Variational PatchMatch MultiView Reconstruction and Refinement

Philipp Heise, Brian Jensen, Sebastian Klose, Alois Knoll
Department of Informatics, Technische Universität München, Germany
{heise,kloses,jensen,knoll}@in.tum.de

Abstract

In this work we propose a novel approach to the problem of multi-view stereo reconstruction. Building upon the previously proposed PatchMatch stereo and PM-Huber algorithm we introduce an extension to the multi-view scenario that employs an iterative refinement scheme. Our proposed approach uses an extended and robustified volumetric truncated signed distance function representation, which is advantageous for the fusion of refined depth maps and also for raycasting the current reconstruction estimation together with estimated depth normals into arbitrary camera views. We formulate the combined multi-view stereo reconstruction and refinement as a variational optimization problem. The newly introduced plane based smoothing term in the energy formulation is guided by the current reconstruction confidence and the image contents. Further we propose an extension of the PatchMatch scheme with an additional KLT *step to avoid unnecessary sampling iterations. Improper camera poses are corrected by a direct image alignment step that performs robust outlier compensation by means of a recently proposed kernel lifting framework. To speed up the optimization of the variational formulation an adapted scheme is used for faster convergence.*

1. Introduction

We consider the problem of performing a dense 3d reconstruction from a set of calibrated 2d images. Many algorithms have been proposed to solve this problem with encouraging results. The widespread availability of 3d printing increases the demand for accurate reconstruction of objects using a set of camera images. It is worth considering that often camera poses are in practice not perfectly accurate, even for externally tracked cameras, and that the manual selection of well-suited camera views is not feasible. Therefore any reconstruction algorithm should not only try to find a dense 3d reconstruction that minimizes the photometric reprojection error between views, but also refine the camera poses for enhanced photo consistency. Also the computational feasibility has to be considered, especially for reconstructions involving many high resolution images. The efficiency of massively parallel systems in combination with parallel algorithms seem to be well suited for processing such big amounts of visual data. Our algorithm tries to address all of the aforementioned challenges of multi-view stereo reconstruction. Typical results for our algorithm are shown in figure 1.

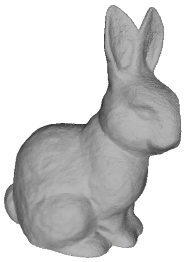

Figure 1: Reconstruction result of the proposed algorithm applied to the Bunny dataset provided by Kolev *et al*. [23]

1.1. Related work

The method of Furukawa and Ponce [14] uses a strategy that matches image patches, expands the correspondences in the neighbourhood and filters based on visibility constraints starting with sparse matches in the images resulting in a semi-dense point-cloud. Kolev *et al*. [23, 24] try to solve the reconstruction problem by minimization of convex energy functionals. In [37] an algorithm based on the fusion of range images is proposed by Zach *et al*. Fusion of depthmaps for reconstruction and meshing has also been proposed in the approach of Curless and Levoy [12].

The estimation of range images from two or views is a widely considered problem on its own and we consider only some closely related works. Similar to the image patch matching and expansion by Furukawa and Ponce [14] is the *PatchMatch* stereo algorithm by Bleyer *et al*. [9] that tries to perform dense stereo matching using sampling and propagation. Several variants of this algorithm have been proposed that introduce explicit regularization like PMBP by Besse *et al*. [6] and PM-Huber by Heise *et al*. [18].

The problem of pose estimation given image and depth

data was addressed in the DTAM algorithm by Newcombe *et al*. [30] and also by Steinbrücker *et al*. [34]. The DTAM system by Newcombe *et al*. [30] also address the problem of reconstruction from the recorded images. The Kinectfusion approach by Newcombe *et al*. [29] and Izadi *et al*. [19] uses an RGB-D sensor for the simultaneous pose estimation and scene reconstruction.

The dense bundle adjustment approach proposed by Amaël Delaunoy and Marc Pollefeys [13] addresses the same issues as our approach but uses a different strategy to optimize the overall photo consistency in terms of the reconstruction and the camera poses.

1.2. Contribution

The main contributions of our paper are:

- An extended truncated signed distance volume representation that uses expectation maximization together with a Gaussian noise plus uniform outlier model for filtering and per voxel confidence
- A new Variational *PatchMatch* MultiView formulation that operates directly on local planes and allows the joint optimization of depth and normals. The resulting efficient second order regularization of the depth also incorporates the confidence of the *TSDF* volume.
- Extension of the *PatchMatch* algorithm with a direct image patch alignment step to speed up convergence and to reduce the number of necessary sampling steps
- Automatic selection of reasonable camera views and direct optimization of the camera poses using the recently proposed kernel lifting framework [39, 38]

2. Method

The overall goal of our algorithm is to increase the photo-consistency of the provided images by minimization of the photometric reprojection error. On the one hand we try to improve our estimate of the scene geometry and on the other hand we try to optimize the initial camera poses to minimize the photometric error. Our approach is comparable and most similar to the dense bundle adjustment approach proposed by Amaël Delaunoy and Marc Pollefeys [13], but with a completely different algorithmic approach. Our algorithm starts with a crude approximation of the surface generated by the visual hull of the object [5, 26]. The current surface is then raycasted from an extended volumetric distance representation based on the method by Curless and Levoy [12, 29, 19] to a depthmap plus normal representation. The extracted information is then refined using a variational *PatchMatch* stereo variant [18, 9, 4, 7] that directly operates on local planes and therefore allows a direct and joint optimization of the depth and normals. Further the refinement of our algorithm is not solely based on sampling as in the original *PatchMatch* algorithm but also incorporates a direct intensity refinement step for our plane formulation similar to the well known *KLT* alignment [3] and uses several images at once for a more robust data term. The refined depthmaps are then re-added to the extended and robustified truncated signed distance volume representation that uses expectation maximization to filter outliers. The raycasting and refinement steps are performed for many view combinations and repeated over several scales. After each scale we furthermore perform direct image alignment to refine the camera poses [34, 30, 21, 22],which in our formulation are assumed to be good initial estimates of the true poses. Additionally image subsets of the *PatchMatch* correspondence estimation phase, which do not have a reasonable reprojection calculated by the direct pose refinement, are pruned. This helps to remove images that do not match very well e.g. due to occlusions induced by the scene geometry or due to changes in the illumination and allows a fully automatic selection of reasonable image combinations for the correspondence estimation. For the pose refinement we use the recently proposed kernel lifting scheme of Zollhöfer *et al*. [39] and Christopher Zach [38]. The exact details for each stage of the algorithm are explained in the following sections. Figure 2 gives an overview of the algorithm.

