PatchMatch Based Joint View Selection and Depthmap Estimation

Enliang Zheng, Enrique Dunn, Vladimir Jojic, and Jan-Michael Frahm
The University of North Carolina at Chapel Hill
{ezheng,dunn,vjojic,jmf}@cs.unc.edu

Abstract

We propose a multi-view depthmap estimation approach aimed at adaptively ascertaining the pixel level data associations between a reference image and all the elements of a source image set. Namely, we address the question, what aggregation subset of the source image set should we use to estimate the depth of a particular pixel in the reference image? We pose the problem within a probabilistic framework that jointly models pixel-level view selection and depthmap estimation given the local pairwise image photoconsistency. The corresponding graphical model is solved by EM-based view selection probability inference and PatchMatch-like depth sampling and propagation. Experimental results on standard multi-view benchmarks convey the state-of-the art estimation accuracy afforded by mitigating spurious pixel-level data associations. Additionally, experiments on large Internet crowd sourced data demonstrate the robustness of our approach against unstructured and heterogeneous image capture characteristics. Moreover, the linear computational and storage requirements of our formulation, as well as its inherent parallelism, enables an efficient and scalable GPU-based implementation.

1. Introduction

Multi-view depthmap estimation (MVDE) methods strive to determine a view dependent depthfield by leveraging the local photoconsistency of a set overlapping images observing a common scene. Applications benefiting from high quality depthmap estimates include dense 3D modeling, classification/recognition [20] and image based rendering [6]. However, achieving highly accurate depthmaps is inherently difficult even for well controlled environments where factors such as viewing geometry, image-set color constancy, and optical distortions are rigorously measured and/or corrected. Conversely, practical challenges for robust depthmap estimation from non-controlled input imagery (*i.e*. Internet collected data) include mitigating heterogeneous resolution and scene illuminations, unstructured viewing geometry, scene content variability and image registration errors (*i.e*. outliers). Moreover, the increasing availability of crowd sourced datasets has explicitly brought efficiency and scalability to the forefront of application requirements, while implicitly increasing the importance of data association management when processing such large scale datasets.

The input for MVDE is commonly assumed to consist of a convergent set of images along with reliable estimates of their pose and calibration parameters. The extracted depthmap will correspond to the pixel-wise 3D structure hypotheses that best explain the available image observations in terms of some measure of visual similarity w.r.t. a reference image. Ironically, the potential robustness afforded by having multiple available images is compromised by the inherent variability in pairwise photoconsistency observations. In practice, correct depth hypotheses may provide low photoconsistency in a source image subset (e.g. occlusions or illumination aberrations), while incorrect depth hypotheses may register high image similarity (e.g. repetitive structure or homogeneous texture). These technical challenges render multi-view depth hypothesis evaluation as a problem of robust model fitting, where a demarcation among inlier and outlier photoconsistency observations is required. We tackle this implicit data association problem by addressing the question: *What aggregation subset of the source image set should be used to estimate the depth of a particular pixel in the reference image*.

We propose a probabilistic framework for depthmap estimation that jointly models pixel-level view selection and depthmap estimation given pairwise image photoconsistency. An overview is depicted in Figure 1. The corresponding graphical model is solved by EM-based view selection probability inference and PatchMatch-like depth sampling and propagation. Our approach iteratively alternates between exploration of the depth search space and updating our formulated probabilistic model. The insight leveraged by our method is the spatial smoothness in the photoconsistency at the correct depth hypothesis of a given pixel w.r.t. the images in the source image dataset [22, 13]. Our expectation of having a high overlap of photoconsistent source images among neighboring pixels in the reference

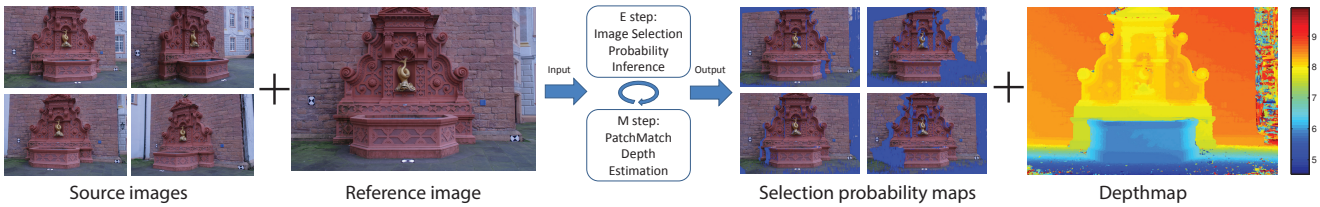


Figure 1. Overview of our approach. Input imagery is used to jointly estimate a depthmap and pixel level view associations. Blue regions in the view selection probacility map indicate pixels in the reference image lacking reliable observations in the corresponding source image.

image, leads to modeling the depth estimation problem as a Markov process where the unobserved states correspond to binary indicator variables for the selection probability of each source image.

We summarize the contributions and advantages of the framework as follows. **Accuracy:** Mitigation of spurious data associations at the pixel level provides state-of-the-art accuracy results for single depthmap estimation. **Efficiency:** Deployment of PatchMatch sampling and propagation enables reduced computational burden as well as GPU implementation. **Scalability:** Linear storage requirement w.r.t. the number of source images, as opposed to the exponential growth in the joint view selection and depth estimation model by Strecha *et al*. [22], enables handling selection instances comprising hundreds of images.

