Refinement of Surface Mesh for Accurate Multi-View Reconstruction

Radim Tyleček and Radim Šára

Center for Machine Perception, Faculty of Electrical Engineering, Czech Technical University in Prague, Czech Republic

***Abstract*—In this paper we propose a pipeline for accurate 3D reconstruction from multiple images that deals with some of the possible sources of inaccuracy present in the input data. Namely, we address the problem of inaccurate camera calibration by including a method adjusting the camera parameters in a global structure-and-motion problem, which is solved with a depth map for representation that is suitable to large scenes.**

Secondly, we take the triangular mesh and calibration improved by the global method in the first phase to refine the surface both geometrically and radiometrically. Here we propose surface energy which combines photoconsistency with contour matching and minimize it with a gradient descent method. Our main contribution lies in effective computation of the gradient that naturally regularization and data terms by employing scale space approach. The results are demonstrated on standard high-resolution datasets and a complex outdoor scene.

***Index Terms*—Structure from motion, dense 3D reconstruction, multi-view, mesh refinement**

I. Introduction

The development of methods for 3D reconstruction from multiple images has led to a number of successful methods, which can be used to construct virtual worlds. They belong to the group of multi-view stereo (MVS) algorithms [1], [2], [3], [4]. Despite the effort and availability of high-resolution images, their performance is still not satisfying when we compare them to laser range measurement systems [5]. The fact that high-resolution images can be easily obtained by consumer cameras or downloaded from the web is a motivation for improving the results of MVS algorithms, especially when the time and hardware costs of range scanning are considered.

Traditionally, evaluation is performed in the terms of *completeness* and *accuracy* [6]. Keeping in mind that these two views still share a wide base, we will focus on the second one in this paper, and propose a pipeline that deals with some of the possible sources of inaccuracy in image-based reconstruction.

The paper is organized as follows. First, sources of inaccuracy and related work are analyzed in Section II. Then the proposed reconstruction pipeline is presented in Section III and its two parts, depth map fusion and subsequent mesh refinement, are detailed in Sections IV and V. Finally, experimental validation is given in Section VI. One of the results of our pipeline is displayed in Figure 1.

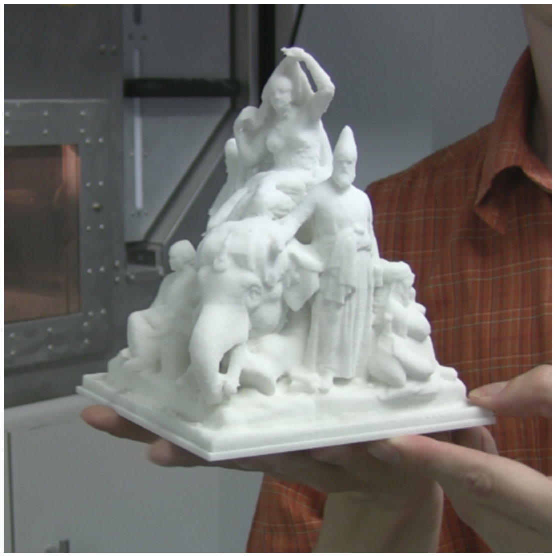


Fig. 1. A replica of the Asia statue produced with rapid prototyping from a 3D model created by our pipeline from 238 input images.

II. Analysis and Related Work

What are the possible sources of inaccuracy in the results of image-based reconstruction methods? How can they be handled? We will offer answers to such questions in the light of existing attempts and include our proposals.

First, we will deal with the **representation** used in the reconstruction process. Accuracy of volumetric [7] and related level-set [8] representations is limited by voxel size, with computational cost increasing typically with $O(N^3)$, which is a high price even when reduced with quad-tree optimizations. Although depth map representations [9] live in the data domain and naturally use the scale of the scene defined by the input images, they have difficulties in modeling parts of surfaces which are observed under small angles. This is a consequence of non-intrinsicity of such representation. Sampling the image space with non-regular image grid could compensate this effect. The representation with rectangular patches [10] does not posses the limitations above, but the connectivity of the surface must be modeled explicitly, i.e. in image space. Finally, triangular mesh representation is often required as the output for visualization or realization, therefore all of the above alternatives are converted to it at some point. The knowledge of connectivity allows direct computation of geometric surface properties, like curvature. In this light the mesh representation

This work was supported by CTU Prague project CTU0912713, by EC project FP6-IST-027113 eTRIMS and by Czech Ministry of Education project MSM6840770012.

Manuscript received on December 1, 2009

Email: tylecr1@cmp.felk.cvut.cz

turns out to be the most useful for final improvements of accuracy. We can also benefit from the experience of computer graphics with this representation [11].

Triangular mesh as a discrete implementation of a continuous surface requires proper topology to effectively sample it, including both density of vertices and the triangulation. Without adaptation of the topology to geometry of the surface, further improvement of accuracy can be impossible: details require a fine mesh. In contrary, for flat regions this would be not efficient. While the problem has been studied from the geometrical point of view [12], [13], image-based optimization of the topology could improve the efficiency even further.

We will turn to the inaccuracy present in the data, in our case **images and camera calibration**. The camera parameters provided have been typically estimated with algorithms that work with sparse correspondences [14] and can be further improved with dense data by one of the following methods. Furukawa [1] has recently applied this approach, when he iteratively switches between the camera calibration with a standard bundle adjustment and running MVS with the refined camera parameters. MVS used in [10] performs final mesh refinement in order to overcome the change of representation from patch-based. Tylecek [15] incorporated this problem differently by solving jointly for both depths and camera centers in depth map framework. Both papers show that the calibration refinement is essential for recovery of fine details, therefore we include this step in our refinement pipeline. Turning back to representation, these methods demonstrate that camera calibration update is easily tractable with image based (depth maps or patches) representations, while other are not suitable for this purpose, i.e. with volumetric or mesh representations the update would be difficult. We have chosen the second method because it is more compact.

Camera lenses, especially on the consumer level, introduce a number of **distortions** in the image, which should be compensated when thinking of accuracy. For example, the strongest radial distortion can reach 20 pixels for 10 Mpix camera with a zoom lens. Compensation of radial distortion is possible with commercial software based on the lens model, but still there can be variations among different exemplars and individual calibration is desirable. Also the level when the standard pin-hole and radial distortion models are limiting the accuracy might be reached soon, resulting in the need for replacing it with a more complex one.

The last group of accuracy limits is related to **photometry** and the acquisition of the images. The Lambertian reflectance model performs well with a number of surfaces, but its deviations become important when comparing different views in detail. While the sources of light are unknown in most situations, explicit modeling with BRDF has too many free parameters. Existing methods extending the reflectance model therefore compensate non-Lambertian effects indirectly, either on the surface [16] or in the image domain [17]. Such approximations can also compensate the effect of different lighting or exposure conditions. We propose a simple image correction which handles them.