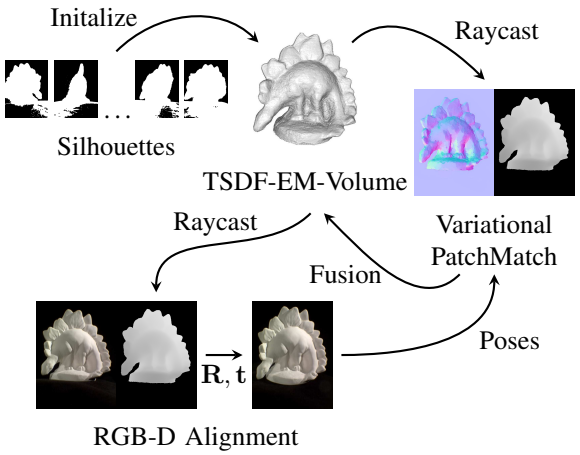

Figure 2: Overview of the proposed algorithm.

2.1. Initialization and Visual Hull Computation

We initialize our volume with the visual hull of the scene. If the visual hull is not available or can not be computed many other basic geometry initialization should be sufficient e.g. simple depthmap fusion. For computation of the visual hull we perform a simple thresholding operation to segment the input images into foreground and background. The original formulation of the volumetric truncated signed distance representation of Cureless and Levoy [12] uses the arithmetic mean to average several input depthmaps. To es-

timate the visual hull from the segmented input images we propose to use the geometric mean, which has the property that if one of the samples is zero then the mean also has zero value. If one voxel is segmented as outside in one view the geometric mean will result in a value of zero although the voxel might be segmented as foreground in other views. As proposed in [19, 29] we store the current value and a weight (here the number of samples) in the volume. The update rules for the running geometric mean with a pixel value of $f \in \{\text{background} = 0, \text{foreground} = 1\}$ for the function $F(\mathbf{x})$ and the weight $W(\mathbf{x})$ with $\mathbf{x} \in \Omega \subset \mathbb{R}^3$ are given by

$$F_{i+1}(\mathbf{x}) = F_i(\mathbf{x})^{\frac{W_i(\mathbf{x})}{W_{i+1}(\mathbf{x})}} \cdot f^{\frac{1}{W_{i+1}(\mathbf{x})}} \tag{1}$$

$$W_{i+1}(\mathbf{x}) = W_i(\mathbf{x}) + 1. \tag{2}$$

The value of f is determined by projecting $\mathbf{x}$ into the segmented image $S_i : \Omega_S \subset \mathbb{R}^2 \rightarrow \{0, 1\}$ using the extrinsics $\mathbf{R}_i, \mathbf{t}_i$ and the intrinsics $\mathbf{K}_i$ of the corresponding camera, resulting in the lookup position $(u, v)^\top = \pi(\mathbf{K}_i[\mathbf{R}_i|\mathbf{t}_i]\,\mathbf{x})$ with $\mathbf{x}$ being in homogeneous coordinates and π being the perspective projection [17]. To simplify the depthmap and normal extraction we actually apply a simple function before storing values in the volume that maps the range $[0, \ldots, 1] \rightarrow [-1, \ldots, 1]$ in order to make the isolevel for surface extraction the zero level [19, 29, 27]. Before the values are updated, the inverse transformation is applied to the stored values. Having calculated the visual hull using all the segmented input images we set all weights to a small constant value, e.g. one. Because the visual hull is only an initial approximation to the real non-convex surface, the influence of this initial approximation using large weights would be too exaggerated with the normal running average for the fusion of the refined depth maps. Typical results for the visual hull computation for the full Dino dataset [32] are shown in figure 3.

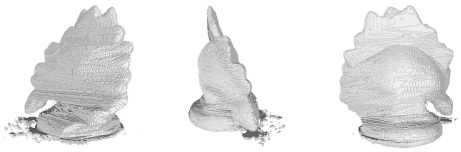

Figure 3: Typical results for the visual hull computation for the full Dino dataset [32].

2.2. Robust Depthmap Fusion

We are using a variant of the truncated signed distance function volume (*TSDF*) proposed by Curless and Levoy [12]. While there are many other efficient and less memory consuming algorithms available to perform depthmap fusion and meshing *e.g. Poisson* surface reconstruction[20], the *TSDF* has the advantage that it is an online algorithm that allows the integration of estimations adaptively and further is able to raycast virtual depthmaps into arbitray views at ease. These properties make the *TSDF* representation unique, but still efficient in combination with parallel algorithms and hardware like GPUs [29, 19]. We extend the *TSDF* representation to make it more robust against outliers, that are common for multiview- and stereo depthmaps. We use a probabilistic uniform outlier plus Gaussian mixture model to represent the truncated distance probability at each voxel

$$p(x|\mu, \sigma^2, w) = (1 - w)\,\mathcal{U}(x| - 1, 1) + w\,\mathcal{N}(x|\mu, \sigma^2). \tag{3}$$

A similar model was used by Geoerge Vogiatzis and Carlos Hernández [35] but works directly on the depth values instead of the truncated distance and was optimized using a parametric approximation to the posterior. Contrary to the results in [35] we found that the optimization with the *EM* algorithm works quite well, given a reasonable initialization. To maintain the online ability of the *TSDF* representation, we need to reformulate the classical expectation maximization (*EM*) algorithm [8] as an online variant suitable for our mixture model. We use an online variant for mixture models similar to the one proposed by Allou Samé *et al.* [31] leading to the following formulation for the update rules. We evaluate the mixture component responsibility in the n-th iteration of the Gaussian as

$$\gamma(z_\mathcal{N}) = \frac{w_n\,\mathcal{N}(x_{n+1}|\mu_n, \sigma_n^2)}{(1 - w_n)\,\mathcal{U}(x_{n+1}| - 1, 1) + w_n\,\mathcal{N}(x_{n+1}|\mu_n, \sigma_n^2)}. \tag{4}$$