2. Related Work

Depthmap estimation handling occlusion firstly emerged in two view stereo [25, 24, 28]. In principle, the additional view redundancy available to MVDE can be leveraged to resolve occlusions. Kang *et al*. [17] explicitly address occlusion in multi-baseline stereo by only using the subset of the heuristically selected overlapping cameras with the minimum matching cost. The heuristic provides occlusion robustness as long as there is a sufficient number of unoccluded views (typically 50%). Campbell *et al*. [5] choose the best few depth hypotheses for each pixel, following with a MRF optimization to determine a spatially consistent depthmap. Their method chooses source images based on spatial proximity of cameras. Strecha *et al*. [21] handle occlusion in wide-baseline multi-view stereo by including visibility within a probabilistic model, where the depth smoothness is enforced on neighboring pixels according to the color gradient. The work of Strecha *et al*. [21] is further extended in [22] where the depth and visibility are jointly modeled by hidden Markov random fields. In [22] the memory used for visibility configuration of each pixel is 2^K, which grows exponentially with respect to the number of input images K. Hence, the approach is limited to very few images (three images in their evaluation). In contrast, our memory usage is linear with the number of images K. Gallup *et al*. [11] present a variable-baseline and variable-resolution framework for MVDE, exploring the attainment of pixel-specific data associations for capture from approximately linear camera paths. While that work illustrates the benefits of fine grain data association strategies in multi-view stereo, it does not easily generalize to irregularly captured datasets.

Lightweight depthmap fusion relies on the mutual depth consistency between multiple depthmaps. Shen [19] computes the depthmap for each image using PatchMatch stereo, and enforces depth consistency over neighboring views. Hu & Mordohai [15] follows a scheme similar to Camppbell *et al*. [5] but select the final depth through a process enforcing mutual consistency across all depthmaps. These methods require the depthmaps of other views to be available, while in contrast our method directly outputs an accurate depthmap. Some other methods aim at generating a consistent 3D model instead of depthmaps. Furukawa *et al*. ([10]) present an accurate Patch-based MVS approach that starts from a sparse set of matched keypoints, which were repeatedly expanded until visibility constraints are invoked to filter out false matches. Zaharescu *et al*. [29] propose a mesh evolution framework based on a new self-intersection removal algorithm. Jancosek *et al*. [16] propose a method that additionally reconstructs surfaces that do not have direct support in the input 3D points by exploiting visibility in 3D meshes. In contrast, our focus is on multi-view depthmap estimation.

Robust stereo performance for crowd sourced data is an ongoing research effort. Frahm *et al*. [8] discern a suitable input datum by using appearance clustering using a color augmented GIST descriptor along with feature-based geometric verification. Furukawa *et al*. [9] use structure from motion (SFM) to purge redundant imagery but retain high resolution geometry. Their iterative clustering merges sparse 3D points and cameras based on visibility analysis. Although intra-cluster image partitioning is not performed, the cluster size is limited in an effort to maintain computational efficiency. Goesele *et al*. [13] address the viewpoint selection for crowd sourced imagery by building small size image clusters using the cardinality of the set of common features among viewpoints and a parallax-based metric. Images were resized to the lowest common resolution in the cluster. Pixel depth is then computed using four images selected from the cluster based on local color consistency. As

our experiments will show, image wide selection may not robust to outlier pose estimates.

The recently proposed PatchMatch is incorporated in our method as an efficient sampling scheme. The PatchMatch was firstly introduced to solve the two view stereo problem in [4]. PatchMatch initializes each pixel with a random slanted plane at random depth, and is followed by the propagations. The nearby and the current pixels' slanted planes are tested and the one with the best cost is kept. Besse *et al*. [2] combine the PatchMatch sampling scheme and belief propagation to infer an MRF model that has smoothness constraints. While the original PatchMatch stereo was a sequential method, Bailer *et al*. [1] parallelize the algorithm by restricting the propagations to only horizontal and vertical directions. We further explore the potential of PatchMatch in wide baseline stereo with large hypotheses space.

3. Joint View Selection and Depth Estimation

In this section we provide an overview of our PatchMatch propagation scheme (§3.1), describe our probabilistic graphic model (§3.2), describe our variational inference approximation to the model's posterior probability (§3.3 and §3.4) and finalize describing our implementation (§3.5).

3.1. PatchMatch Propagation for Stereo

Our algorithm uses single oriented planes instead of the multiple oriented in [1], to reduce the three-dimensional search space (depth and two angles for the orientated plane) to one dimension. We alternatively perform upward/downward propagations during the odd iterations and perform rightward/leftward propagations during even iterations. To calculate the depth at pixel (i, j) for the rightward propagation, only the depth at positions $(i, j-1)$ and (i, j) are tested on pixel (i, j) (Fig. 2). Likewise, only one neighbor is considered for all other propagations. The propagation schemes of [4] and [1] are shown in Fig. 2.

In the absence of proper depth hypotheses, we additionally draw and test H random depth hypotheses for each pixel during propagations. We use $H = 1$ and have 3 depth hypotheses tested per pixel in a propagation, i.e. the depths of current and the neighboring pixel along with one random depth. Without loss of generality, we limit our discussion henceforth to the rightward horizontal propagation.

3.2. Graphical Model

In our algorithm, the depth is estimated for a reference image X^{ref}, given a set of M (unstructured) source images $X^1, X^2, ...X^M$ with known camera calibration parameters, which are the output of a typical structure from motion system such as VisualSFM[27]. We denote the correct depth associated with each pixel l on image X^{ref} as θ_l.

