Hornegger, H. Niemann, and J. Denzler. An information theoretic approach for next best view planning in 3-d reconstruction. In *ICPR*, pages 103–106, 2006.

[29] R. Yang and M. Pollefeys. A versatile stereo implementation on commodity graphics hardware. *Journal of Real-Time Imaging*, 11(1):7–18, 2005.