The parameters of the Gaussian are then updated using

$$N_{\mathcal{N},n+1} = N_{\mathcal{N},n} + \gamma(z_\mathcal{N}) \tag{5}$$

$$\mu_{n+1} = \frac{\mu_n\,N_{\mathcal{N},n} + \gamma(z_\mathcal{N})\,x_{n+1}}{N_{\mathcal{N},n+1}} \tag{6}$$

$$\bar{x}_{n+1}^2 = \frac{\bar{x}_n^2\,N_{\mathcal{N},n} + \gamma(z_\mathcal{N})\,x_{n+1}^2}{N_{\mathcal{N},n+1}} \tag{7}$$

$$\sigma_{n+1}^2 = \bar{x}_{n+1}^2 - \mu_{n+1}^2. \tag{8}$$

The mixture weight w is updated using the rule

$$w_{n+1} = \frac{N_{\mathcal{N},n+1}}{N + 1}, \tag{9}$$

and directly represents the number of inliers according to the Gaussian noise component. Initially we set the mean μ for each voxel to the value from the visual hull computation and the other values to a constant values leading to a high variance for the Gaussian component. In figure 4 we plot the results of the proposed online algorithm applied to a synthetic signal sampled from a *Gaussian* component with $\mu = 0, \sigma^2 = 0.1$ and $w = 0.8$ leading to 0.2 for the uniform outlier component.

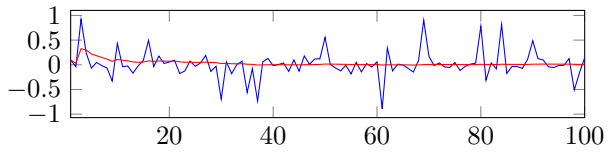

Figure 4: Synthetic samples drawn from the mixture with mean $\mu = 0$, $\sigma^2 = 0.1, w = 0.8$ in blue. The *EM* filtered result is shown in red.

The authors of the Kinectfusion system [29, 19] proposed to weight each depth sample added to the volume *e.g.* by the normal direction in the view. We can also easily incorporate such a weighting scheme by multiplying the responsibility $\gamma(z_n)$ with our weight ω and by adding ω to N in equation (9) instead of adding 1. In practice we store $\mu, \bar{x}^2, N_{\mathcal{N}}$ and N at each voxel leading to a doubled memory consumption compared to the original formulation of Cureless and Levoy [12]. For the weighting ω we use a combination of the normal direction and the data term from the *PatchMatch* phase.

Despite the new update rules, the integration of depthmaps in the volume is performed as described in [29, 19]. Raycasting of depth and normal maps into new views is also performed as described in [29, 19], but we treat the μ component of each voxel as our current distance value. We are also able to raycast our weight w into the view, which results in a confidence value describing the percentage of inliers.

2.3. Variational Depth and Normal Map Optimization

For the variational depthmap- and normal estimation and refinement the current contents of the TSDF volume are raycasted into the ith camera with the projection matrix $\mathbf{K}_i[\mathbf{R}_i|\mathbf{t}_i]$ resulting in a current estimate of the depthmap and normal-map for the camera i. For the refinement of the raycasted estimate we propose a multiview stereo algorithm based on the minimization of an energy function

$$E = E_{\text{data}} + \lambda E_{\text{smooth}}, \tag{10}$$

consisting of a data term describing the similarity between patches in several images, which are related using a local plane approximation of the surface, and a smoothness term favoring similar local planes in adjacent pixels.

2.3.1 Smoothness Term

Given a camera $\mathbf{K}[I|\mathbf{0}]$ at the origin and a plane $\pi(x) = \left[\mathbf{n}^\top d\right]^\top$ with the normal $\mathbf{n}$ and a distance d. Any ray $\mathbf{X} = [(\mathbf{K}^{-1}\mathbf{x})^\top \rho]^\top$ parameterized by ρ must fulfill the following equation for a ray-plane intersection [17]

$$\pi^\top [(\mathbf{K}^{-1}\mathbf{x})^\top \rho]^\top = 0 \tag{11}$$

$$[\frac{\mathbf{n}^\top}{d}\ 1]\,[(\mathbf{K}^{-1}\mathbf{x})^\top \rho]^\top = 0 \Rightarrow \rho = -\frac{\mathbf{n}^\top}{d}\,\mathbf{K}^{-1}\mathbf{x}. \tag{12}$$

In the following we refer to $\frac{\mathbf{n}}{d}$ as $\pi' : (\Omega \subset \mathbb{R}^2) \rightarrow \mathbb{R}^3$ and ρ refers to the inverse depth. Given a plane for each pixel we assume that the neighbouring pixels should also have a similar local plane with only small deviations except at surface boundaries where the planes differ a lot. As we have seen in equation (12) the resulting inverse depth of course also depends on the pixel coordinates, such that a small change e.g. of the normal leads to different inverse depth changes in the image at different pixel positions. This behaviour is illustrated in figure 5. Given two neighbouring pixels $\mathbf{x}_a$ and

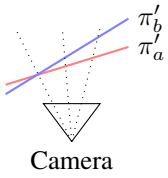

Figure 5: Two planes π'_a and π'_b lead to very different depth values at the intersections depending on the ray's pixel position on the camera plane. The camera rays are shown as dotted lines.

$\mathbf{x}_b$ we are interested in second order smoothness instead of a first order smoothness given by $\nabla\rho$; the ray originating at $\mathbf{x}_a$ intersected with the neighbouring plane should result in a similar inverse depth value as from the own plane:

$$\pi'(\mathbf{x}_a)^\top \mathbf{K}^{-1}\mathbf{x}_a - \pi'(\mathbf{x}_b)^\top \mathbf{K}^{-1}\mathbf{x}_a \tag{13}$$

$$= \nabla_{a,b}\pi'(\mathbf{x}_a)^\top \mathbf{K}^{-1}\mathbf{x}_a, \tag{14}$$

where $\nabla_{a,b}$ refers to the gradient in the direction from $\mathbf{x}_a$ to $\mathbf{x}_b$. The combined objectives of plane similarity and second order smoothness lead to the following smoothness term

$$E_{\text{smooth}}(\pi') = \lambda_1 \sum_{d=1}^{3} |\mathbf{D}\,\nabla\pi'_d(\mathbf{x})| + \lambda_2 \left|\mathbf{D} \begin{pmatrix} (\mathbf{K}^{-1}\mathbf{x})^\top & \mathbf{0} \\ \mathbf{0} & (\mathbf{K}^{-1}\mathbf{x})^\top \end{pmatrix} \mathbf{P}\,\nabla\pi'(\mathbf{x})\right|, \tag{15}$$

where the scalar values λ_1 and λ_2 allow to balance between the two smoothness terms and the data term. $\mathbf{P}$ refers to a 6×6 permutation matrix that maps the components of the gradient in x direction to the first three entries and the y direction components in the last three entries. The second term results in the inverse depth difference between the neighbouring plane in x direction and the local plane in the first vector entry and the according difference in y direction in the second entry. The first term leads to an anisotropic