Photo-consistency values for the correct depth of a given pixel across a set of source images may be incongruent for

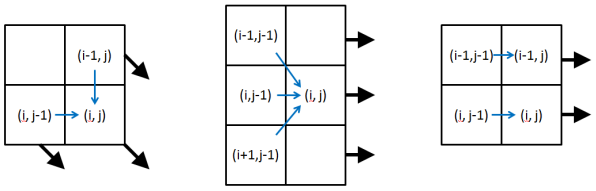


Figure 2. The black and blue arrows show the propagation directions and the sampling schemes. Left: Top left to bottom right propagation in [4]. Middle: Rightward propagations in [1]. Right: Our rightward propagation.

some of the source images. This may be attributed to a diversity of factors such as occlusions, calibration errors, illumination aberration, etc. Therefore, depth estimation for a given pixel entails the determination of which subset of source images will provide the most robust estimate. Our model defines M binary variables $Z_l^m \in \{0, 1\}, m = 1, 2...M$ for each pixel l in the reference image X^{ref}, where Z_l^m is 1 if image X^m is selected for depth estimation of pixel l, and 0 otherwise.

We first define the likelihood function. We denote the color patch centered at pixel l in the reference image as X_l^{ref}. Given a pixel l and its correct depth θ_l in the reference image X^{ref}, a color patch X_l^m on source image m can be determined through homography warping [19]. If $Z_l^m = 1$, the probability that the observed color patch X_l^m is color-consistent with X_l^{ref} should be high. We use NCC (normalized cross correlation) to compare the two color patches X_l^m and X_l^{ref} as a robust proxy to single pixel comparisons, and denote the NCC measurement as ρ_l^m. In the case when $Z_l^m = 0$, X_l^m has arbitrary colors due to factors such as occlusion or calibration errors, so the probability of observing X_l^m is unrelated to X_l^{ref} and considered uniformly distributed. Therefore we propose the following likelihood function

$$P(X_l^m|Z_l^m, \theta_l, X_l^{\text{ref}}) = \begin{cases} \frac{1}{NA} e^{-\frac{(1-\rho_l^m)^2}{2\sigma^2}} & \text{if } Z_l^m = 1 \\ \frac{1}{N}\mathcal{U} & \text{if } Z_l^m = 0, \end{cases} \quad (1)$$

where A equals to $\int_{-1}^{1} exp\{-\frac{(1-\rho)^2}{2\sigma^2}\}d\rho$ and N is a constant. Note that NCC value ranges in $[-1, 1]$ and equals 1 with the best color consistency. Consistent with our intuition, a color patch X_l^m with high NCC value ρ_l^m has high probability $P(X_l^m|Z_l^m = 1, \theta_l, X_l^{\text{ref}})$. $\mathcal{U}$ is the uniform distribution in the range $[-1, 1]$ with probability density 0.5. Note that NCC computation is affine invariant and multiple pairs of color patches can generate the same NCC value. To simplify the analysis without affecting depthmap quality, Eq. (1) assumes the number of color patches X_l^m that can generate any specific NCC value is the same and equals to N. Since only the ratio $P(X_l^m|Z_l^m = 1, \theta_l, X_l^{\text{ref}})/P(X_l^m|Z_l^m = 0, \theta_l, X_l^{\text{ref}})$ matters in the model

inference discussed in §3.3 and §3.4, we can safely ignore the constant N in Eq. (1).

In Eq. (1) σ is the parameter determining the suitability of an image based on NCC measurement ρ_l^m. As seen in Fig. 3(b) a soft threshold τ is determined by σ. If ρ_l^m is larger than τ, it is more likely that image m is selected, and vice versa. Since X_l^{ref} is observed for each pixel, $P(X_l^m|Z_l^m, \theta_l, X_l^{\text{ref}})$ is simply denoted as $P(X_l^m|Z_l^m, \theta_l)$ in the rest of the paper.

The depths of nearby pixels are considered independent, while the pairwise smoothness is put on the nearby selection variables along the current propagation direction (Fig. 3(a)) through the transition probabilities:

$$P(Z_l^m|Z_{l-1}^m) = \begin{pmatrix} \gamma & 1-\gamma \\ 1-\gamma & \gamma \end{pmatrix}. \tag{2}$$

Setting γ close to 1 encourages neighboring pixels to have similar selection preference for source images X^m. To enable parallel computation, we only enforce pairwise constraint on the pixels of the same row in the horizontal propagations. Note Fig. 3(a) only shows one row of selection variables for each of the source images.

Finding the optimal selection $\boldsymbol{Z}$ and depth $\boldsymbol{\theta}$ given all the images $\boldsymbol{X}$ equates to computing the maximum of the posterior probability (MAP) $P(\boldsymbol{Z}, \boldsymbol{\theta}|\boldsymbol{X})$. The Bayesian approach firstly computes the joint probability based on the graphical model (Fig. 3(a)) and normalizes over $P(\boldsymbol{X})$. The joint probability is

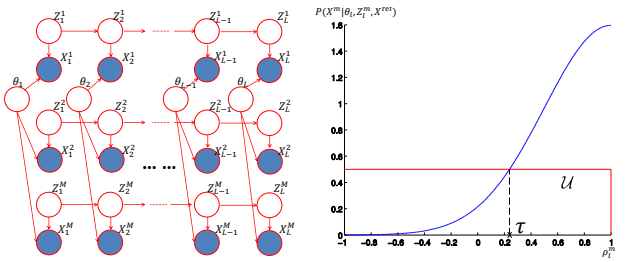
$$P(\boldsymbol{X}, \boldsymbol{\theta}, \boldsymbol{Z}) = \tag{3}$$
$$\prod_{m=1}^{M} [P(Z_1^m) \prod_{l=2}^{L} P(Z_l^m|Z_{l-1}^m) \prod_{l=1}^{L} P(X_l^m|Z_l^m, \theta_l)] \prod_{l=1}^{L} P(\theta_l),$$

where L is the number of pixels along the propagation direction of the reference image. We use an uninformative uniform distribution for prior $P(Z_1^m)$ as well as depth prior $P(\theta_l)$ since we have no preference without observations. However, computing $P(\boldsymbol{X})$ is intractable as it requires to sum over all possible values of $\boldsymbol{Z}$ and $\boldsymbol{\theta}$.

