Quasi-Dense 3D Reconstruction using Tensor-based Multiview Stereo*

Tai-Pang Wu[†‡] Sai-Kit Yeung[†] Jiaya Jia[‡] Chi-Keung Tang[†]

[†] The Hong Kong University of Science and Technology
[‡] The Chinese University of Hong Kong

Abstract

We propose tensor-based multiview stereo (TMVS) for quasi-dense 3D reconstruction from uncalibrated images. Our work is inspired by the patch-based multiview stereo (PMVS), a state-of-the-art technique in multiview stereo reconstruction. The effectiveness of PMVS is attributed to the use of 3D patches in the match-propagate-filter MVS pipeline. Our key observation is: PMVS has not fully utilized the valuable 3D geometric cue available in 3D patches which are oriented points. This paper combines the complementary advantages of photoconsistency, visibility and geometric consistency enforcement in MVS via the use of 3D tensors, where our closed-form solution to tensor voting provides a unified approach to implement the match-propagate-filter pipeline. Using PMVS as the implementation backbone where TMVS is built, we provide qualitative and quantitative evaluation to demonstrate how TMVS significantly improve the MVS pipeline.

1. Introduction

Match-propagate-filter is a competitive approach to multiview stereo reconstruction for computing a (quasi) dense representation. Starting from a sparse set of initial matches with high confidence, matches are propagated using photoconsistency to produce a (quasi) dense reconstruction of the target shape. Visibility consistency can be applied to remove outliers.

Among the existing works using the match-propagate-filter approach, patch-based multiview stereo (or PMVS) proposed in [5, 6] has produced some of the best results to date. The central idea of PMVS is the use of 3D patches in the match-propagate-filter pipeline, which is more effective than operating in the 2D domain, fitting local planes, or adopting simplified assumptions such as homography [16, 8]. In particular, PMVS's propagation step (or expansion) contributes a lot to the excellent results produced. Starting with sparse geometry, PMVS effectively used photoconsistency and visibility consistency to process unmatched regions in the matching step. The propagated 3D patch coordinates were shown to be very accurate [22] due to their effective enforcement of photoconsistency and visibility.

We observe however that PMVS did not fully utilize the 3D information inherent in the sparse and dense geometry before, during and after propagation, as patches do not adequately communicate among each other. As noted in [5], this communication should not be done by smoothing, but the lack of communication will cause perturbed surface normals and more patch outliers during propagation even for simple geometry (Figure 5).

This paper proposes tensor-based multiview stereo (TMVS) and uses 3D tensors which communicate among each other via a closed-form solution to tensor voting [28]. We found that such tensor communication not only improves propagation in MVS without undesirable smoothing but also benefits the entire match-propagate-filter pipeline within a unified framework (Figure 1):

- *Match.* In the uncalibrated scenario, robust parameter estimation employing the closed-form tensor voting effectively discards epipolar geometries induced by wrong matches, such as similar points on two different sides on the same object. In the calibrated scenario, TMVS produces better 3D normals than PMVS by 3D tensors communication.
- *Propagate.* Tensor communication enables better surface normals reconstruction by combining photoconsistency, visibility and geometry information. This significantly improves tensor propagation in the 3D space without using visual hulls.
- *Filter.* When needed, tensor voting can be deployed to remove outliers after the propagation process. The energy function turns out to be a quadratic optimization and thus can be solved efficiently using Gauss-Seidel method.

We believe these improvements are quite significant. Using PMVS as the implementation backbone where TMVS is built, we focus on quantitative evaluation on *normal* reconstruction accuracy, and refer readers for location recon-

*The research was supported by the Hong Kong Research Grant Council under grant nos. 620309, 619208, and 412307.

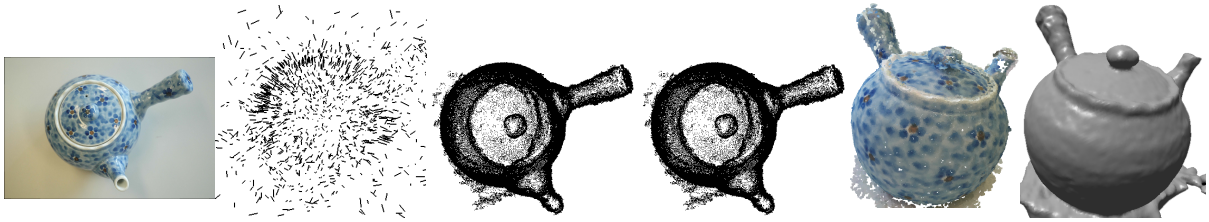


Figure 1. From left to right: input image, initial patches, propagated patches, filtered patches (not necessary here), the quasi-dense reconstruction, and one view of the reconstructed surface.

struction accuracy to [22]. The quasi-dense reconstruction produced by TMVS can be deployed in existing surface reconstruction, such as [13, 5] to produce a surface representation. While related to quasi-dense 3D reconstruction, surface reconstruction is not our focus. Rather than comparing reconstructed surfaces, we directly compare patch normals and normals from tensors, the raw outputs of PMVS and TMVS. Like PMVS, TMVS is available in C++ source codes (in the supplemental material) which include the implementation of the closed-form tensor voting.