TV regularization as used in an optical flow formulation by Werlberger *et al*. [36]. The matrix $\mathbf{D}$ is a 2×2 scaled diffusion tensor as employed by Werlberger *et al*. [36] and Kuschk *et al*. [25] uses the image contents to guide the regularization. Given an image I the diffusion tensor at the location x is given by

$$\mathbf{D}_{\text{Tensor}}(x) = \exp(\alpha |\nabla I(x)|^{\beta}) \cdot \mathbf{n}\mathbf{n}^{\top} + \mathbf{n}_{\perp}\mathbf{n}_{\perp}^{\top}, \quad (16)$$

with $\mathbf{n} = \frac{\nabla I}{|\nabla I|}$ being the gradient direction and $\mathbf{n}_{\perp}$ being a vector in the orthogonal direction [36]. To further guide the regularization we use the raycasted confidence c stemming from the *TSDF* w value at the assumed surface voxel to perform less regularization in regions with a high inlier rate leading to the final $\mathbf{D}$ matrix

$$\mathbf{D} = \frac{1}{1 + \tau\, c}\mathbf{D}_{\text{Tensor}}(x). \quad (17)$$

The parameters α, β and τ are scalar parameters that we set to the values $10, 0.8$ and 1. The smoothness as a whole favours similar planes but allows to balance between pure plane similarity and the inverse depth difference when intersected with a ray.

2.3.2 Data term

As proposed by Gallup *et al*. [15] and also used by Heise *et al*. [18] we use the plane-induced homography [17] to evaluate the likelihood of local planes approximating the real surface. The homography from the camera s to the camera t induced by the plane $\pi = [\mathbf{n}^{\top}\ d]^{\top}$ is given by

$$\begin{aligned} H_{s,t}(\mathbf{n}, d) =& H(\mathbf{K}_t, \mathbf{R}_t, \mathbf{t}_t, \mathbf{K}_s, \mathbf{R}_s, \mathbf{t}_s, \mathbf{n}, d) \\ =& \mathbf{K}_t(\mathbf{R}_t\mathbf{R}_s^{\top} - \frac{1}{d}(\mathbf{t}_t - \mathbf{R}_t\mathbf{R}_s^{\top}\mathbf{t}_s)\mathbf{n}^{\top})\mathbf{K}_s^{-1} \\ =& \mathbf{K}_t(\mathbf{R}_t\mathbf{R}_s^{\top} - (\mathbf{t}_t - \mathbf{R}_t\mathbf{R}_s^{\top}\mathbf{t}_s)\pi'^{\top})\mathbf{K}_s^{-1} \\ =& H_{s,t}(\pi') \end{aligned} \quad (18)$$

For the evaluation of the likelihood we use the cameras in the neighbourhood $\mathcal{N}$ of our currently selected camera i and their corresponding images I_k with $k \in \mathcal{N}(i)$. Given the color images $I_i, I_k : (\Omega \subset \mathbb{R}^2) \rightarrow \mathbb{R}^3$ and the local plane map $\pi' : \Omega \rightarrow \mathbb{R}^3$ we can evaluate the corresponding data term with

$$E_{\text{data}}(\pi') = \frac{1}{Z_K Z_w} \sum_{k \in \mathcal{N}(i)} \sum_{\mathbf{q} \in \mathcal{N}(\mathbf{x})} w_i(\mathbf{x}, \mathbf{q}) \quad (19)$$

$$\rho_{i,k}(\mathbf{x}, H_{i,k}(\pi'(\mathbf{x}))\,\mathbf{q}). \quad (20)$$

The function $\rho_{i,k}$ is the pixel similarity measurement function between the images I_i and I_k as proposed by Bleyer *et al*. [9]

$$\begin{aligned} \rho_{i,k}(\mathbf{p}, \mathbf{q}) =& (1 - \alpha)\min(\|I_i(\mathbf{p}) - I_k(\mathbf{q})\|_1, \tau_{\text{col}}) + \\ & \alpha \min(\|\nabla I_i(\mathbf{p}) - \nabla I_k(\mathbf{q})\|_1, \tau_{\text{grad}}). \end{aligned} \quad (21)$$

The weighting function w_i is identical to the one proposed in [18] and weights according to the color similarity of the pixel within the image I_i and the influence changes with the distance to the center. The factors Z_K and Z_w normalize the data term according to the number of images used from the neighbourhood and the sum of the weighting factors w_i for each of the pixels in the patch.

2.3.3 Optimization

To minimize the overall energy E we couple the data term E_{data} and smoothness term E_{smooth} using a quadratic term to perform a relaxation of the optimization problem [2, 33]

$$E(\pi') = \lim_{\theta \rightarrow \infty} \int_{\Omega} E_{\text{data}}(\pi'_u) \quad (22)$$

$$+ \frac{\theta}{2}(\pi'_u - \pi'_v)^2 + E_{\text{smooth}}(\pi'_v)\,\mathrm{d}\mathbf{x}. \quad (23)$$

In the limit the difference between π_u and π_v has to be zero otherwise the difference would dominate the energy. The coupling simplifies our optimization problem and allows sampling of the data term given a smoothed π_v. For a fixed π_u we have to solve a *ROF* subproblem. The θ parameter is changed multiplicatively after each iteration and we alternate between the two sub-problems keeping the other parameter fixed.