We interleave pixel level inference of image selection probability with fixed depth, and depth updating with fixed image selection probability. Our approach is a variant of the generalized EM (GEM)[18]. Similarly to [18], we use variational inference theory to justify our algorithm.

3.3. Variational Inference

Variational inference is to consider a *restricted* family of distributions $q(\boldsymbol{Z}, \boldsymbol{\theta})$ and then seek the member of this family to approximate the real posterior distribution $P(\boldsymbol{Z}, \boldsymbol{\theta}|\boldsymbol{X})$, in the sense that the KL divergence between these two is minimized [3]. *The restriction is imposed purely to achieve tractability.* The real posterior distribution is over the set of unobserved variables $\boldsymbol{\theta} = \{\theta_l|l = 1, ..., L\}$ and $\boldsymbol{Z} = \{\boldsymbol{Z}^m|m = 1, ..., M\}$, where $\boldsymbol{Z}^m = \{Z_1^m, Z_2^m, ..., Z_L^m\}$ is a chain in the graph. We put restrictions on the family of distributions $q(\boldsymbol{Z}, \boldsymbol{\theta})$, assuming that it is factorizable into a set of distributions ([3]):



(a) The graphical model (b) Distribution of Eq. (1)

Figure 3. (a) θ_l is the depth of pixel l. Z_l^m is the selection of image m at pixel l. X_l^m is the observation (colors) on the source image m given depth θ_l.

$$q(\boldsymbol{Z}, \boldsymbol{\theta}) = \prod_{m=1}^{M} q_m(\boldsymbol{Z}^m) \prod_{l=1}^{L} q_l(\theta_l). \tag{4}$$

For tractability, we further constrain each $q_l(\theta_l)$, $l = 1, 2, ..., L$ to the family of Kronecker delta functions:

$$q_l(\theta_l) = \delta(\theta_l = \theta_l^*) = \begin{cases} 1, & \text{if } \theta_l = \theta_l^* \\ 0, & \text{otherwise} \end{cases} \tag{5}$$

where θ_l^* is a parameter to be estimated. This assumption is in contrast to most other works [21, 22, 25, 24], which discretize the depth as a means to recover the whole posterior distribution of the depth. Once the distribution $q_l(\theta_l)$ is determined, θ_l is set to θ_l^* to maximize the approximate posterior distribution Eq. (4), so θ_l^* is actually the final estimated depth. Conversely, the depths $\boldsymbol{\theta}$ can be considered as parameters shared by different chains instead of as variables. This assumption seamlessly combines the PatchMatch sampling scheme in the graph model inference.

The variational method seeks to find a member $q^{\text{opt}}(\boldsymbol{Z}, \boldsymbol{\theta}) = \prod_{m=1}^{M} q_m^{\text{opt}}(\boldsymbol{Z}^m) \prod_{l=1}^{L} q_l^{\text{opt}}(\theta_l)$ from the family $q(\boldsymbol{Z}, \boldsymbol{\theta})$, minimizing the KL divergence between $q(\boldsymbol{Z}, \boldsymbol{\theta})$ and $P(\boldsymbol{Z}, \boldsymbol{\theta}|\boldsymbol{X})$ under the constraint that $q_m(Z^m), m = 1, ...M$ are normalized ($q_l(\theta_l)$ is guaranteed to be normalized as it is constrained to be a Kronecker delta function):

$$\begin{aligned} &\underset{q(\boldsymbol{Z}, \boldsymbol{\theta})}{\text{minimize}} \quad \text{KL}(q(\boldsymbol{Z}, \boldsymbol{\theta})||P(\boldsymbol{Z}, \boldsymbol{\theta}|\boldsymbol{X})) \\ &\text{subject to} \quad \sum\nolimits_{Z^m} q_m(Z^m) = 1, m = 1, \ldots, M. \end{aligned} \tag{6}$$

Note the optimization is performed over distributions, but not over variables. To optimize over $q_m(\boldsymbol{Z}^m)$, the standard solution [3] is $\log(q_m(\boldsymbol{Z}^m)) = \mathbb{E}_{\setminus m}[\log(P(\boldsymbol{X}, \boldsymbol{\theta}, \boldsymbol{Z}))] + const$, where $\mathbb{E}_{\setminus m}$ is the expectation of $\log(P(\boldsymbol{X}, \boldsymbol{\theta}, \boldsymbol{Z}))$ taken over all variables not in $q_m(\boldsymbol{Z}^m)$ [3]. Then we have

$$q_m^{\text{opt}}(\boldsymbol{Z}^m) \propto \Psi(\boldsymbol{Z}^m) \prod_{l=1}^{L} P(X_l^m|Z_l^m, \theta_l = \theta_l^*), \tag{7}$$

where $\Psi(\boldsymbol{Z}^m)$=$P(Z_1^m)\prod_{l=2}^{l=L} P(Z_l^m|Z_{l-1}^m)$. The right side of Eq. (7) has form of joint probability of a Hidden Markov Chain with fixed transition probability from Eq. (2) and fixed emission probability Eq. (1). The probability of each hidden variable $q(Z_l^m)$ can be efficiently inferred by forward-backward algorithm [3]. See §3.4 for more details. This corresponds to the E step of the GEM algorithm.