The following sections present a reconstruction pipeline that takes into account the analysis above and includes the mentioned methods for improvement of the resulting accuracy. The key idea is in first using a global method to improve calibration and obtain possibly inaccurate estimate of the surface, represented as a set of depth maps, which is followed by change of representation to a 3D mesh that allows local approach to variational correction of its vertices. We focus on efficient computation of the surface flow that naturally balances regularization and data terms by employing scale space approach to find the correct local minimum. Additionally, we include novel formulation of contour matching term in our measure of photoconsistency.

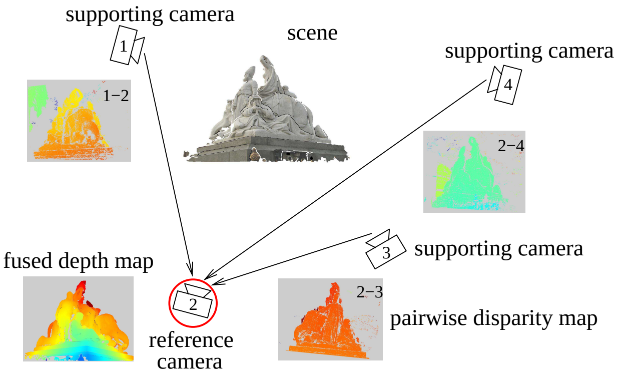


Fig. 2. Idea of Depth Map Fusion. Information from pair-wise disparity maps is fused in a set of reference cameras.

III. Reconstruction Pipeline for High Accuracy

Taking the above analysis into account, our idea is first to use a global structure-and-motion method [15] to obtain inaccurate estimate of the surface, represented as a set of depth maps and simultaneously improve the calibration. We revisit this method in Section IV.

Once we have a fair estimate of the surface, we undergo a change of representation to a 3D mesh that allows a local approach to variational correction of vertices. This approach allows us to introduce details by evaluating the photoconsistency of the surface in Section V.

IV. Depth Map Fusion

This section presents the main points of a method for surface reconstruction based on depth map fusion, which simultaneously refines camera positions [15].

The input to the algorithm is a set of images $\mathcal{I} = \left\{ I_p^i \mid i = 1, \ldots, N;\, p = 1, \ldots, N^i \right\}$, where N is the number of cameras and N^i is the number of pixels in image i. The possibly inaccurate camera calibration $\mathcal{P} = \left\{ \mathbf{P}^i \mid i = 1, \ldots, c \right\}$ is obtained by a robust structure and motion algorithm [14]. Once the geometry is obtained, camera pairs for pair-wise stereo matching are automatically selected in a way that the both cameras are close both in position and view direction, see Figure 2. Disparities of rectified image pairs are then computed with a publicly available dense matching stereo algorithm GCS [18]. The resulting point cloud $\mathcal{X}$ is triangulated in the form of pair-wise disparity maps back-projected to space.

Bayesian estimate of depth maps $\Lambda = \left\{ \lambda_p^i \mid i = 1, \ldots, c; p = 1, \ldots, n^i \right\}$ is then found, where

$\lambda_p^i \in \mathbb{R}$ is a reconstructed depth in pixel p of image i, along with visibility maps $V = \left\{ v_p^i \mid i = 1, \ldots, c; p = 1, \ldots, n^i \right\}$, where $v_p^i \in \{0, 1, 2\}$ is the visibility of pixel p of image i in all cameras $i = 1, \ldots, c$ such that $v_p^i = 0$ marks invisible and non-zero v_p^i visible pixels. The task leads to the maximization of the posterior probability, which can be formally written as

$$(\Lambda^*, V^*, \mathcal{C}^*) = \arg\max_{\Lambda, V, \mathcal{C}} P(\Lambda, V, \mathcal{C} \mid \mathcal{X}, \mathcal{I}). \tag{1}$$

The output of the algorithm is the structure Λ^*, V^* together with adjusted camera calibration $\mathcal{C}^*$. The algorithm alternates between two sub-problems conditioned on each other: estimation of $(\Lambda, \mathcal{C})$ and V. The output of the first subproblem is used as the input to the second, and vice versa.

Firstly, a subset of cameras where the depth and visibility maps will be estimated is manually chosen. Then visibility maps $V(0)$ and depths maps $\Lambda(0)$ are initialized from projection of input data $\mathcal{X}$ to the images $i = 1, 2, \ldots, c$.

The procedure of visibility and depth estimation alternates the following steps.

1) In the **visibility estimation** task the estimate of visibility V^* is obtained from

$$V^* = \arg\max_{V} P(\mathcal{I} \mid V, \Lambda, \mathcal{X})\, P(\Lambda \mid V, \mathcal{X})\, P(V), \tag{2}$$

where the image likelihood $P(\mathcal{I} \mid V, \Lambda, \mathcal{X})$ makes a non photoconsistent surface less probable, i.e. where the image intensities of projections to corresponding cameras do not match. The depth map probability $P(\Lambda \mid V, \mathcal{X})$ assumes locally flat surface and penalizes visibility of high depth variations (outliers, discontinuities). The prior on $P(V)$ favors compact visible regions in images. The solution of this task is found by a minimum cut algorithm.

2) In the **depth estimation** task the estimate of depths is obtained from

$$(\Lambda^*, \mathcal{C}^*) = \arg\max_{\Lambda, \mathcal{C}} P(\mathcal{X} \mid \Lambda, \mathcal{C}, V)\, P(\Lambda, \mathcal{C}, V), \tag{3}$$

where $P(\mathcal{X} \mid \Lambda, \mathcal{C}, V)$ builds projection constraints to minimize distance between visible data points $\mathcal{X}$ and points corresponding to the depth maps Λ, while camera centers $\mathcal{C}$ are also free parameters. The prior $P(\Lambda, \mathcal{C}, V)$ is represented by a second-order surface model enforcing curvature consistency. This task leads to a system of linear equations, which is solved by quasi-minimal residual method. The solver can use two ways to reduce the discrepancy in data: either by adjusting the estimated depths Λ or moving the camera centers $\mathcal{C}$. Bundle adjustment methods [1] apply a similar approach, but this solution goes further by exploiting dense data and including a surface model.

When the iterative procedure converges and a consistent set of depth maps is obtained, depth maps are projected to 3D space to obtain a cloud of points with normals estimated from points neighboring in the image.

Finally, the points are merged into continuous surface with PSR [19] and the result is filtered to remove introduced big triangles based on average edge size.