Fixed π'_u, solve for π'_v:

At the first sight it might not be obvious that we can recast equation (15) combined with a quadratic term into an anisotropic *ROF* problem, but since equation (15) only involves linear terms we can stack the terms into one 8×6 matrix $\mathbf{A}$ leading to the anisotropic *TV* of one linear term with the gradient $\nabla\pi'$ as parameter. We dualize our problem following Chambolle and Pock [10] and introduce a dual variable $\mathbf{p}$ consisting of stacked vectors $\mathbf{p}_i \in \mathbb{R}^2$ with $i \in [1, \ldots, 4]$ leading to the dual anisotropic *ROF* optimization problem

$$\arg\min_{\pi'_v} \max_{\mathbf{p}_i, |\mathbf{p}_i| \leq 1} \int_{\Omega} \frac{\theta}{2}(\pi'_u - \pi'_v)^2 + \langle \mathbf{p}, \mathbf{A}\nabla\pi'_v \rangle - \sum_{i=1}^{4} \delta(\mathbf{p}_i)\,\mathrm{d}\mathbf{x}, \quad (24)$$

where $\delta(\mathbf{q})$ is the indicator function such that $\delta(\mathbf{q}) = 0$ if $|\mathbf{q}| \leq 1$ and otherwise ∞. Kuschk and Cremers [25] used an Augmented Lagrangian update scheme to speed up convergence. Similar formulations were also used by Chan *et al*. [11] and Afonso *et al*. [1]. We propose to perform an additional dualization of the quadratic term using the Legendre-Fenchel pair $f(\mathbf{x}) = \frac{\lambda}{2}\mathbf{x}^{\top}\mathbf{x} \Leftrightarrow f^*(\mathbf{p}) = \frac{1}{2\lambda}\mathbf{p}^{\top}\mathbf{p}$. We apply this dualization to one term after the decomposition of

the quadratic term into a sum of two halves $\frac{\theta}{2}(\pi'_u - \pi'_v)^2 = \frac{\theta}{4}(\pi'_u - \pi'_v)^2 + \frac{\theta}{4}(\pi'_u - \pi'_v)^2$ leading to

$$\frac{\theta}{2}(\pi'_u - \pi'_v)^2 = \max_{\mathbf{q}} \frac{\theta}{4}(\pi'_u - \pi'_v)^2 + (\pi'_u - \pi'_v)^\top \mathbf{q} + \frac{1}{\theta}\mathbf{q}^\top \mathbf{q} \tag{25}$$

In our case θ will go to infinity and therefore the quadratic term $\frac{1}{\theta}q^\top q$ will vanish and it becomes obvious that the formulation is equivalent to the method of the Augmented Lagrangian [11, 1, 25] in the limit up to a scale factor for θ. We found empirically that the splitting of the quadratic term and its half-dualization leads to a faster convergence that is at least as good as the Augmented Lagrangian method and often slightly better. For optimization we perform gradient ascent on the dual variables $\mathbf{p}, \mathbf{q}$ and gradient descent on the primal variable π_v. For further details regarding the primal dual optimization we refer to Chambolle and Pock [10] and Handa *et al.* [16].

Fixed π_v, solve for π_u:

As in the *PatchMatch* stereo algorithm by Bleyer *et al.* [9] and *PMHuber* by Heise *et al.* [18] we also employ a variant of the *PatchMatch* algorithm and evaluate several samples $\mathcal{S}$ for each pixel $\mathbf{x}$ and keep the best sample $s^\star$ minimizing our energy:

$$s^\star = \arg\min_{\pi'_u \in \mathcal{S}(\mathbf{x})} E_{\text{data}}(\pi'_u) + \frac{\theta}{2}(\pi'_u - \pi'_v)^2. \tag{26}$$

In [9] the authors proposed several sources for the set $\mathcal{S}(\mathbf{x})$ of samples to test. We found that the fixed selection of samples from the neighbourhood of the previous iteration introduces a significant bias and that in uncertain areas samples get propagated from a fixed direction. Bleyer *et al.* [9] avoided this problem by changing the traversal and propagation direction in the images. Our fully parallel implementation circumvents this problem by selecting a set of samples $\mathcal{S}_{RN}$ of randomly chosen neighbours within a certain neighbourhood. Joined with a small set of completely random samples $\mathcal{S}_R$ and the current value from the regularization subproblem $\mathcal{S}_{\text{smooth}}$ the complete set $\mathcal{S}$ is given

$$\mathcal{S} = \mathcal{S}_{RN} \cup \mathcal{S}_R \cup \mathcal{S}_{\text{smooth}}. \tag{27}$$

As in most *PatchMatch* variants we try to randomly refine the current best particle $s^\star$ by applying a small perturbation to the parameters.

While the overall strategy of random sampling and propagation is very successful, we have found out that a huge number of samples is necessary for finding the local optima. Further the propagation should be faster if particles are closer to their local optimal configuration e.g. best local plane fit. The data term given in equation (20) is difficult to optimize so that we therefore fall back to the simple sum of squared distances (*SSD*) to iteratively optimize the plane parameters using the Lucas-Kanade [3] algorithm. The image warping function is given by the plane induced homography from equation (18) and we want to optimize our plane parameter π'. Given the warping function

$$W_{s,t}(\mathbf{x}, \pi') = \Pi(H_{s,t}(\pi')\,\mathbf{x}) \tag{28}$$

with Π being the perspective division here. We are interested in minimizing the *SSD* between our image I_i and the destination images I_d using the linearized expression with respect to the additive parameters $\Delta\pi'$

$$\arg\min_{\Delta\pi'} \sum_{d \in \mathcal{N}(i)} \sum_{\mathbf{x}} (I_d(W_{i,d}(\mathbf{x}, \pi')) + \nabla I_d \frac{\partial W_{i,d}}{\partial \pi'} \Delta\pi' - I_i(\mathbf{x}))^2. \tag{29}$$