To optimize over $q_l(\theta_l)$ we seek an optimal parameter θ_l^{opt} for the distribution $q_l(\theta_l)$ that minimizes Eq. (6). Suppressing the terms not involving θ_l gives

$$\theta_l^{\text{opt}} = \underset{\theta_l^*}{\text{argmax}} \sum_{m=1}^{M} q(Z_l^m{=}1) \ln P(X_l^m|Z_l^m{=}1, \theta_l{=}\theta_l^*). \quad (8)$$

By substituting Eq. (1) into Eq. (8), we get

$$\theta_l^{\text{opt}} = \underset{\theta_l^*}{\text{argmin}} \sum\nolimits_{m=1}^{M} q(Z_l^m = 1)(1-\rho_l^m)^2, \quad (9)$$

where ρ_l^m is a function of θ_l^*. To find θ_l^{opt} in the above equation, 3 depth hypotheses sampled based on PatchMatch are tested, and the one that maximizes Eq. (9) is assigned to the parameter of the distribution $q_l(\theta_l)$. This step is the M step of the GEM algorithm. Note that the righthand side of Eq. (9) is a weighted sum of $(1-\rho_l^m)^2$ with weight equal to the image selection probability. Hence, a small value of $q(Z_l^m = 1)$, designating image m as not favorable, contributes less in evaluating the parameter θ_l^*.

Improvement: Eq. (9) is computationally expensive for hundreds of source images. Based on Eq. (9), it is unnecessary to compute ρ_l^m if the corresponding image selection probability $q(Z_l^m = 1)$ is very small. Hence, we propose a Monte Carlo based approximation [3]. Rewriting Eq. (9) as

$$\theta_l^{\text{opt}} = \underset{\theta_l^*}{\text{argmin}} \sum\nolimits_{m=1}^{M} P(m)(1-\rho_l^m)^2 \quad (10)$$

where the new distribution $P(m) = \frac{q(Z_l^m=1)}{\sum_{m=1}^{M} q(Z_l^m=1)}$ can be deemed as the probability of image m being the best for depth estimation of pixel l. We draw samples based on the distribution $P(m)$ to obtain a subset S, then

$$\theta_l^{\text{opt}} = \underset{\theta_l^*}{\text{argmin}} \frac{1}{|S|} \sum\nolimits_{m\in S} (1-\rho_l^m)^2. \quad (11)$$

Empirically, 15 samples suffice to attain good results.

Both distributions $q_m^{\text{opt}}(\boldsymbol{Z})$ and $q_l^{\text{opt}}(\theta_l)$ are coupled. The computation of θ_l^* requires $q(Z_l^m)$ to be known (Eq. (9)), but to infer $q(Z_l^m)$ in Eq. (7), we need θ_l^* available. The next subsection introduces the update scheme that computes the distributions iteratively.

3.4. Update Schedule

The common way to compute approximate distributions is coordinate descent optimization method. Namely, one distribution is optimized while other distributions remain fixed. Choosing which distribution to optimize over in each step is arbitrary or scheduled based on application, but it always decreases the cost function in Eq. (6). We choose to interleave updates of $q_l(\theta_l)$ and $q_m(\boldsymbol{Z}^m)$ as it is able to quickly propagate the correct depth into nearby pixels. For clarity, our explanations below use one chain and omit the image index m for each variable.

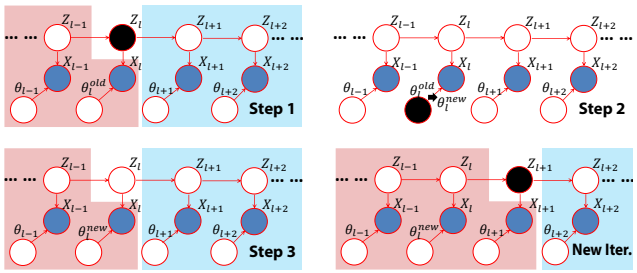


Figure 4. Update schedule. See text for more details.

For more details on Hidden Markov Chain inference, we refer the reader to text [3]. The forward-backward algorithm is used to infer the probability of hidden variables Z_l.

$$q(Z_l) = \frac{1}{A}\alpha(Z_l)\beta(Z_l), \quad (12)$$

where A is the normalization factor. $\alpha(Z_l)$ and $\beta(Z_l)$ are the forward and backward message for variable Z_l computed using the following Equations,

$$\alpha(Z_l) = p(X_l|Z_l,\theta_l) \sum_{Z_{l-1}} \alpha(Z_{l-1})P(Z_l|Z_{l-1}), \quad (13)$$

$$\beta(Z_l) = \sum_{Z_{l+1}} \beta(Z_{l+1})P(X_{l+1}|Z_{l+1},\theta_{l+1})P(Z_{l+1}|Z_l). \quad (14)$$

Both the forward and backward messages are computed recursively (e.g. $\alpha(Z_l)$ is computed using $\alpha(Z_{l-1})$). In Fig. 4, the variables covered in red area and blue area contribute to the forward and backward messages respectively.

We perform the following update schedule as is shown in Fig. 4. In step 1, compute $q(Z_l)$ using Eq. (12), (13) and (14) for each source image (*i.e.* $q(Z_l^m)$,$m = 1...M$). In step 2, update the depth from θ_l^{old} to θ_l^{new} using Eq. (9) or Eq. (11). In step 3, with θ_l^{new}, we recompute forward message $\alpha(Z_l)$, which is further used to compute $\alpha(Z_{l+1})$ recursively in Eq. (13). Next we start at variable Z_{l+1} with the same process until reaching the end of the row in the image. Before the update process, the backward message for each variable can be computed recursively (Eq. (14)) and stored in memory.