2. Related Work

Volumetric stereo methods make use of the photo-consistency constraint to build a 3D map from which the target shape is extracted or segmented. The MVS problem was thus translated into a 3D segmentation problem. Shape from silhouettes [25] is a special case of voxel labeling in which the target shape (visual hull) is given by intersecting the projected volumes of the object's silhouettes on the images. The voxel coloring algorithm [23] computes a photo-consistent shape by projecting voxels and correlating pixel colors among visible set of images. The space carving algorithm [14] adopted a multi-pass sweeping method to carve out non-photo-consistent voxels.

A straightforward approach merges depth maps [7] by using [4], which computes signed distance function from each depth map. This merging approach relies on depth maps which can be improved using multiple hypotheses [3]. To handle occlusion in MVS, shiftable windows combined with temporal selection yields significant improvements near depth discontinuities [12]. Also, in [7], SSSD-style multi-baseline window matching were used to compute depth at high confidence points. Depth maps are also used in other top-performers such as graph-cuts: in [10], a robust, voting-based photo-consistency metric that does not need visibility reasoning [9, 27] was used to create depth maps for the subsequent graph-cuts minimization.

Graph-cuts [10, 27] is one successful technique in solving the MVS problem posed as one of 3D segmentation. Because photo-consistency basically uses pixel correspondence to triangulate points lying on the 3D object, the energy functional usually include two terms: the discontinuity cost, derived from photo-consistency measurements; and an additional labeling cost, which produces a "ballooning" effect to fill in the volume roughly bounded by voxels with high discontinuity cost. Graph-cuts MVS then focused on the proper design of the two costs. Another graph-theoretic approach [11] defines a "crust" using photo-consistency scores from which a manifold surface can be extracted via dual graph embedding.

Normal is a useful cue for robust surface reconstruction. In MVS, photo-flux [2] was introduced, but it required surface orientations information for foreground/background modeling. In [8], surface normals were considered within a photoconsistency measure. By assuming the scene geometry visible centered around a pixel to be locally planar, the depth, color scale, and normal can be related using an over-determined nonlinear system, which can be solved using iterative techniques.

3. Tensor Voting

A concise review of tensor voting [20] is given in [28]. In essence, *tensor* is used for token representation, and *voting* is used for non-iterative token-token communication. Tensor and voting are related by a *voting field* . A voting field is a dense tensor field for postulating smooth connection and discontinuity in a neighborhood. In this section we state two new results [28]: CFTV (closed-form tensor voting) and EMTV, which will be used in the following sections.

Closed-form solution to tensor voting. In tensor voting, voting fields are precomputed and stored as discrete voting fields for execution efficiency. Although precomputed once, discrete approximations involve uniform and dense sampling of tensor votes $\tilde{\mathbf{n}}\tilde{\mathbf{n}}^T$ where $\tilde{\mathbf{n}}$ is a normal vector. We proved a closed-form solution to tensor voting, which provides an efficient solution to computing an optimal tensor *without* resorting to discrete and dense sampling.

Given two sites $\mathbf{x}_i, \mathbf{x}_j \in \mathbb{R}^D$, and $\mathbf{K}_j$ which is a second order symmetric tensor represented by a $D \times D$ matrix ($D = 3$ in this paper), a tensor at site $\mathbf{x}_j$, the optimal tensor $\mathbf{S}_{ij}$ at $\mathbf{x}_i$ induced by $\mathbf{x}_j$ is given by[1]:

$$\mathbf{S}_{ij} = c_{ij}\mathbf{R}_{ij}\mathbf{K}_j\mathbf{R}'_{ij}, \tag{1}$$

[1]Initial $\mathbf{K}_i$ and $\mathbf{K}_j$ can be derived when the input direction is available (in the matching stage of TMVS), or simply assigned as an zero matrix (in the propagation stage of TMVS). This will be explained in the following sections.

stereo step	tv	purpose
matching, uncalibrated	`cftv`, `emtv`	F-matrix estimation
matching, calibrated	`cftv`	normal estimation
propagation	`cftv`	patch propagation
filtering	`cftv`, `mrftv`	outlier rejection

Table 1. The roles of `cftv`, `emtv`, and `mrftv` in the match-propagate-filter stereo pipeline.

where

$$c_{ij} = \exp(-\frac{||\mathbf{x}_i - \mathbf{x}_j||^2}{\sigma_d}), \tag{2}$$

σ_d is the scale of analysis which is the only free parameter, and

$$\mathbf{R}_{ij} = \mathbf{I} - 2\mathbf{r}_{ij}\mathbf{r}_{ij}^T, \quad \mathbf{R}'_{ij} = \mathbf{R}_{ij}(\mathbf{I} - \frac{1}{2}\mathbf{r}_{ij}\mathbf{r}_{ij}^T), \tag{3}$$

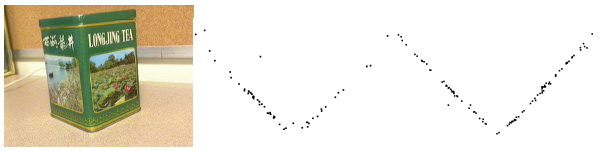
where $\mathbf{I}$ is an identity matrix and $\mathbf{r}_{ij}$ is an unit vector at $\mathbf{x}_j$ pointing to $\mathbf{x}_i$. Full derivation is given in [28].