The minimizing step for each iteration is then computed by [3]

$$\Delta\pi' = \mathbf{H}^{-1} \sum_{d \in \mathcal{N}(i)} \sum_{\mathbf{x}} \mathbf{J}_{i,d}^\top (I_i(\mathbf{x}) - I_d(W_{i,d}(\mathbf{x}, \pi'))) \tag{30}$$

with $\mathbf{H}$ being the Gauss-Newton approximation of the Hessian $\mathbf{H} = \sum_{d \in \mathcal{N}(i)} \sum_{\mathbf{x}} \mathbf{J}_{i,d}^\top \mathbf{J}_{i,d}$. For solving the system the Jacobi $\mathbf{J}_{i,d}$ matrix of the plane induced homography needs to be calculated and applying the chain rule with $\mathbf{x}' = (u \;\; v \;\; w)^\top = H_{s,t}(\pi')\,\mathbf{x}$ and $\mathbf{p} = [\frac{u}{w} \; \frac{v}{w}]^\top$ results in

$$\mathbf{J}_{s,t} = \frac{\partial I_t(W_{s,t})}{\partial \pi'} = \frac{\partial I_t(\mathbf{p})}{\partial \mathbf{p}} \frac{\partial \Pi(\mathbf{x}')}{\partial \mathbf{x}'} \frac{\partial H_{s,t}(\pi')\mathbf{x}}{\partial \pi'} \tag{31}$$

$$= \nabla I_t \begin{pmatrix} 1/w & 0 & -u/w^2 \\ 0 & 1/w & -v/w^2 \end{pmatrix} \mathbf{K}_t(\mathbf{t}_t - \mathbf{R}_t\mathbf{R}_s^\top \mathbf{t}_s)(\mathbf{K}_s^{-1}\mathbf{x})^\top. \tag{32}$$

Having performed a few iterations we treat the optimized plane $\pi'^\star$ just as any other sample and evaluate our original likelihood function from equation (26). We have found that this simple *KLT* refinement speeds up the propagation and also leads to better local plane approximations, even when using much smaller patch sizes than the ones proposed in [9, 18]. To keep the notation uncluttered we omitted an additional weighting term that incorporates the same pixel intensity weighting scheme as used by our implementations data term and a Tikhonov regularization avoiding numerical issues in low gradient regions. The weighting leads to an additional diagonal matrix that needs to be integrated into the original least squares formulation and the Tikhonov regularization to an addition of a diagonal matrix to the Hessian.

2.4. Pose Optimization

Most reconstruction algorithms assume that the camera positions are fixed and exact but in reality this is rarely the

case. Delaunoy *et al.* [13] showed that also the poses in many typically used datasets are not completely accurate. To refine the initial poses we use direct image alignment as commonly used for Visual Odometry estimation using *RGB-D* sensors [34, 30, 21]. We use an forward compositional approach and try to align one image $I_s : \Omega \to \mathbb{R}$ and it's corresponding depthmap $D_s : \Omega \to \mathbb{R}$ raycasted from the *TSDFEM* volume to the second image $I_t : \Omega \to \mathbb{R}$ refining the relative pose $\mathbf{T}_{t,s}$ between the images. Instead of using ordinary least squares or iterative reweighted least squares we employ the recently proposed kernel lifting framework of Zollhöfer *et al.* [39] and Christopher Zach [38] for robust estimation

$$\arg\min_{\mathbf{p}} \sum_{\mathbf{u}\in\Omega} \frac{1}{2}\psi(\underbrace{I_t(\Pi(\mathbf{K_t}\mathbf{T_{t,s}}(\mathbf{p})\Pi^{-1}(\mathbf{u}, D_s(\mathbf{u})))) - I_s(\mathbf{u})}_{r(\mathbf{u},\mathbf{p})})$$
$$= \arg\min_{\mathbf{p}} \min_{w} \sum_{\mathbf{u}\in\Omega} \frac{1}{2}\left(w^2 r(\mathbf{u},\mathbf{p})^2 + \kappa^2(w^2)\right). \quad (33)$$

We perform a first order Taylor expansion of the residuals $r(\mathbf{p})$ as usually done for nonlinear least squares and try to solve for an optimal step update $\Delta\mathbf{p} \in \mathfrak{se}(3)$ minimizing the residuals. Our transformation is then iteratively updated until convergence

$$T(\mathbf{p}) \leftarrow T(\Delta\mathbf{p})T(\mathbf{p}), \quad (34)$$

with $T(\mathbf{p})$ being the exponential mapping relating the Lie algebra $\mathfrak{se}(3)$ to the Lie group $\mathbb{SE}(3)$. For robustness we use a smooth truncated quadratic function as described in [38] for the residual penalizing function ψ resulting in $\kappa^2(w^2) = \frac{\tau}{2}(w^2-1)^2$ with τ controlling the point of truncation. For details and a complete derivation we refer to the paper of Zach [38] that contains all the details. It is worth mentioning that we only optimize the pose between two images and therefore may introduce misalignments in other views or only compensate for pose error originating from the first image. Therefore we perform several iterations with randomly selected neighbouring image pairs too avoid the introduction large error and bias. Initially we select for each image its nearest neighbours and build subsets used in the *PatchMatch* depth estimation. These subsets have to be below a certain error in the relative pose refinement and are otherwise removed. The assumption is that if the direct reprojection error is very high that these images are either occluded, differently exposed or completely misaligned.

3. Evaluation

To evaluate the proposed extension of the *TSDF* with expectation maximization and the *KLT* step we generated a synthetic dataset of a scene containing the Standford Bunny on a plane. Our dataset contains 60 images of the color data as well as the depth data and also the exact camera

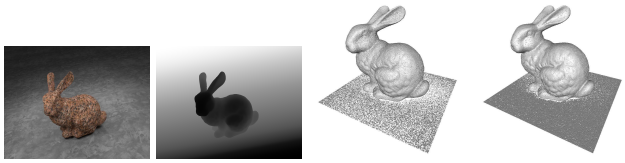

Figure 6: One color image and depth image of the synthetic dataset. Phong shaded reconstruction of the scene using a standard *TSDF* and our *TSDF-EM* variant using only the depth images with 2.5% uniform noise. The EM formulation is clearly able to filter much more of the noise.

poses describing approximately a half circle. In figure 6 a color and depth image of the synthetic dataset are shown along with the Phong shaded *TSDF* reconstructions using the depth images with 2.5% uniform noise with and without our EM extension. The reconstruction using the proposed EM extension is smoother and most of the floor is much better reconstructed because the zero isolevel is not pushed out of the volume. We also compared the synthetic depthmaps with the raycasted depthmaps from the *TSDF* reconstruction and as shown in figure 7 the error is much smaller with the EM filtering. The positive bias in the histogram comes from the truncation of the distance function and outliers in front of the correct value outside of the truncation distance do not contribute anymore. Therefore outliers behind the true surface contribute more and push the zero level outwards.