V. Mesh Refinement

While the depth map representation in image space is useful for large scenes and natural to the input data, it has limits for modeling arbitrary surfaces as it is not intrinsic to them. A change of representation is thus required for further improvement of the surface accuracy. The global method [15] provides us with a good initial estimate of the surface, represented by a discrete triangular mesh, and a refined camera calibration. We choose this mesh as a surface model and represent it as a set of vertices $\mathbf{X}_i \in \mathbb{R}^3, i = 1, \ldots, n_X$ and triangle indices $\mathbf{T}_j \in \{1, \ldots, m\}^3, j = 1, \ldots, n_T$.

For the purpose of deriving our method, we will start with continuous definition, and later discretize the results. In this task, our goal will be to find the estimate of surface S by the minimization of a surface energy E_ϕ:

$$E_\phi(S) = \int_S \phi(\mathbf{X}) dA, \tag{4}$$

where $\phi(\mathbf{X})$ is a photoconsistency measure and dA is surface element. Since we assume a good initial estimate of the surface S, we can resort in our method to implicit regularization of the surface based on the minimal surface area.

The primary goal in multi-view reconstruction is to find a surface with photoconsistent projections to multiple images.

Photoconsistency is efficient when a given surface point is seen in close-to-normal direction, where non-Lambertianity is not critical. Close to occluding contours even the Lambertian model breaks, but here we can exploit contour matching.

In the following sections we will combine these two sources to construct ϕ and next propose a method for its minimization.

A. Photoconsistency measure

We define a photoconsistency function ϕ_I for a given world point $\mathbf{X}$ and a set of images $I_i, i = 1, \ldots, N$ in the following way:

$$\phi_I(\mathbf{X}) = \frac{1}{|V(\mathbf{X})|} \sum_{\substack{i,j \in V(\mathbf{X}) \\ i \neq j}} \frac{2\|I_i(\pi_i(\mathbf{X})) - I_j(\pi_j(\mathbf{X}))\|^2}{\sigma_i^2(\pi_i(\mathbf{X})) + \sigma_j^2(\pi_j(\mathbf{X}))} \tag{5}$$

where $V(\mathbf{X})$ is a set of images in which point $\mathbf{X}$ is visible, and $\pi_i(\mathbf{X}) \simeq \mathbf{P}_i\mathbf{X}$ is the perspective projection function ($\mathbf{P}_i$ is a camera matrix). The normalizing factors $\sigma_{i,j}$ are independently pre-computed variances of image functions $I_{i,j}$ in visible regions and they estimate expected measurement error assuming Poisson distribution of the image values. They allow the range of the measuring function to be $\phi \in \langle 0, 1 + \epsilon \rangle, \epsilon \geq 0$, unless the mean intensities differ wildly. Our resulting measure is thus a *normalized sum of squared differences* (NSSD). As pointed out in [2], we avoid the use of normalized cross correlation (NCC), which introduces additional ambiguities.

The traditional Lambertian assumption allows us to use simple difference of pixel intensities, unfortunately this model is often violated, for instance, the exposure parameters are often different in available input images. Since modeling of reflectance properties is complex, i.e. with radiance tensors [16], we will limit ourselves to intensity offset correction. We will thus use the knowledge of current shape to estimate the 'true'

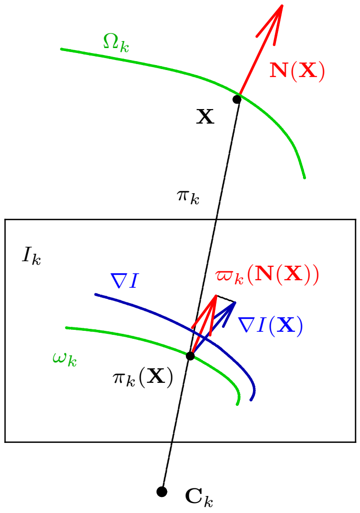


Fig. 3. Demonstration of a suboptimal situation in contour matching, where the image edge (local maxima of image gradient, ∇I) does not correspond to the projected contour ω_k.

offset-corrected images $I_i^* = I_i - C_i$ which minimize the total error (5) by choosing the offset C_i to be the mean radiance error of the surface visible in camera i:

$$C_i = \frac{1}{N_i} \sum_{j \,|\, i \in V(\mathbf{X}_j)}^{N_i} \Big(I_i(\pi_i(\mathbf{X}_j)) - \bar{I}(\mathbf{X}_j) \Big), \tag{6}$$

where N_i is the number of vertices X visible in camera i and $\bar{I}(\mathbf{X})$ is the mean of the projections of point $\mathbf{X}$ to images where it is visible:

$$\bar{I}(\mathbf{X}_j) = \frac{1}{|V(\mathbf{X}_j)|} \sum_{i \in V(\mathbf{X}_j)} I_i(\mathbf{X}_j), \tag{7}$$

being the best estimate of radiance with respect to the square error in (5). The $\bar{I}(\mathbf{X}_i)$ is also used as consistent surface color for texturing the neighborhood of point $\mathbf{X}_j$.

Once we have obtained all offsets $C_i, i = 1, \ldots, N_i$, we can use them to recompute the mean $\bar{I}$ from corrected images I^* and iteratively improve our estimate of the correcting offsets.

Now we can replace original images I_i with corrected I_i^* in all our image terms derived from (5).

B. Contour matching

The analysis of [20] has first brought the observation that projection of contour generators on a smooth surface should match local maxima of image gradient ∇I (apparent contours), which has recently been an inspiration for [17], [21]. Similarly to [21] we avoid explicit detection of contours in images by a more general formulation, but we additionally take into account the directions of ∇I and surface normals $\mathbf{N}$ projected to the image, see Figure 3. It is formalized by maximization of a contour matching function $\phi_C(\mathbf{X})$:

$$\phi_C(\mathbf{X}) = \frac{1}{|\Omega(\mathbf{X})|} \sum_{k \in \Omega(\mathbf{X})} \Big| \Big\langle \nabla I\big(\pi_k(\mathbf{X})\big), \varpi_k\big(\mathbf{N}(\mathbf{X})\big) \Big\rangle \Big|, \tag{8}$$

where $\varpi_k\big(\mathbf{N}(\mathbf{X})\big) = \frac{\pi_k(\mathbf{N}(\mathbf{X}))}{\|\pi_k(\mathbf{N}(\mathbf{X}))\|}$ is a unit normal projected to the image and $\langle \cdot, \cdot \rangle$ is a scalar product. We denote here $\Omega(\mathbf{X})$ as the set of cameras that see $\mathbf{X}$ as a contour point. Inversely, for a given camera k, we can find contours Ω_k on the surface S as curves, where normal $\mathbf{N}(\mathbf{X})$ of each of its visible points is perpendicular to the viewing direction $\mathbf{X} - \mathbf{C}_k$:

$$\Omega_k = \big\{ \mathbf{X} \,|\, \big\langle \mathbf{N}(\mathbf{X}), \mathbf{X} - \mathbf{C}_k \big\rangle = 0, k \in V(\mathbf{X}) \big\}\,, \tag{9}$$

where $\mathbf{C}_k$ is the camera center. On discrete meshes, we identify contour vertices by a change of sign of the dot product above and a simultaneous change of visibility. Now we can partition surface points in the following sets for every camera k: V_k – set of points visible in camera k, $\bar{V}_k$ – set of points not visible in camera k and Ω_k – points generating contour in camera k.