3.5. Algorithm Integration

We now describe the computational framework implementing our depth estimation and view selection formula-

Input: All images, depthMap (randomly initialized or from previous propagation) **Output**: Updated depthMap m – image index, l – pixel index		
	Eq.	Step
For $l = L$ to 1		
For $m = 1$ to M		
Compute backward message β_l^m	(14)	1
For $l = 1$ to L		
For $m = 1$ to M		
Compute forward message α_l^m	(13)	1
Compute $q(Z_l^m)$	(12)	1
Draw depth hypotheses by PatchMatch		
Estimate θ_l^* for $q_l(\theta_l)$	(9 / 11)	2
For $m = 1$ to M		
Recompute forward message α_l^m	(13)	3

Table 1. The algorithm of a row/column propagation.

tion. The depthmap is initialized with random values within the depth range. Alternatively, sparse 3D measurements may be included within our initialization. Next, the rightward, downward, leftward and upward propagations are applied in sequence. Each propagation (except in the first iteration) uses the depth results of the former propagation. Within each propagation, updates of the depth and the selection probability are interleaved as described in §3.4. After two or three sweeps, each containing the four direction propagations, the depthmap reaches a stable state. Convergence may alternatively be verified through tracking the number of modified depth estimates up to a threshold. As each row is independent from other rows given our graphical model and processed in exactly the same way during one propagation, it can be easily parallelized for leveraging GPUs. We describe the algorithm for processing one row within rightward propagation in Table 1.

Discussion. The estimation of the exact image-wide MAP for our graphical model would require a Hidden Markov Random Field (MRF) formulation instead of our Hidden Markov Chain approximation. Our choice of using propagation direction specific chain models was driven by computational efficiency/tractability. The proposed framework enables us to easily interleave the propagation with hidden variable inference while fostering implementation parallelism. The enforcement of smoothness constraints on the hidden variables enables non-oscillating behavior of our evolving depth estimates. Our PatchMatch based framework has linear computational and storage complexity w.r.t. to input data size while being independent of the size of the depth search space. Namely, since the number of tested depth hypotheses (3 for each propagation) is small and constant, the computation complexity of our method is $O(WHM)$, where W, H, and M are the width, height and number of images. Methods using complete hypotheses search, e.g. [25, 22], require $O(WHMD)$ computations,

	2cm	10cm	2 cm	10cm
Error	fountain-P11		Herzjesu-P9	
Ours	0.732	0.911	0.619	0.833
Ours(P)	0.769	0.929	0.650	0.844
LC[15]	0.754	0.930	0.649	0.848
FUR[10]	0.731	0.838	0.646	0.836
ZAH[29]	0.712	0.832	0.220	0.501
TYL[26]	0.732	0.822	0.658	0.852
JAN[16]	0.824	0.973	0.739	0.923

Table 2. The percentage of pixels with absolute error less than 2cm and 10cm. Entries *Ours(P)* and *Ours* denote our results with and without postprocessing. Reported values are from [15])

where D is the size of hypotheses space normally reaching up to thousands of hypotheses.

4. Experiments

We evaluate the accuracy of our method on standard ground truth benchmarks and highlight our robustness on multiple crowd sourced datasets. In both evaluation scenarios we juxtapose our results with current state-of-the-art methods. We implemented our method in CUDA and executed on an Nvidia GTX-Titan GPU. For all experiments, the total number of multi-directional propagations is set to 3 and we use $\sigma = 0.45$ in the likelihood function (Eq. (1)) and $\gamma = 0.999$ in the transition probabilities (Eq. (2)).

Ground truth evaluation. We evaluated on the Strecha datasets (Fountain-P11 and Herzjesu-P9) [23] as they include ground truth 3D structure measurements. We use all dataset images at full resolution, set the NCC patch size to 15 by 15, and approximate the depth range from sparse 3D points. We measure pixel-wise depth errors as our goal is to generate a single depthmap instead of one consistent 3D scene model. We calculate the number of pixels with a depth error less than 2cm and 10cm from the ground truth and compare with [15, 10, 29, 26, 16]. All the pixels with accessible ground truth depth are evaluated to convey both the accuracy and the completeness of the estimated depthmaps. We omit evaluation of the dataset's two extremal views as done in [15].

We use slanted planes of a single orientation instead of fronto-parallel planes [12]. The single dominant orientation direction can be estimated by projecting sparse 3D points onto the ground plane as described in [12]. We further apply two optional depthmap refinement schemes to increase the final accuracy. Our basic depth refinement uses a smaller NCC patch (5x5), while eliminating random depth sampling, during an additional propagation sweep. We then use deterministic fine-grain sampling (20 hypotheses) in the depth neighborhood (± 1 cm) of each pixel's depth estimate as proposed in [19]. Finally, a median filter of size 9x9 is applied to each raw depthmap. Table 2 shows our method is

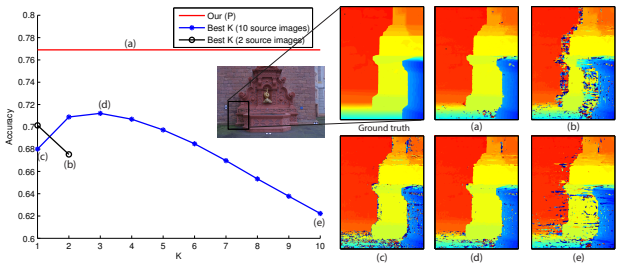


Figure 5. Left: Comparison against best-K aggregation. Right: Raw depthmap output of a partially occluded subregion with results for different dataset-aggregation combinations.

comparable to the state-of-the-art methods. Note the results of [15, 26, 16] are obtained through multi-depthmap fusion, while our method directly estimates individual depthmaps.