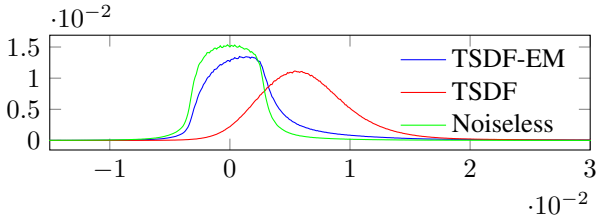

Figure 7: Histogram of the differences between the ground-truth depthmaps and the raycasted depthmaps from the volume with and without EM. For both *TSDF* variants the synthetic depthmaps with 2.5% uniform noise were used.

For the evaluation of the *KLT* step in *PatchMatch* we used 20 image triples and compared the results of our implementation with and without the *KLT* step against the ground-truth depth values after each of our 20 iterations. The percentage of depth errors < 0.025 after each iteration is shown in figure 8. Already after the random initialization the percentage of correct matches is much higher and also the overall inlier rate using *KLT* is better. The additional runtime overhead of the *KLT* step is negligible compared to the time for the likelihood evaluation for each sample. We perform at most 3 *KLT* iterations in our implementation. The graph in figure 8 shows that identical or even better results can be achieved even when the overall number of

PatchMatch iterations is reduced which would significantly reduce the overall runtime.

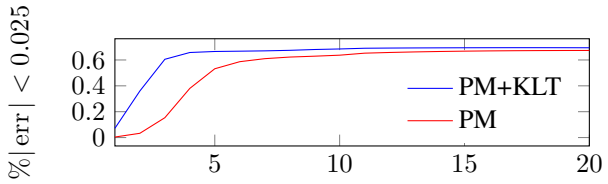

Figure 8: Percentage of pixels with depth $|\,\text{error}\,| < 0.025$ after each *PatchMatch* iteration averaged from 20 different views of the synthetic dataset with and without the *KLT* step. Convergence with the *KLT* step is much faster and also the percentage of correct matches is higher.

For evaluation of the complete proposed algorithm we use the Middlebury Multi-View Stereo benchmark by Seitz *et al*. [32]. We applied our algorithm to the full and ring datasets. The results in terms of accuracy and completeness can be found in table 1. The table also contains the results of the algorithms from Furukawa and Ponce [14], Mücke *et al*. [28] and Delaunoy and Marc Pollefeys [13].

In figure 9 the ground truth data, two views of the input images and the results for the methods from table 1 are shown. While the smoothing effect of our algorithm is clearly visible, fine details are still retained. Our method gives visually pleasing results that are competitive with the top performing methods.

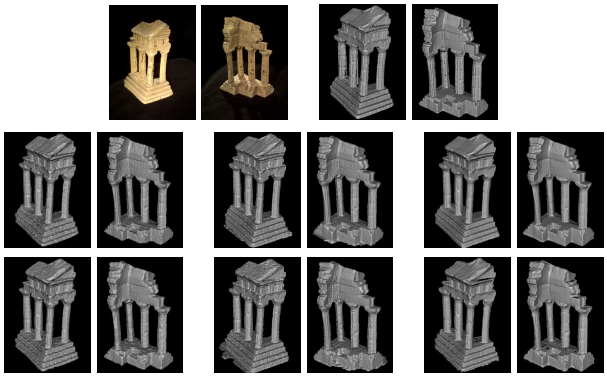

Figure 9: *Left to right and top to bottom*: two of the input images and ground truth data, reconstruction results as presented in the multi-view Middlebury benchmark for the method of Furukawa [14], Mücke [28] and the proposed method. In the middle row the reconstructions for the full dataset and at the bottom row the results for the ring dataset are shown.

The result for the Dino ring dataset in the brackets in the table 1 was the evaluation result when vertices inside the reconstruction were removed. Images of the reconstruction for the Dino are shown in figure 10 along with the ground truth data and the reconstruction results from Furukawa *et al*. [14].

Figure 10: *Left to right*: the ground truth data, reconstruction results for the full dino dataset for the method of Furukawa [14] and the proposed method.

For the evaluation we only used triples of images and we took triple combinations of the four closest images to our reference image. The patch size varied between 7 and 16 pixels. Our completely parallel implementation of the algorithm runs on a single GPU. On a NVidia GTX 770 the ring datasets took about 25 minutes to complete and for the full datasets it took approximately 2 hours and 40 minutes. As previously mentioned we actually perform several iterations of all images at different scales and therefore the runtime is also highly dependent on the scale space settings and the image size. We mainly used three pyramid levels with scaling factors $0.5, 1.0$ and 1.5, where the scaling affected only the images and the camera intrinsics but the *TSDF-EM* volume size did not change.

4. Conclusion

We have presented an algorithm that performs an iterative refinement of depth and normal maps that are raycasted and fused using a *TSDF-EM* volume. Our algorithm uses a variational *PatchMatch* method with an additional *KLT* refinement step, that integrates the current confidence in the depth value estimation, and tries to directly minimize the intensity difference using a local plane approximation in image space. Camera poses are refined using a direct image alignment step combined with recently proposed kernel lifting framework [39, 38].

There are many opportunities to improve the presented algorithm by using more information for guiding the regularization e.g. information from the depth and normal-maps for finding sharp edges and discontinuities of the model. Consideration of occlusions within the selected images for refinement and using more images could improve the data term quality.

One limitation of the current approach is that the regularization is only taking place in the image space and therefore occluded areas are not regularized at all. Additional regularization operating on the whole volume could possibly alleviate this issue.

	Temple		Temple Ring		Dino		Dino Ring	
	Acc.	Comp.	Acc.	Comp.	Acc.	Comp.	Acc.	Comp.
Furukawa [14]	0.49	99.6	0.47	99.6	0.33	99.8	0.28	99.8
Proposed method	0.45	99.2	0.56	99.2	0.35	99.5	1.05 (0.46)	99.2 (98.7)
Mücke [28]	0.36	99.7	0.46	99.1	-	-	-	-
Delaunoy [13]	-	-	0.51	99.1	-	-	0.51	98.7

Table 1: Results for the Multiview Middlebury Benchmark [32].