If a point lies on a 3D surface, the stick votes received in its neighborhood reinforce each other with a high agreement of tensor orientations. The accumulated tensor should be stick-like, or $\lambda_1 \gg \lambda_2, \lambda_3$, where $\lambda_1, \lambda_2, \lambda_3$ are the eigenvalues of the eigensystem. This tensor indicates certainty in a single direction. On the other hand, an outlier receives a few inconsistent votes, so all the corresponding eigenvalues are small. We can thus define *surface saliencies* by $\lambda_1 - \lambda_2$, with the eigenvector $\hat{e}_1$ corresponding to λ_1 to denote the normal direction to the surface. Furthermore, if it is a discontinuity or a point junction where several surfaces intersect exactly at a single point, it indicates a high *dis*agreement of tensor votes where not a single direction is preferred. Junction saliency is indicated by high values of λ_3 (and thus all eigenvalues). Outlier noise is characterized by low vote saliency and low vote agreement. Therefore, by using surface saliency, our filtering can reject outliers while not smoothing out sharp features.

EMTV. While tensor voting can reject outliers well, it falls short of producing very accurate parameter estimation, explaining the use of RANSAC in the final parameter estimation step after outlier rejection [26]. We summarize below the EMTV algorithm for optimizing (1) the tensor $\mathbf{K}$ at each input site, and (2) the parameters of a single plane $\mathbf{v}$ of *any* dimensionality containing the inliers (e.g. epipolar geometry is a high-dimensional plane $\mathbf{v}$ estimation problem). The expectation-maximization algorithm (full derivation available in [28]) is suitable for such alternating optimization as (1) and (2) are interdependent:

E-Step: Let w_i to be the probability of an observation o_i being an inlier. Then

$$w_i = \frac{1}{2\pi\sigma\sigma_1}\exp(-\frac{||\mathbf{x}_i^T\mathbf{v}||^2}{2\sigma^2})\exp(-\frac{||\mathbf{v}^T\mathbf{K}_i^{-1}\mathbf{v}||}{2\sigma_1^2}) \tag{4}$$



(a) (b) (c)

Figure 2. *LongJing*: (a) One of the two input images. (b) Sparse reconstruction generated by using `KeyMatchFull`. (c) Sparse reconstruction generated by using `emtv_match`.

M-Step:

$$\begin{aligned}
\mathbf{K}_i^{-1} &= \frac{1}{\sum_{j\in\mathcal{G}(i)} w_j}(\sum_{j\in\mathcal{G}(i)} \mathbf{S}_{ij}^{-1}w_j - \frac{\sigma_2^2}{2\sigma_1^2}\mathbf{v}\mathbf{v}^T w_i) \\
\mathbf{M}\mathbf{v} &= 0 \quad \text{(solve for } \mathbf{v}) \\
\sigma^2 &= \frac{\sum_i ||\mathbf{x}_i^T\mathbf{v}||^2 w_i}{\sum_i w_i} \\
\sigma_1^2 &= \frac{\sum_i ||\mathbf{v}^T\mathbf{K}_i^{-1}\mathbf{v}||w_i}{\sum_i w_i} \\
\sigma_2^2 &= \frac{\sum_i \sum_{j\in\mathcal{G}(i)} ||\mathbf{K}_i^{-1} - \mathbf{S}'_{ij}||_F^2 w_i w_j}{\sum_i w_i}
\end{aligned} \tag{5}$$

where $\mathbf{M} = \sum_i \mathbf{x}_i\mathbf{x}_i^T w_i + \frac{\sigma^2}{\sigma_1^2}\sum_i \mathbf{K}_i^{-1}w_i$ and $\mathcal{G}(i)$ is a set of neighbors of i.

To conclude this section, Table 1 summarizes the roles of tensor voting in the match-propagate-filter stereo pipeline. We explained closed-form tensor voting (`cftv`) and EMTV (`emtv`) above. Markov Random Field tensor voting (`mrftv`) will be described shortly.

4. Matching

In the uncalibrated scenario, EMTV estimates parameter accurately by employing closed-form tensor voting, and effectively discards epipolar geometries induced by wrong matches (section 4.1). In the calibrated scenario, TMVS produces better 3D normals than PMVS by utilizing tensors and their communication via closed-form tensor voting (section 4.2).

4.1. Uncalibrated Images

When the input images are uncalibrated, camera calibration is performed using nonlinear least-squares minimization and bundle adjustment [17] which requires good matches as input.

We provide our method, `emtv_match`, to show the efficacy of EMTV on camera calibration while noting others can be used. EMTV estimates the fundamental matrix (F-matrix) by hyperplane fitting [28]. Here, SIFT [18] is used to detect image keypoints. Candidate matches are generated by comparing the resulting 128D feature vectors, so many

(a) (b) (c)

Figure 3. *Teapot*: (a) Sparse reconstruction (360 points) generated by using `KeyMatchFull`. (b) Sparse reconstruction (37 points) generated by using `ransac_match`. (c) Sparse reconstruction (2152 points) generated by using `emtv_match`. The candidate matches returned by SIFT are extremely noisy due to the ambiguous patchy patterns. On average 17404 trials were run in `ransac_match`. It is very time consuming to run more trials on this noisy and large input where an image pair can have as many as 5000 similar matches. `emtv_match` does not need random sampling.

matched keypoints are not corresponding. The epipolar constraint is enforced in the matching process using EMTV, which returns the fundamental matrix *and* the probability w_i (Eqn (4)) of a keypoint pair i being an inlier. In the following experiments, we assume keypoint pair i is an inlier if $w_i > 0.8$. Note that no random sampling is used.