To adapt our method for large datasets, we limit the size of V_k by choosing only a given number of the best views based on the angle between the normal and view direction, calculated from the dot product in (9).

We can now put together photometric and contour measures in

$$E_\Omega(S) = \int_S \Big(\phi_I(\mathbf{X}) - \alpha \phi_C(\mathbf{X}) \Big) dA = \int_S \phi(\mathbf{X}) dA, \tag{10}$$

where $\phi_I(\mathbf{X})$ is integrated in cameras $k \in V(\mathbf{X})$ and $\phi_C(\mathbf{X})$ in $k \in \Omega(\mathbf{X})$. Parameter α controls the preference between contour and image matching; we used $\alpha = 1$ in our experiments.

C. Gradient-based approach

According to [22, p. 22], we can obtain a surface flow that minimizes the energy (4) by

$$\frac{\partial S}{\partial t}(\mathbf{X}) = \Big(H(\mathbf{X}) \phi(\mathbf{X}) - \langle \nabla \phi(\mathbf{X}), \mathbf{N} \rangle \Big) \mathbf{N}, \tag{11}$$

where $H(\mathbf{X})$ is the mean surface curvature at point $\mathbf{X}$. The solution S^* is found by Euler's time integration of (11), hence deforming the surface by

$$\mathbf{X}_{t+dt} = \mathbf{X}_t + dt \frac{\partial S}{\partial t}(\mathbf{X}_t), \tag{12}$$

where dt is a chosen time step.

The first part of the flow (11) performs implicit regularization, for $\phi(\mathbf{X}) \to 1$ this flow corresponds to *mean curvature flow*, which leads to minimization of surface area. In our flow this applies to areas with high photometric error. On the other hand, low error $\phi(\mathbf{X}) \to 0$ has no effect. This kind of balancing between regularization and data gets around the shrinking effects of pure surface minimization present in many variational methods. The second part of (11) moves the surface along surface normal $\mathbf{N}(\mathbf{X})$ in the direction where $E(S)$ will decrease, which can be calculated by taking the negative projection of the gradient to the normal movement direction. For regions with missing data (vertices $\mathbf{X}_0$ visible in less than two views), the minimal surface should be the optimal solution, which is accomplished by setting $\phi(\mathbf{X}_0) = 1$.

We compute the directional derivative $\langle \nabla \phi(\mathbf{X}), \mathbf{N} \rangle$ by sampling points $\tilde{\mathbf{X}}(a), a \in \langle -\tau, \tau \rangle$ along the normal in images I^* for $k \in V(\mathbf{X})$ or in the image gradient ∇I^* for $k \in \Omega(\mathbf{X})$ and computing $\phi(\tilde{\mathbf{X}}(a))$, like in Figure 4. At this point we discretize the problem by computing the energy integral (10) only in the vertices $\mathbf{X}_i$ of the mesh, so the photoconsistency is evaluated in individual mesh vertices and no image neighborhood is used. We use this simplification efficiently with mesh resampled so that the mean of edge projection to images is around 2-3 pixels.

In order to avoid falling to a local minimum, the derivative is computed from a quadratic polynomial $\phi'(\tilde{\mathbf{X}}(a)) = p_1 a^2 + p_2 a + p_3$ fit to the samples. In order to perform with a limited number of samples, the window specified by τ is gradually decreased in iterations: $\tau_t = \tau_0 \gamma^{t-1}$, where t is iteration, $\gamma = 0.95$ is the decrease rate, and τ_0 is the initial window size determined from average edge sizes around vertex $\mathbf{X}$. This means that in the first iterations the decision is based on a wider support and allows us to find a global minimum in the initial window. In later iterations the region near this minimum is sampled more densely, producing a more precise estimate.

This can also be thought of as regularizing the problem (11) with a scale determined by the window size. When computing a gradient from the initial large window, the curve cannot fit the data exactly and is rather flat, resulting in a smaller gradient and greater smoothing. The data weight is increased as the window size decreases, when the fitted curve gets steeper and the gradient magnitude is greater. Window size control is more natural than explicitly adjusting the second term in (11) with a constant increasing over iterations: If there is no strong minimum (i.e. in noisy conditions) in the latter method, the gradient will not increase and the model will not over fit there.

VI. Experiments

First, we have evaluated our method on four high-accuracy datasets from a publicly available benchmark [5], which allows comparison of the results with a number of other state-of-the-art methods both in quantitative and qualitative ways, by analyzing occupancy histograms and diffuse renderings. The original results of the depth map fusion [15] were taken as the input for the mesh refinement procedure. In all cases, the algorithm was run for 30 iterations, when the window size τ drops to 20% of the initial size.

The quantitative evaluation in [5] was performed with ground truth acquired with time-of-flight laser measurement. Evaluated scene is projected to the input cameras and obtained depths are compared with the ground truth depths in the scale of their accuracy σ, which is shown in Figure 5 for *fountain-P11* dataset. More results are available on the benchmarking website[1]. The results of refinement ('our') show relative increase of accuracy from initial depth map fusion ('TYL') output by 2.1% at $\sigma \leq 2$. Use of this score for direct comparison of accuracy with other methods is difficult, since we are here evaluating our surface very close to the accuracy limit of the ground truth (σ is the measurement variance). Also the result depends substantially on the completeness of the surface, i.e. the currently best-scoring method [4], which combines the best of several previous methods, succeeds in reconstructing the ground plane of *fountain-P11*, which adds to all bins of the histogram in Figure 5. Still, [4] misses the camera calibration adjustment, and thanks to this feature our method is able to achieve higher accuracy in certain areas, like in Figure 7 g), h) and i), while the error is distributed evenly over the surface in Figure 6 c).

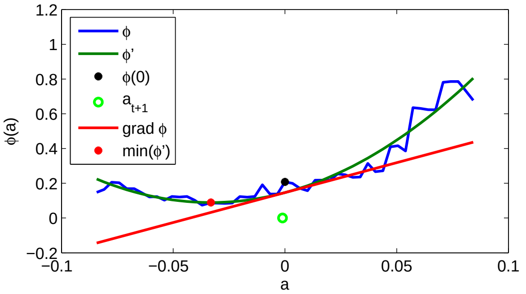


Fig. 4. Photoconsistency $\phi(\tilde{\mathbf{X}}(a))$ sampled in the normal direction with curve ϕ' fitted to it. The $a = 0$ corresponds to the current vertex position.