Advantages of pixel level view selection. Figure 5 shows our comparison to the occlusion-robust best-K planesweeping method [17], where for a given depth hypothesis, the cost is the average of the best K costs, with K being predefined. When K is set to the number of source images, it degenerates to the basic planesweeping algorithm that computes the cost using all source images. We compute depthmaps of the fountain-P11 data with varying K and otherwise fixed parameters, using 2000 planes. The percentage of pixels within 2cm difference from the ground truth is taken as a measure of the error. We run the planesweeping using two different dataset types. In the first case, all 10 source images are used. Alternatively, we use the neighboring left and the right images. Fig. 5 shows our results outperform all fixed aggregation schemes and illustrates the raw depthmap output of a partially occluded subregion.

Run times for our method are compared with optimized GPU planesweeping code. Fig. 7(a) shows the linear dependence of computation time to the number of planes, as well the diminishing accuracy improvements provided by increasing the search space resolution. Our PatchMatch sampling and propagation scheme only requires depth range specification, foregoing explicit search space discretization.

Robustness to noisy SFM estimates. The advantage of pixel-level view selection across the entire dataset is highlighted in Fig. 6, where we compare our results for corrupted SFM estimates against those obtained using the approach in [13]. Fig. 6 depicts Alexander Nevsky Cathedral in Sofia having indistinguishable structure in the tower structure (*i.e*. view invariant appearance due to structural symmetry). A set of 136 images, comprised by two mutually exclusive subsets observing the front or back, was fed into VisualSFM [27] yielding a corrupted 3D model where symmetric structure is fused along with the disjoint camera clusters. The approach in [13] initially selects a global subset of 20 images based on the corrupted SFM estimates and select independently for each pixel's depth estimation a fixed number (typically 4) of images from the global subset (similar to using K-best aggregation with K=4). If the global subset is unbalanced or is contaminated by corrupted estimates, the completeness of the model is compromised, as shown in Figure 6 where the background dome is missing. We consider the entire dataset and implicitly mitigate such outliers. Moreover, we re-executed [13] with manually filtered camera poses and indeed achieved correct results.

Figure 6. Top: Front and back of Alexander Nevsky Cathedral and estimated 3D model. Bottom: original image, depthmap of our method and [13] with wrong and correct camera poses.

Robustness to varying capture characteristics. We tested our algorithm on Internet photo collections (IPC) downloaded from the Flickr for six different scenes: Paris Triumphal Arch (195 images), Brandenburg Gate (300 images), Notre Dame de Paris (300 images), Great Buddha (212 images), Mt. Rushmore (206 images), and Berlin Cathedral (500 images). In order to control GPU memory, we optionally resize imagery to no more than 1024 pixels for each dimension. Camera poses were calculated using VisualSFM [27]. The average run time for Berlin Cathedral is 98.3 secs/image. For illustration, sky region pixels are masked out using [7] as post-processing. To compare with Goesele's method [13], we run the author's code on the same dataset with default parameters except for setting the matching window size to the same as ours (7x7). The results shown in Fig. 8 illustrate that, while both approaches are robust to wide variations in illumination, scale and scene occlusions across the datasets, our approach tends to provide increased completeness of depthmap estimates. We attribute this to our more flexible view selection framework. In contrast to [13], we avoid making initial hard image discriminations through an initial global image subset.

To quantitatively compare the accuracy of our results with [13], in the absence of ground truth geometry for crowd sourced datasets, we revisit the accuracy of both methods in the Strecha Fountain dataset. The method in [13] rejects outlier depth estimates based on the NCC values and the viewing angles. Hence, we only compare the accuracy of the reliable pixels as classified by [13] (comprising 75.4% of total image pixels). Figure 7(b) shows our approach outperforming both [13] and planesweep for high accuracy thresholds. We expect the same accuracy ranking to carry over to the crowd sourced data results.

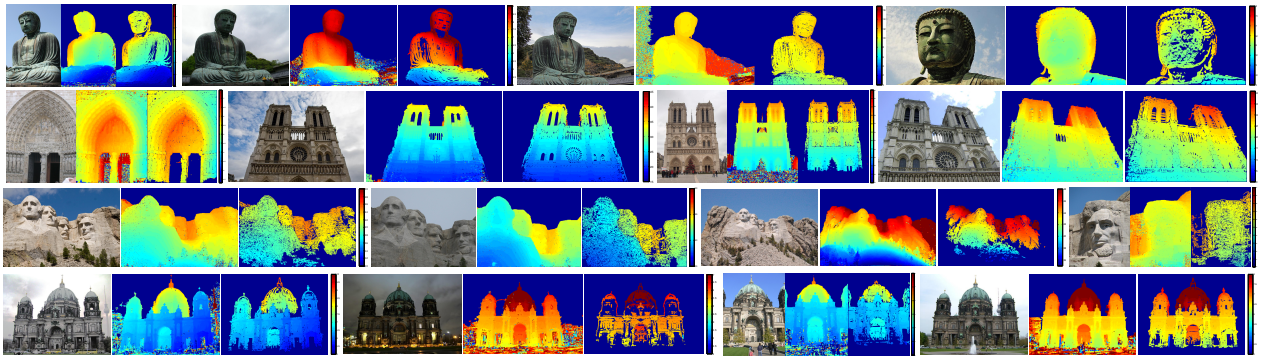


Figure 8. Each image triplet depicts a reference image along with our and Goesele's ([13]) depthmap output (Best viewed in color).