The following compares `emtv_match` with `KeyMatchFull` [24] and `ransac_match`. `ransac_match` solves $\mathbf{v}$ (hyperplane fitting) by using RANSAC.

Tea Can. Figure 2 shows that, by using our filtered matches, even in the absence of any focal length input, our sparse reconstruction of the tea can (the image pair was obtained from [29]), produced by the nonlinear least-squares minimization and bundle adjustment [17], is denser and contains less errors as compared with [24], where we can faithfully reconstruct the right-angled container.

Teapot. Figure 3 shows our running example *teapot* which contains repetitive patterns across the whole object. Wrong matches can be easily produced by similar patterns on different parts of the teapot. This data set contains 30 images captured using a Nikon D70 camera. Automatic configuration was set during the image capture.

Visually, the result produced using `emtv_match` is much denser than the one produced with `KeyMatchFull` and `ransac_match`, the latter of which solves the hyperplane fitting by using RANSAC. While `KeyMatchFull` can still handle this data set, we observe that many outliers *and* inliers were rejected as well. This is because `KeyMatchFull` employed a restrictive criterion to drastically reduce the number of outliers. Specifically, they used $d_1 < 0.6d_2$, where d_1 and d_2 are respectively the shortest and second shortest distance between a point and a candidate match in the 128D feature space. In other words, many similar structures or repeated patterns were filtered out, and only very distinctive feature pairs were retained for the following bundle adjustment stage. On the other hand, `emtv_match` utilizes the epipolar geometry constraint by computing the fundamental matrix in a data driven manner. Note the result obtained using `ransac_match` is extremely sparse, which can be attributed to two reasons: (1) the fundamental matrix is rank 2 which implies that $\mathbf{v}$ spans a subspace $\leq$ 8-D rather than a 9-D hyperplane; (2) the input matches contain too many outliers.

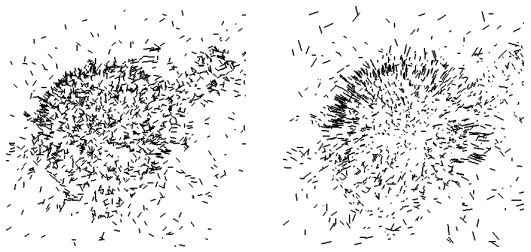


Figure 4. Comparing the initial patches generated by PMVS (left) and TMVS (right). Normals shown on the right are attenuated by surface saliency, so potential outliers are detected early in the stereo pipeline.

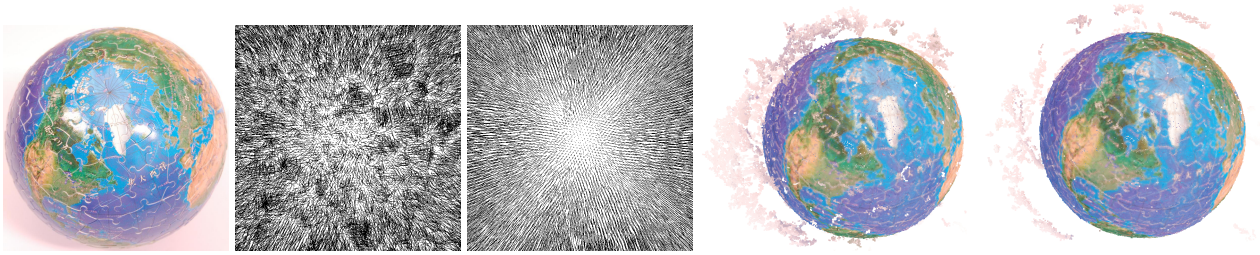
4.2. Calibrated Images

If the input images are calibrated, we proceed to produce initial matches as in PMVS. In TMVS, we encode each 3D patch into a 3D tensor $\mathbf{K}$. To initialize $\mathbf{K}$ the initial normals $\tilde{\mathbf{n}}$ given by PMVS can be used, that is, $\mathbf{K} = \tilde{\mathbf{n}}\tilde{\mathbf{n}}^T$. Or we can simply initialize $\mathbf{K}$ as an identity to indicate that we have no orientation preference. We found the optimal tensors produced by closed-form tensor voting, that is, Eqn. (1), in both cases are quite similar. The tensor votes collected are summed up using tensor addition which simply adds up the collected matrices computed using Eqn. (1).

Figure 4(a) shows the initial 3D patches and the perturbed normals estimated by PMVS. Figure 4(b) shows the improved set of normals produced by TMVS where the $(\lambda_1 - \lambda_2)\hat{e}_1$ components of all tensors are shown. A more accurate set of 3D normals will improve the propagation process by better predicting which 3D direction to explore next when combined with photoconsistency and geometric consistency enforcement.