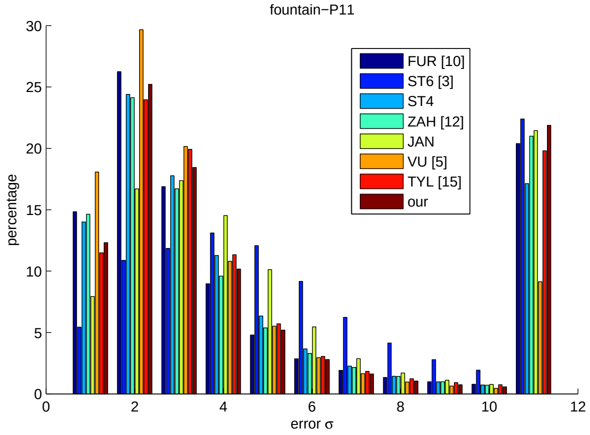
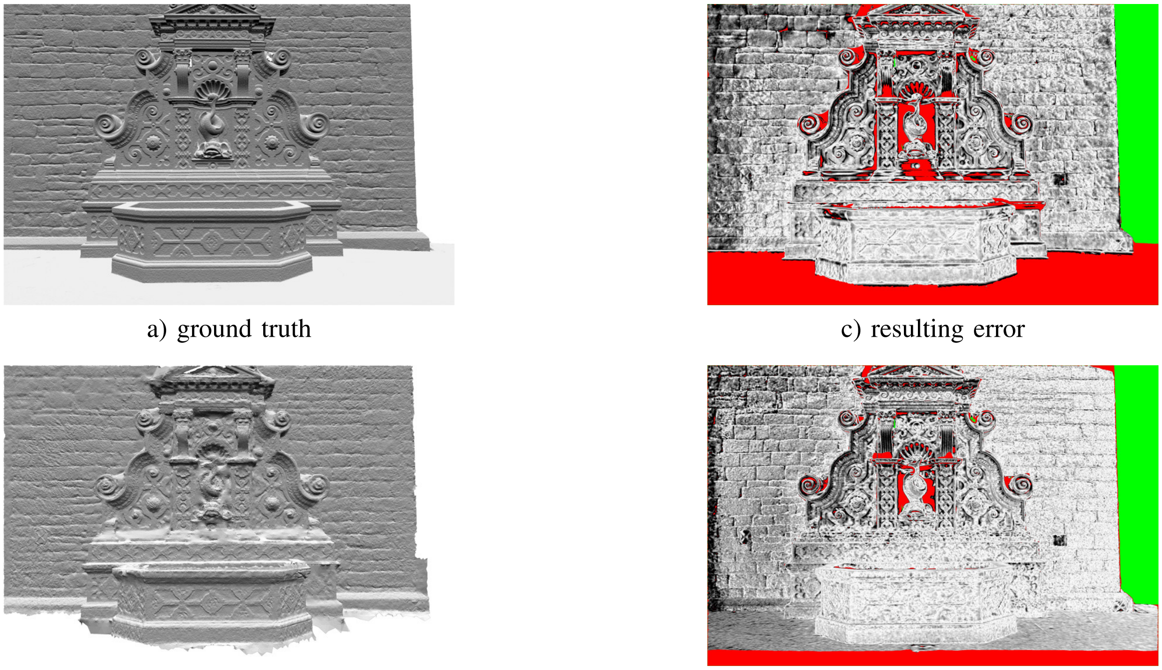


Fig. 5. Histogram from [5], each bin represents percentage of pixels with an error equal to $n\sigma$. Accuracy results in higher values in bins $n = 1, 2$.

The quantitative evaluation does not take into account the visual quality of the surface. Although the estimated surface may be close to the ground truth, the human observer is influenced by regularity or smoothness of the surface, e.g. when resulting 3D models are used for visualization. For this purpose, comparison of surface normals would be appropriate, but while it is not included in [5], we will use the renderings in its place. Figure 7 presents results in this way and shows how the initial result of depth map fusion in c) was improved by the refinement in d) with flat surfaces are smoothed and edges emphasized. Here similar results of the best performing state-of-the-art methods [10], [4] in e) and f) still show notable roughness.

In order to evaluate the effect of individual contributions to the accuracy of the proposed method, we have run it with different modifications on the *fountain-P11* dataset. The results can be compared visually in detail in Figure 8. The importance of the contour matching term is demonstrated on the difference between a) and d), where the edges become bumpy. It can