References

[1] M. V. Afonso, J. M. Bioucas-Dias, and M. A. Figueiredo. Fast image recovery using variable splitting and constrained optimization. *Transactions on Image Processing*, 2010.

[2] J.-F. Aujol, G. Gilboa, T. Chan, and S. Osher. Structure-texture image decomposition—modeling, algorithms, and parameter selection. *IJCV*, 2006.

[3] S. Baker and I. Matthews. Lucas-Kanade 20 Years On: A Unifying Framework. *IJCV*, 2004.

[4] C. Barnes. *PatchMatch: a randomized correspondence algorithm for structural image editing*. PhD thesis, Princeton University, 2011.

[5] B. G. Baumgart. *Geometric Modeling for Computer Vision.* PhD thesis, DTIC Document, 1974.

[6] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz. PMBP: PatchMatch Belief Propagation for Correspondence Field Estimation. In *BMVC*, 2012.

[7] F. Besse, C. Rother, A. Fitzgibbon, and J. Kautz. PMBP: PatchMatch Belief Propagation for Correspondence Field Estimation. *IJCV*, 2013.

[8] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer Verlag, 2006.

[9] M. Bleyer, C. Rhemann, and C. Rother. PatchMatch Stereo - Stereo Matching with Slanted Support Windows. *BMVC*, 2011.

[10] A. Chambolle and T. Pock. A first-order primal-dual algorithm for convex problems with applications to imaging. *Journal of Mathematical Imaging and Vision*, 2011.

[11] S. H. Chan, R. Khoshabeh, K. B. Gibson, P. E. Gill, and T. Q. Nguyen. An Augmented Lagrangian Method for Total Variation Video Restoration. *Transactions on Image Processing*, 2011.

[12] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH*, 1996.

[13] A. Delaunoy and M. Pollefeys. Photometric Bundle Adjustment for Dense Multi-view 3D Modeling. *CVPR*, 2014.

[14] Y. Furukawa and J. Ponce. Accurate, Dense, and Robust Multiview Stereopsis. *PAMI*, 2010.

[15] D. Gallup, J. M. Frahm, P. Mordohai, Q. Yang, and M. Pollefeys. Real-time plane-sweeping stereo with multiple sweeping directions. *CVPR*, 2007.

[16] A. Handa, R. A. Newcombe, A. Angeli, and A. J. Davison. Applications of Legendre-Fenchel transformation to computer vision problems. Technical report.

[17] R. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2004.

[18] P. Heise, S. Klose, B. Jensen, and A. Knoll. PM-Huber: PatchMatch with Huber Regularization for Stereo Matching. *ICCV*, 2013.

[19] S. Izadi, R. A. Newcombe, D. Kim, O. Hilliges, D. Molyneaux, S. Hodges, P. Kohli, J. Shotton, A. J. Davison, and A. Fitzgibbon. KinectFusion: real-time dynamic 3D surface reconstruction and interaction. *SIGGRAPH*, 2011.

[20] M. M. Kazhdan and H. Hoppe. Screened poisson surface reconstruction. *ACM Transactions on Graphics*, 2013.

[21] C. Kerl, J. Sturm, and D. Cremers. Robust odometry estimation for rgb-d cameras. *ICRA*, 2013.

[22] S. Klose, P. Heise, and A. Knoll. Efficient Compositional Approaches for Real-Time Robust Direct Visual Odometry from RGB-D Data. In *IROS*, 2013.

[23] K. Kolev, M. Klodt, T. Brox, and D. Cremers. Continuous global optimization in multiview 3d reconstruction. *IJCV*, 2009.

[24] K. Kolev, T. Pock, and D. Cremers. Anisotropic minimal surfaces integrating photoconsistency and normal information for multiview stereo. *ECCV*, 2010.

[25] G. Kuschk and D. Cremers. Fast and Accurate Large-Scale Stereo Reconstruction Using Variational Methods. In *ICCV Workshops*, 2013.

[26] A. Laurentini. The visual hull concept for silhouette-based image understanding. *PAMI*, 1994.

[27] W. E. Lorensen and H. E. Cline. Marching cubes: A high resolution 3D surface construction algorithm. In *SIGGRAPH*, 1987.

[28] P. Mücke, R. Klowsky, and M. Goesele. Surface reconstruction from multi-resolution sample points. *Vision, Modeling, and Visualization (2011)*, 2011.

[29] R. A. Newcombe, S. Izadi, O. Hilliges, D. Molyneaux, D. Kim, A. J. Davison, P. Kohli, J. Shotton, S. Hodges, and A. W. Fitzgibbon. KinectFusion: Real-time dense surface mapping and tracking. *ISMAR*, 2011.

[30] R. A. Newcombe, S. J. Lovegrove, and A. J. Davison. DTAM: Dense Tracking and Mapping in Real-Time. *ICCV*, 2011.

[31] A. Samé, C. Ambroise, and G. Govaert. An online classification EM algorithm based on the mixture model. *Statistics and Computing*, 2007.

[32] S. M. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. *CVPR*, 2006.

[33] F. Steinbrücker, T. Pock, and D. Cremers. Large displacement optical flow computation without warping. *ICCV*, 2009.

[34] F. Steinbrücker, J. Sturm, and D. Cremers. Real-Time Visual Odometry from Dense RGB-D Images. In *ICCV Workshops*, 2011.

[35] G. Vogiatzis and C. Hernández. Video-based, real-time multi-view stereo. *IMAVIS*, 2011.

[36] M. Werlberger, W. Trobin, T. Pock, A. Wedel, D. Cremers, and H. Bischof. Anisotropic Huber-L1 optical flow. *BMVC*, 2009.

[37] C. Zach. Fast and high quality fusion of depth maps. *3DPVT*, 2008.

[38] C. Zach. Robust Bundle Adjustment Revisited. *ECCV*, 2014.

[39] M. Zollhöfer, C. Theobalt, M. Stamminger, M. Nießner, S. Izadi, C. Rehmann, C. Zach, M. Fisher, C. Wu, A. Fitzgibbon, and C. Loop. Real-time non-rigid reconstruction using an RGB-D camera. *ACM Transactions on Graphics*, 2014.