5. Propagation

In PMVS, patch expansion proceeds from initial patches in the 3D space to process unmatched image regions in the previous step. Using patch normals and their neighborhood, the algorithm generated a set of candidate 3D positions, each passing through the respective lines of sight. Then, at each candidate position the patch normal and location along the line of sight are optimized using photoconsistency (local pixel colors) and visibility (the subset of visible images). Overall, the patch expansion is a 3D floodfill algorithm by considering normal directions, photoconsistency, and visibility. This expansion algorithm does not however take active consideration geometric cues available in the ori-



(a) (b) (c) (d) (e)

Figure 5. *Earth*. (a) one input image (81 in total), (b) and (c) show zoom-in views of the normal reconstruction produced by PMVS and TMVS after the propagation step, (d) and (e) show respectively one view of the quasi-dense reconstruction by PMVS and TMVS.

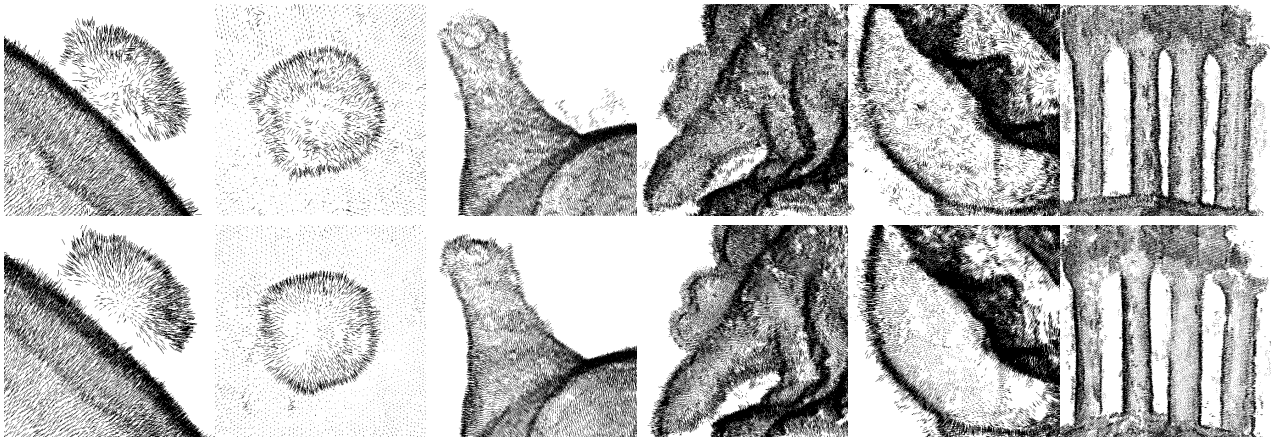


Figure 6. Comparing the optimized normals generated by PMVS (top) and TMVS (bottom) in the propagation step. The left three are different zoom-in views of *teapot*, the rest are zoom-in views of the normal results of the Middlebury dataset: *dinoRing* and *templeRing*. No silhouette or visual hull information is used in propagation.

	min	avg	max
TMVS	0.0032	1.1236	5.7442
PMVS	0.01	11.7417	89.9998

Table 2. Quantitative comparison on normal estimation accuracy. The angular errors shown are in degrees.

ented set of 3D points, thus resulting in perturbed normals (as well as positions). Erroneous patch normals will in turn adversely affect the propagation accuracy, producing more outliers as demonstrated in the following quantitative study.

We first performed quantitative comparison between PMVS and TMVS on normals accuracy. Figure 5 shows an example *Earth* where the analytical geometry is known (the ground-truth is a sphere). This data set contains 81 images captured by a Nikon D40 camera with fixed intrinsic camera parameters. One input image is shown in Figure 5(a). To show the significance of our improvement, we executed both PMVS and TMVS on this example and compared quantitatively the estimated surface normals. Table 2 tabulates the angular errors (in degree) produced by the respective methods[2]. It shows that TMVS is a clear winner where the maximum error produced is much smaller than the average error produced by PMVS. Qualitatively, we observe the difference from Figure 5(b) and (c), which are the zoom-in views of the surface normals generated by the tested methods. The surface normals produced by TMVS radiates from the center while those by PMVS oriented quite randomly. Moreover, because TMVS utilizes geometric cues via tensor communication, TMVS generated less outliers compared to PMVS in the propagation stage, as shown in Figure 5(d) and (e).

After showing the improvement, here we discuss the reason why, at this stage, TMVS outperforms PMVS. In TMVS, we incorporate closed-form tensor voting into the patch propagation step, which imposes the uniqueness constraint along the line of sight. Specifically, given a candidate position p_0, we sample along its line of sight a set of normalized cross correlations and surface saliencies. Normalized cross correlation can be computed using camera and image information (i.e., visibility and photoconsistency) which is similar to [5]. Surface saliencies are obtained by sampling tensor votes along the line of sight using closed-

[2]Outliers are ignored in the calculation. A point is regarded as an outlier if $d_p > r_{gt} + 0.06$, where d_p is the distance of the point measured from the center of the *Earth* and $r_{gt} = 0.6533$ is the radius of the *Earth*.

form tensor voting. If the site being sampled is not an input site, which is usually the case, then the initial $\mathbf{K}$ is set to be a zero matrix. Else, initial $\mathbf{K}$ is simply the tensor obtained in the previous matching step.