[1] http://cvlab.epfl.ch/ strecha/multiview/denseMVS.html



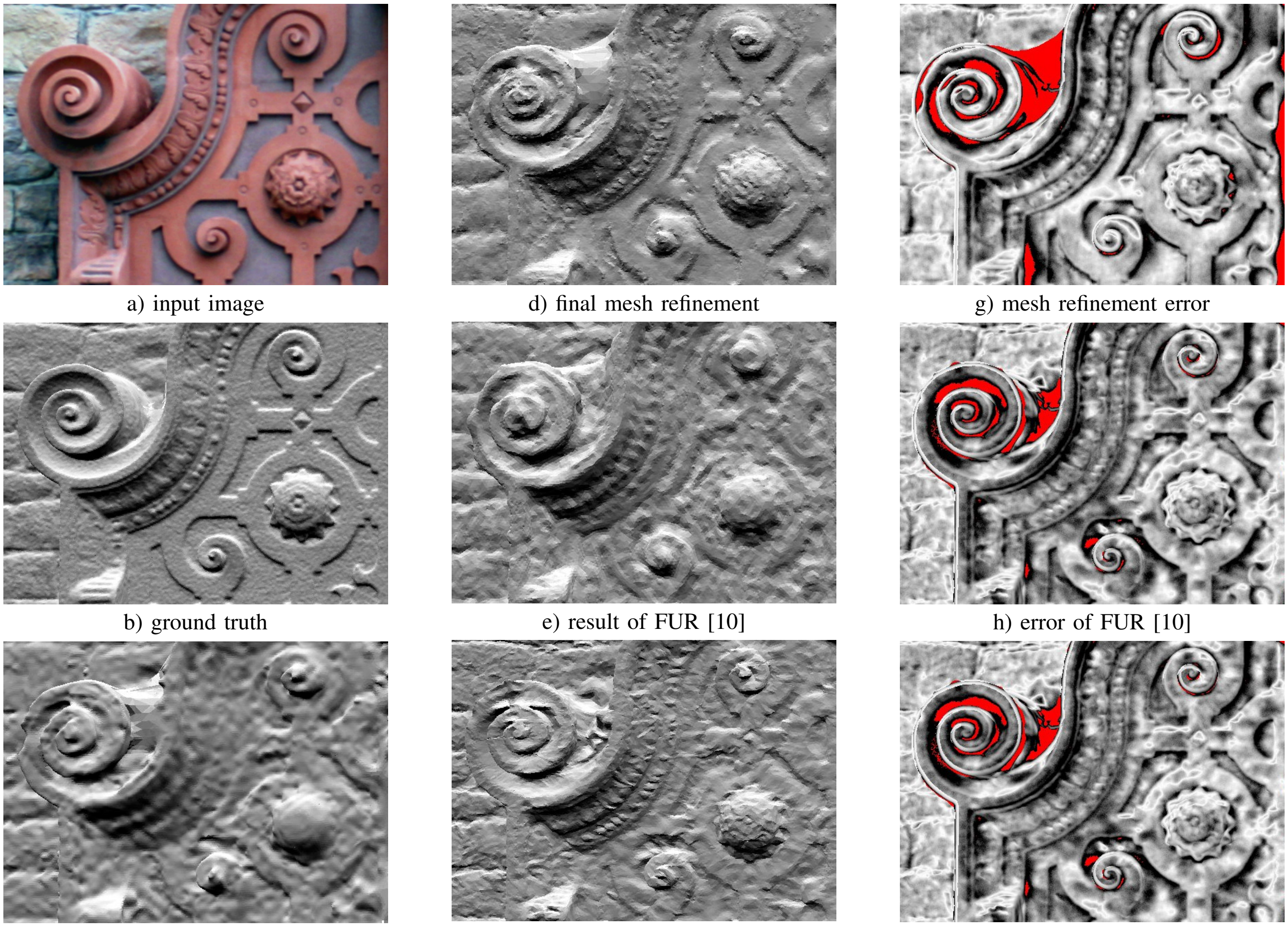
a) ground truth

c) resulting error

b) our result

d) error of VU [4]

Fig. 6. *Fountain-P11* dataset [5] overview diffuse rendering and error maps. Accurate regions are white, missing reconstruction is red and green area was not evaluated.



a) input image

d) final mesh refinement

g) mesh refinement error

b) ground truth

e) result of FUR [10]

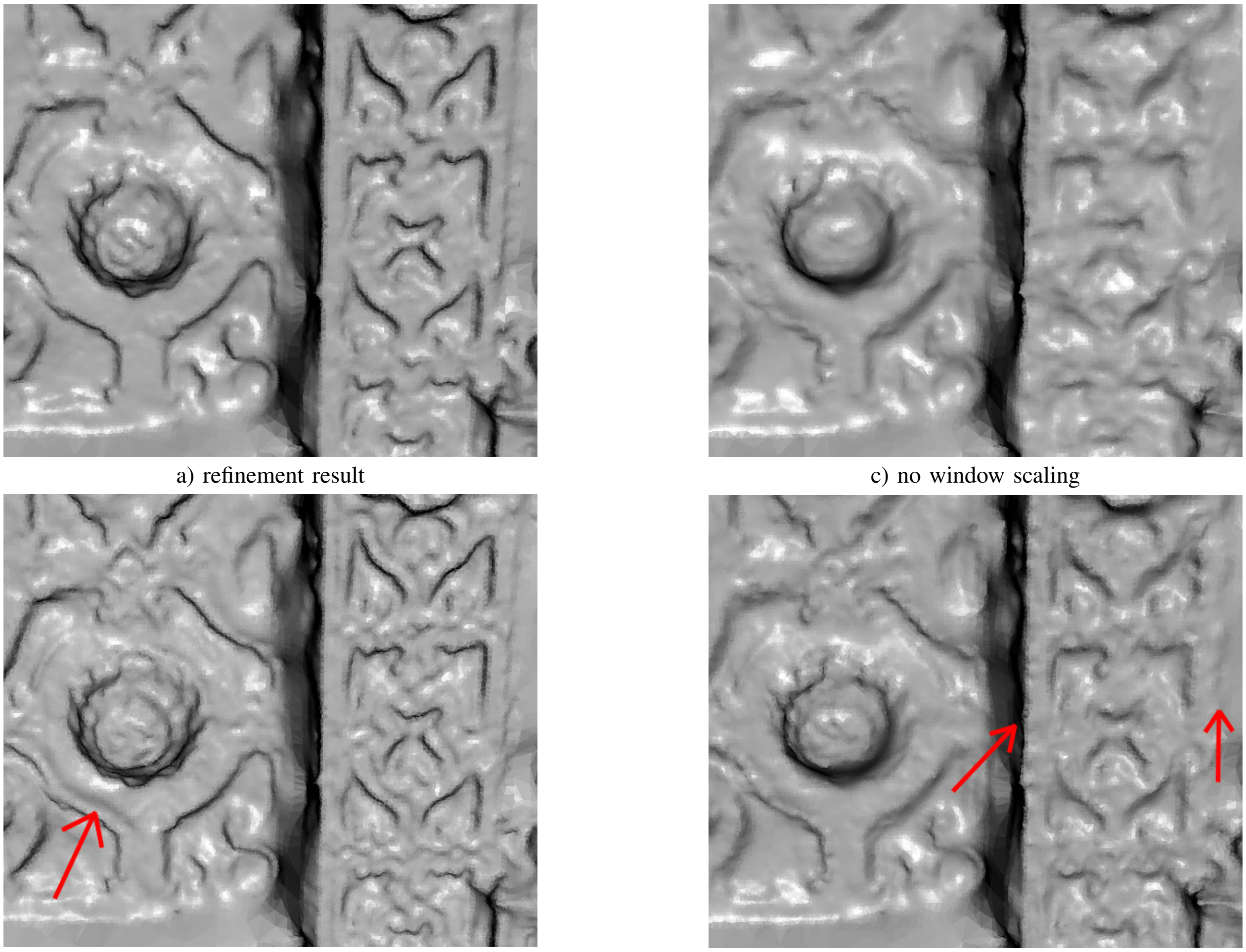
h) error of FUR [10]

c) initial depth map fusion

f) result of VU [4]

i) error of VU [4]

Fig. 7. *Fountain-P11* dataset [5] detailed rendering and error maps (white=accurate, black=inaccurate,red=missing).



a) refinement result

c) no window scaling

b) no offset correction

d) no contour term

Fig. 8. Demonstration of effect of individual contributions on *Fountain-P11* dataset [5].

be also seen from this comparison that the majority of the edges are recognized as contour generators (ϕ_C), including the sunken ornaments, after they are first 'discovered' by image matching (ϕ_I). On the other hand, we can encounter false contour generators detected on noisy initial surface, which can cause the surface to create phantom edges. This particularly affects textured surfaces, and it has to be avoided by more robust detection of contour generators. Next, without image offset correction in b), surface in flat regions becomes sinuous while the edges are correct thanks to the contour information as it is invariant to image offset errors. Finally, when we omit the iterative scale space approach in c), the surface becomes globally oversmoothed or eventually overfitted to data depending on the fixed window size.