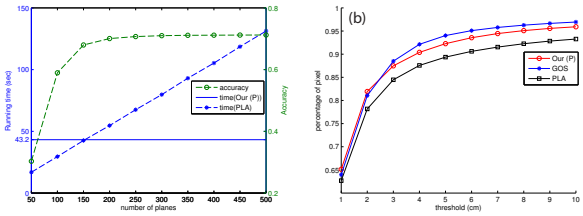


Figure 7. Fountain dataset performance. Left: Average running time. Right: Percentage of pixels given different thresholds. PLA is the planesweep algorithm with all source images and K=3, while GOS is the method in [13].

5. Future Work

We presented an efficient and effective method for joint view selection and depthmap estimation. Future research direction includes integrating online plane normal estimation for each pixel. We will explore the use of more sophisticated filtering mechanisms such as the one presented in [14] to further improve both efficiency and accuracy.

Acknowledgement. This work was supported by NSF IIS-1349074 and NSF IIS-1252921.

References

[1] C. Bailer, M. Finckh, and H. P. A. Lensch. Scale robust multi view stereo. In *ECCV*, 2012.

[2] F. Besse, C. Rother, and J. Kautz. Pmbp: Patchmatch belief propagation for correspondence field estimation. In *BMVC*, 2012.

[3] C. M. Bishop. *Pattern Recognition and Machine Learning*. Springer-Verlag New York, Inc, NJ, USA, 2006.

[4] M. Bleyer, C. Rhemann, and C. Rother. Patchmatch stereo - stereo matching with slanted support windows. In *BMVC*, 2011.

[5] N. D. F. Campbell, G. Vogiatzis, C. H. Esteban, and R. Cipolla. Using multiple hypotheses to improve depthmaps for multi-view stereo. In *ECCV*, 2008.

[6] S. E. Chen and L. Williams. View interpolation for image synthesis. In *Proceedings of the 20th annual conference on Computer graphics and interactive techniques*, 1993.

[7] M. H. Derek Hoiem, Alexei A. Efros. Geometric context from a single image. In *ICCV*, 2005.

[8] J.-M. Frahm, P. Fite-Georgel, D. Gallup, T. Johnson, R. Raguram, C. Wu, Y.-H. Jen, E. Dunn, B. Clipp, S. Lazebnik, and M. Pollefeys. Building rome on a cloudless day. In *ECCV*. 2010.

[9] Y. Furukawa, B. Curless, S. M. Seitz, and R. Szeliski. Towards Internet-scale multi-view stereo. In *CVPR*, 2010.

[10] Y. Furukawa and J. Ponce. Accurate, dense, and robust multiview stereopsis. In *PAMI*, 2010.

[11] D. Gallup, J.-M. Frahm, P. Mordohai, and M. Pollefeys. Variable baseline/resolution stereo. In *CVPR*, 2008.

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

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

[14] A. Hosni, C. Rhemann, M. Bleyer, C. Rother, and M. Gelautz. Fast cost-volume filtering for visual correspondence and beyond. *PAMI*, 2012.

[15] X. Hu and P. Mordohai. Least commitment, viewpoint-based, multi-view stereo. In *3DIMPVT*, 2012.

[16] M. Jancosek and T. Pajdla. Robust, accurate and weaklysupported-surfaces preserving multi-view reconstruction. In *CVPR*, 2011.

[17] S. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *CVPR*, 2001.

[18] R. Neal and G. E. Hinton. A view of the EM algorithm that justifies incremental, sparse, and other variants. In *Learning in Graphical Models*, 1998.

[19] S. Shen. Accurate multiple view 3d reconstruction using patch-based stereo for large-scale scenes. In *TIP*, 2013.

[20] J. Shotton, A. Fitzgibbon, M. Cook, T. Sharp, M. Finocchio, R. Moore, A. Kipman, and A. Blake. Real-time human pose recognition in parts from single depth images. In *CVPR*, 2011.

[21] C. Strecha, R. Fransens, and L. V. Gool. Wide-baseline stereo from multiple views: a probabilistic account. In *CVPR*, 2004.

[22] C. Strecha, R. Fransens, and L. V. Gool. Combined depth and outlier estimation in multi-view stereo. In *CVPR*, 2006.

[23] C. Strecha, W. von Hansen, L. V. Gool, P. Fua, and U. Thoennessen. On benchmarking camera calibration and multi-view stereo for high resolution imagery. In *CVPR*, 2008.

[24] J. Sun, Y. Li, S. B. Kang, and H.-Y. Shum. Symmetric stereo matching for occlusion handling. In *CVPR*, 2005.

[25] J. Sun, H.-Y. Shum, and N.-N. Zheng. Stereo matching using belief propagation. In *ECCV*, 2002.

[26] R. Tylecek and R. Sara. Refinement of surface mesh for accurate multi-view reconstruction. In *Int'l Journal of VR*, 2010.

[27] C. Wu. Visualsfm: A visual structure from motion system. In *http://homes.cs.washington.edu/ ccwu/vsfm/*, 2011.

[28] J. Xiao, J. Chen, D.-Y. Yeung, and L. Quan. Learning two-view stereo matching. In *ECCV*, 2008.

[29] A. Zaharescu, E. Boyer, and R. P. Horaud. Topologyadaptive mesh deformation for surface evolution, morphing, and multi-view reconstruction. In *PAMI*, 2011.