Let ncc_p and sal_p be the normalized cross correlation and the surface saliency (normalized to $[0, 1]$ using the maximum eigenvalue of $\mathbf{K}$ at p) respectively at a position p along the line of sight. We detect the maximum of

$$(1 - \rho)\text{ncc}_p + w\text{sal}_p \tag{6}$$

along the line of sight passing through a given candidate position p_0, and $\rho \in [0, 1]$ is a weight factor which is set to (0.2–0.4) in our experiments. Note that [15, 21] also considers surface saliency maxima along lines of sight. They select matches from the point cloud of candidate matches generated by the initial matching stage by examining the amount of support received from their neighboring candidate matches after tensor voting. However, we have two differences: (1) closed-form tensor voting contributes a faster and accurate implementation without discrete approximations using pre-computed tensor voting fields; (2) both photoconsistency and geometric consistencies are considered in our optimization process. In TMVS, the precise patch locations are optimized using photoconsistency and geometric consistency prescribed by tensor voting.

Figure 6 compares the propagation results of PMVS and TMVS on *teapot*, *dinoRing* and *templeRing*. For the *teapot*, our normals are smoother while important features such as the spout and the lid are preserved. For *dinoRing* we have less noise over the fins compared with PMVS. Note our better normals on the base which supports the *dinoRing* (pointing upward rather than oriented randomly as shown in the PMVS result). Our *templeRing* result has less outliers. Note we did not use any object masks when the above data were processed.

6. Filtering

In PMVS, visibility consistency is applied to reject outliers. Tricky outliers are close to the target shape and may accidentally form a structure by themselves, which are much less salient compared with the quasi-dense reconstruction.

In TMVS, because of tensor voting, less outliers are generated during the propagation. For outliers that escape from the propagation process, they can be removed by running MRF-TV, tensor voting on MRF, by tensor communications over the entire geometry. Figure 7 shows the result before and after applying MRF-TV.

Recall that $\mathbf{K}_j$ denotes the tensor residing at $\mathbf{x}_j$. To obtain the estimated tensor at $\mathbf{x}_i$ induced by $\mathbf{x}_j$, we employ Eqn (1) to estimate $\mathbf{S}_{ij}$. In MRF, a Markov network is a graph consists of two types of node – a set of hidden variables $\mathbf{E}$ and a set of observed variables $\mathbf{O}$, where the edges

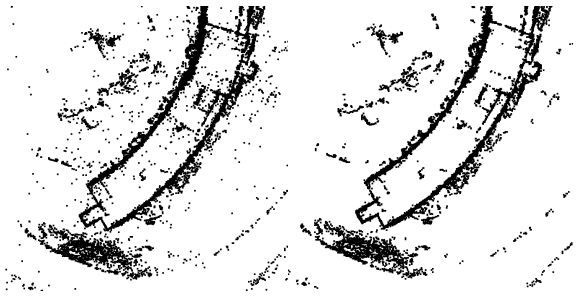


Figure 7. Results before and after filtering of *Hall 3* (images shown in Figure 10). Top view of the reconstructed building is shown here. All salient 3D structures are retained in the filtered result, including the bushes near the left facade and planters near the right facade in this top view of the building.

of the graph are described by the following posterior probability $\text{P}(\mathbf{E}|\mathbf{O})$ with standard Bayesian framework:

$$\text{P}(\mathbf{E}|\mathbf{O}) \propto \text{P}(\mathbf{O}|\mathbf{E})\text{P}(\mathbf{E}) \tag{7}$$

By letting $\mathbf{E} = \{\mathbf{K}_i | i = 1, 2, \cdots, N\}$ and $\mathbf{O} = \{\tilde{\mathbf{K}}_i | i = 1, 2, \cdots, N\}$, where N is total number of points and $\tilde{\mathbf{K}}_i$ is the known tensor at $\mathbf{x}_i$, and suppose that inliers follow Gaussian distribution, we obtain the the likelihood $\text{P}(\mathbf{O}|\mathbf{E})$ and the prior $\text{P}(\mathbf{E})$ as the following:

$$\text{P}(\mathbf{O}|\mathbf{E}) = \prod_i \text{p}(\tilde{\mathbf{K}}_i|\mathbf{K}_i) = \prod_i e^{-\frac{||\mathbf{K}_i - \tilde{\mathbf{K}}_i||_F^2}{\sigma_h}} \tag{8}$$

$$\text{P}(\mathbf{E}) = \prod_i \prod_{j \in \mathcal{N}(i)} \text{p}(\mathbf{S}_{ij}|\mathbf{K}_i) \tag{9}$$

$$= \prod_i \prod_{j \in \mathcal{N}(i)} e^{-\frac{||\mathbf{K}_i - \mathbf{S}_{ij}||_F^2}{\sigma_s}} \tag{10}$$

where $||\cdot||_F$ is Frobenius norm, $\tilde{\mathbf{K}}_i$ is the known tensor at $\mathbf{x}_i$, $\mathcal{N}(i)$ is the set of neighbor corresponds to $\mathbf{x}_i$ and σ_h and σ_s are two constants respectively. By taking the logarithm of Eqn (7), we obtain the following energy function:

$$E(\mathbf{E}) = \sum_i ||\mathbf{K}_i - \tilde{\mathbf{K}}_i||_F^2 + g \sum_i \sum_{j \in \mathcal{N}(i)} ||\mathbf{K}_i - \mathbf{S}_{ij}||_F^2 \tag{11}$$

where $g = \frac{\sigma_h}{\sigma_s}$. Theoretically, this quadratic energy function can be directly solved once and for all by Singular Value Decomposition (SVD). Since N can be large thus making direct SVD impractical, we adopt an iterative approach: by taking the partial derivative of Eqn (11) (w.r.t. to $\mathbf{K}_i$) the following update rule is obtained:

$$\mathbf{K}_i^* = (\tilde{\mathbf{K}}_i + 2g \sum_{j \in \mathcal{N}(i)} \mathbf{S}_{ij})(\mathbf{I} + g \sum_{j \in \mathcal{N}(i)} (\mathbf{I} + c_{ij}^2 {\mathbf{R}'_{ij}}^2))^{-1} \tag{12}$$

which is a Gauss-Seidel solution. When successive over-relaxation (SOR) is employed, the update rule becomes:

$$\mathbf{K}_i^{(m+1)} = (1 - q)\mathbf{K}_i^{(m)} + q\mathbf{K}_i^* \tag{13}$$

Figure 8. *Tripp* reconstruction from sparse data set: three input images (left) and the quasi-dense 3D reconstruction produced by PMVS (middle) and TMVS (right).

where $1 < q < 2$ is the SOR weight and m is the iteration number.

When the energy function (Eqn (11)) is minimized, we can obtain the surface saliency for each $\mathbf{x}_i$ by applying eigen-decomposition on the corresponding estimated $\mathbf{K}_i$. We consider $\mathbf{x}_i$ is an outlier if the respective surface saliency (i.e. $\lambda_1 - \lambda_2$) is smaller than t (we set $t = 0.1$ for all experiments).

7. More Results

Tripp and *George*. We performed stress test on TMVS using sparse and unevenly-spaced cameras. 25 images of *Tripp* and 14 images of *George* were obtained. All images were casually captured using an off-the-shelf digital camera. We compare the quasi-dense reconstruction results of *Tripp* produced by PMVS and TMVS, as shown in Figure 8. Because TMVS produced more accurate normals, it can fill more holes during the propagation step. Figure 9 shows a few images and several views of the quasi-dense reconstruction produced by TMVS.

Hall3. Finally, we captured photos all around a building using an off-the-shelf digital camera. All images were taken on the ground level not higher than the building, so we have very few samples of the rooftop. The building facades are curved and the windows on the building look identical to each other. The patterns on the front and back facade look nearly identical. These ambiguities cause significant challenges in the matching stage especially for wide-base stereo. The input photos (179 images in total) were first calibrated as described, followed by running TMVS to obtain the quasi-dense reconstruction as shown in Figure 10. The 3D reconstruction is faithful to the real building.

8. Discussion

The only free parameter in tensor voting is the scale of visual analysis σ_d in Eqn (2) which can be estimated by analyzing local tensor densities. In our experiments, all tensors are sorted using the ANN tree [1] which allows efficient access of each tensor's neighbors. Let d be the average distance to each tensor's closest neighbor. Then, σ_d is given by $a\sqrt{(-d^2/\log(\epsilon)}$ where $\epsilon = 0.075$ is the minimum strength of the tail of the Gaussian, and a is a positive constant. We found that a wide range of σ_d operates well, while an excessively large σ_d will produce wrong and over-smoothed normals. It is not difficult for user to obtain a good σ_d by tuning a: run our system (the efficient closed-form tensor voting) a few times on the initial *sparse* tensors only, and visualize the initial results such as Figure 4.

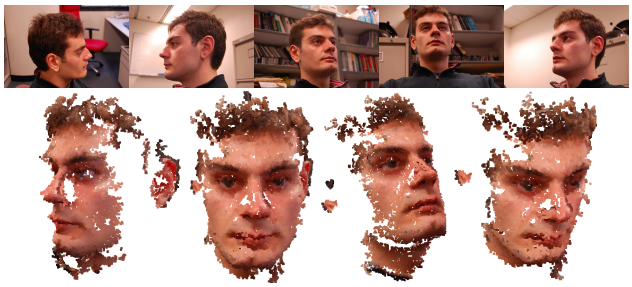


Figure 9. *George* reconstruction from sparse data set: five input images (top) and four views of the quasi-dense 3D reconstruction (bottom).

Our experiments were run on multicore Linux machines in a multiuser environment. Similar to [5], the bottleneck of TMVS is tensor propagation. Depending on the input size, our processing time ranges from 10 minutes to a few hours. For the *Earth*, it has about 6000 initial tensors and each communicates with around 100 neighbors. The running time is about 1 hour on a quadcore machine with 4 x AMD Opteron 844 (1.8GHz) CPU with 8GB RAM.

9. Concluding Remarks

We described TMVS which is founded on our new closed-form solution to tensor voting [28], and provides a unified approach to implement the match-propagate-filter stereopsis pipeline with theoretical guarantees: CFTV is a closed form solution, EMTV has been shown to be convergent (where EM's convergence is well known [19]), and we provided an efficient solution to MRFTV in this paper. The implementation strategy is straightforward because it is not difficult to implement Eqns (1), (4), (5), and (12)–(13).