To demonstrate the possibilities of the method on large-scale data, we have used it to reconstruct the statue *Asia*, which is a part of the Albert Memorial in London. We captured a suite of 238 photographs (Figure 9), which consists of several semi-rings, three monocular from about 2m, 4m and 40m distance and one stereo with non-uniform (free-hand) vertical baseline from about 8m distance plus some additional images. All photos have been shot by Canon PowerShot G7 (10 Mpix) with variable focal length and with image stabilization on, and carefully corrected for radial distortion. The variable lighting conditions (moving clouds) were compensated by our offset correction (up to 25% of the intensity range). The model reconstructed with depth map fusion [15] shown in Figure 10 includes intricate features like elephant's tusks, but some parts of the surface are only approximated due to missing data (tops and some back parts of the statue). We performed subsequent refinement in the same way as previous datasets. Since we have no ground truth data available, the effect of refinement can be demonstrated visually by introduction of details, like rug fringes on the elephant's head in Figure 11 c).

VII. Conclusion

We have proposed a method towards increasing accuracy in MVS. Variable 3D surface representation allows us to achieve efficient camera pose refinement together with surface geometry refinement. Surface contour modeling helps utilize independent sources of 3D shape information present in the images, while image offset correction compensates for the effect of their exposure. A scale-space approach is employed to find the correct surface within noisy data. In our future work we plan tying the processes of calibration and refinement more closely together.

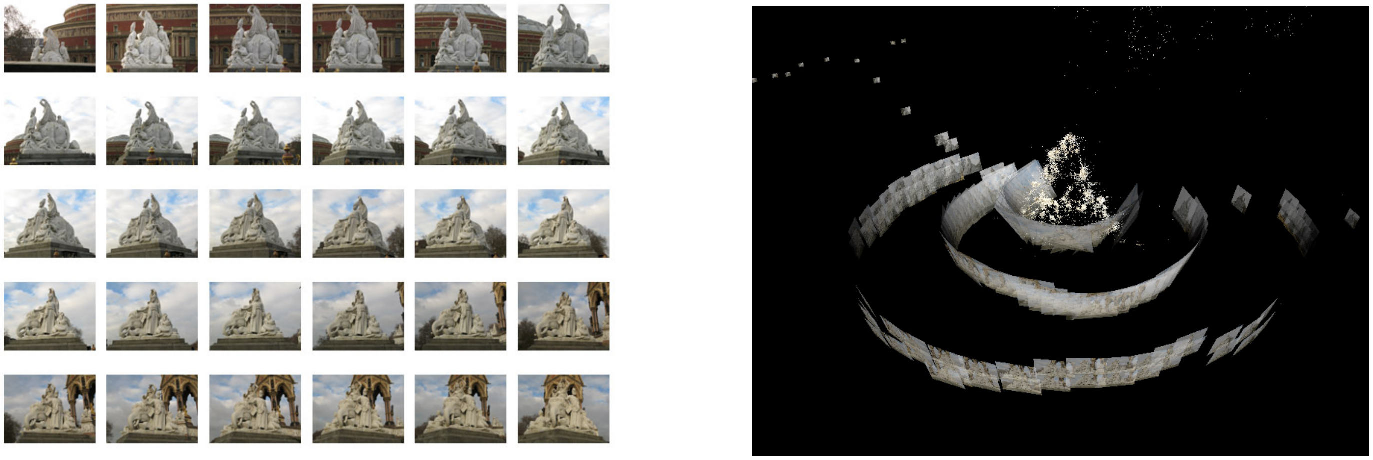


Fig. 9. The *Asia* dataset (Albert Memorial, London). Left: some of 238 input images. Right: scene with camera positions and sparse points computed from initial sparse matching.

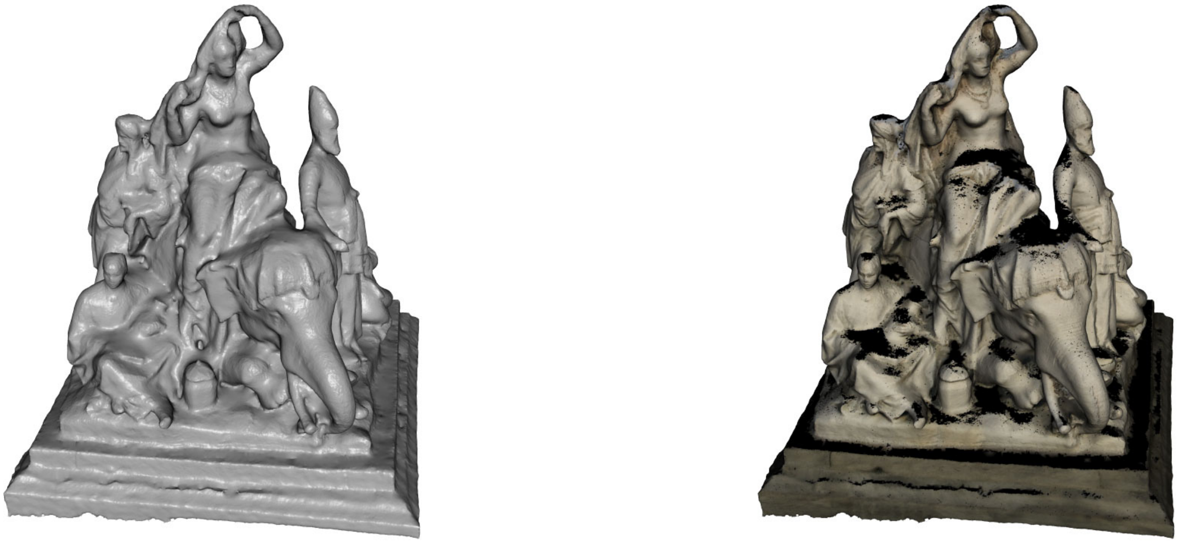
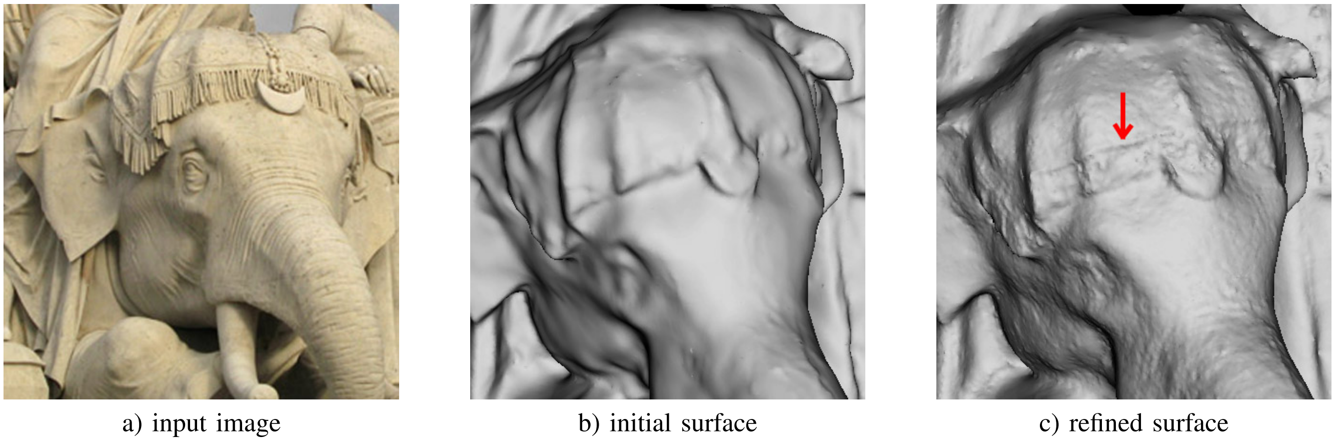


Fig. 10. Mesh refinement results on the *Asia* dataset. Left: final model without texture. Right: final textured model. The viewpoint is different from input images, untextured (black) parts are not visible in any of them.



a) input image

b) initial surface

c) refined surface

Fig. 11. Demonstration of mesh refinement on the *Asia* dataset, elephant's head in detail (without texture).

Acknowledgment

The authors would like to thank Christoph Strecha for help with his evaluation system.

References

[1] Y. Furukawa and J. Ponce, "Accurate Camera Calibration from Multi-View Stereo and Bundle Adjustment," in *Proc CVPR*, 2008.

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

[3] C. Strecha, R. Fransens, and L. Van Gool, "Combined Depth and Outlier Estimation in Multi-View Stereo," in *Proc CVPR*, 2006, pp. 2394–2401.

[4] H. Vu, R. Keriven, P. Labatut, and J.-P. Pons, "Towards high-resolution large-scale multi-view stereo," in *Proc CVPR*, 2009.

[5] C. Strecha, W. von Hansen, L. Van Gool, P. Fua, and U. Thoennessen, "On Benchmarking Camera Calibration and Multi-View Stereo for High Resolution Imagery," in *Proc CVPR*, 2008.

[6] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski, "A Comparison and Evaluation of Multi-View Stereo Reconstruction Algorithms," in *Proc CVPR*, 2006, pp. 519–528.

[7] G. Vogiatzis, P. H. S. Torr, and R. Cipolla, "Multi-View Stereo via Volumetric Graph-Cuts," in *Proc CVPR*, 2005, pp. 391–398.

[8] O. D. Faugeras and R. Keriven, "Complete dense stereovision using level set methods," in *Proc ECCV*, 1998, pp. 379–393.

[9] C. Strecha, R. Fransens, and L. V. Gool, "Wide-baseline stereo from multiple views: A probabilistic account," in *Proc CVPR*, 2004, pp. 552–559.

[10] Y. Furukawa and J. Ponce, "Accurate, dense, and robust multi-view stereopsis," in *Proc CVPR*, 2007.

[11] M. Desbrun, M. Meyer, P. Schröder, and A. H. Barr, "Implicit fairing of irregular meshes using diffusion and curvature flow," in *SIGGRAPH*, 1999.

[12] A. Zaharescu, E. Boyer, and R. P. Horaud, "Transformesh: A topology-adaptive mesh-based approach to surface evolution," in *Proc ACCV*, ser. LNCS 4844. Springer, November 2007, pp. 166–175.

[13] N. Dyn, K. Hormann, S.-J. Kim, and D. Levin, "Optimizing 3D triangulations using discrete curvature analysis," in *Mathematical Methods for Curves and Surfaces*, 2001, pp. 135–146.

[14] D. Martinec and T. Pajdla, "Robust rotation and translation estimation," in *Proc CVPR*, June 2007.

[15] R. Tyleček and R. Šára, "Depth Map Fusion with Camera Position Refinement," in *Proc Computer Vision Winter Workshop*, Eibiswald, Austria, February 2009, pp. 59–66.

[16] S. Soatto, A. J. Yezzi, and H. Jin, "Tales of shape and radiance in multi-view stereo," in *Proc ICCV*, 2003, pp. 974–981.

[17] A. Delaunoy, E. Prados, P. Gargallo, J. Pons, and P. Sturm, "Minimizing the Multi-view Stereo Reprojection Error for Triangular Surface Meshes," in *Proc BMVC*, 2008.

[18] J. Čech and R. Šára, "Efficient sampling of disparity space for fast and accurate matching," in *BenCOS 2007: CVPR Workshop Towards Benchmarking Automated Calibration, Orientation and Surface Reconstruction from Images*. IEEE, June 2007.

[19] M. Kazhdan, M. Bolitho, and H. Hoppe, "Poisson surface reconstruction," in *Proc Eurographics Symposium on Geometry Processing*, 2006, pp. 61–70.

[20] J. Koenderink, "What does the occluding contour tell us about solid shape," *Perception*, vol. 13, no. 3, pp. 321–30, 1984.

[21] R. Keriven, "A variational framework for shape from contours," Ecole Nationale des Ponts et Chaussees, CERMICS, France, Tech. Rep., 2002.

[22] H. Jin, "Variational methods for shape reconstruction in computer vision," Ph.D. dissertation, Washington University, St. Louis, MO, USA, 2003.

Radim Tyleček is a PhD student at the Center for Machine Perception, which is a part of Departement of Cybernetics, Faculty of Electrical Engineering, Czech Technical University in Prague. He received his master degree there in 2008. Since his master studies, he has been interested in computer vision, where he focuses on accurate image-based 3D reconstruction.

Radim Šára is an associate professor at the Czech Technical University in Prague since 2008. He received his PhD degree in 1994 from the Johannes Kepler University in Linz, Austria. From 1995 to 1997 he worked at the GRASP Laboratory at University of Pennsylvania. In 1998 he joined the Center for Machine Perception where he is currently. His research interests center on computer vision, including robust stereovision, shape-from-X methods, and structural object recognition.