Using PMVS's match-propagate-filter pipeline as our implementation backbone, TMVS has performance similar to PMVS when TMVS is tested on the Middlebury MVS dataset. For surface normals, our qualitative and quantitative evaluation show that TMVS produced significantly improved normals. As 3D patch (oriented points) is the main processing token in PMVS (analogously 3D tensors in TMVS), this improvement leads to less accumulation errors and outliers in the propagation results.

As a side benefit, TMVS has led us to develop the following utilities: CFTV (for perceptual grouping), EMTV (for parameter estimation), and MRFTV (for outlier rejec-

Figure 10. The *Hall 3* reconstruction: ten input images (top) and five views of the quasi-dense 3D reconstruction (bottom).

tion). They are included in the supplemental material and available to the community, and we believe they are useful in many MVS and other vision systems as well.

References

[1] S. Arya and D. M. Mount. Approximate nearest neighbor searching. *ACM-SIAM SODA'93*, pages 271–280.

[2] Y. Boykov and V. Lempitsky. From photohulls to photoflux optimization. In *BMVC06*, page III:1149, 2006.

[3] N. Campbell, G. Vogiatzis, C. Hernandez, and R. Cipolla. Using multiple hypotheses to improve depth-maps for multi-view stereo. In *ECCV08*, pages I: 766–779, 2008.

[4] B. Curless and M. Levoy. A volumetric method for building complex models from range images. In *SIGGRAPH96*, pages 303–312, 1996.

[5] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. In *CVPR07*, pages 1–8, 2007.

[6] Y. Furukawa and J. Ponce. Accurate, dense, and robust multi-view stereopsis. In *PAMI09*, 2009.

[7] M. Goesele, B. Curless, and S. Seitz. Multi-view stereo revisited. In *CVPR06*, pages II: 2402–2409, 2006.

[8] M. Goesele, N. Snavely, B. Curless, H. Hoppe, and S. Seitz. Multi-view stereo for community photo collections. In *ICCV07*, pages 1–8, 2007.

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

[10] C. Hernindez Esteban, G. Vogiatzis, and R. Cipolla. Probabilistic visibility for multi-view stereo. In *CVPR07*, pages 1–8, 2007.

[11] A. Hornung and L. Kobbelt. Hierarchical volumetric multi-view stereo reconstruction of manifold surfaces based on dual graph embedding. In *CVPR06*, pages I: 503–510, 2006.

[12] S. Kang, R. Szeliski, and J. Chai. Handling occlusions in dense multi-view stereo. In *CVPR01*, pages I:103–110, 2001.

[13] M. Kazhdan, M. Bolitho, and H. Hoppe. Poisson surface reconstruction. In *Symposium on Geometry Processing*, pages 69–78, 7 2006.

[14] K. Kutulakos and S. Seitz. A theory of shape by space carving. *IJCV*, 38(3):199–218, July 2000.

[15] M. Lee, G. Medioni, and P. Mordohai. Inference of segmented overlapping surfaces from binocular stereo. *PAMI*, 24(6):824–837, June 2002.

[16] M. Lhuillier and L. Quan. A quasi-dense approach to surface reconstruction from uncalibrated images. *PAMI*, 27(3):418–433, March 2005.

[17] M. Lourakis and A. Argyros. The design and implementation of a generic sparse bundle adjustment software package based on the levenberg-marquardt algorithm. In *Technical Report 340*, 2004.

[18] D. Lowe. Distinctive image features from scale-invariant keypoints. *IJCV*, 60(2):91–110, November 2004.

[19] G. J. McLachlan and T. Krishnan. *The EM Algorithm and Extensions*. Wiley, 2008.

[20] G. Medioni, M. S. Lee, and C. K. Tang. *A Computational Framework for Segmentation and Grouping*. Elsevier, 2000.

[21] P. Mordohai and G. Medioni. Stereo using monocular cues within the tensor voting framework. *PAMI*, 28(6):968–982, June 2006.

[22] S. Seitz, B. Curless, J. Diebel, D. Scharstein, and R. Szeliski. A comparison and evaluation of multi-view stereo reconstruction algorithms. In *CVPR06*, pages I: 519–528, 2006.

[23] S. Seitz and C. Dyer. Photorealistic scene reconstruction by voxel coloring. *IJCV*, 35(2):151–173, November 1999.

[24] N. Snavely, S. Seitz, and R. Szeliski. Modeling the world from internet photo collections. *IJCV*, 80(2), November 2008.

[25] R. Szeliski. Rapid octree construction from image sequences. *CVGIP*, 58(1):23–32, July 1993.

[26] W. Tong, C. Tang, and G. Medioni. Simultaneous two-view epipolar geometry estimation and motion segmentation by 4d tensor voting. *PAMI*, 26(9):1167–1184, September 2004.

[27] G. Vogiatzis, C. Hernindez Esteban, P. Torr, and R. Cipolla. Multiview stereo via volumetric graph-cuts and occlusion robust photo-consistency. *PAMI*, 29(12):2241–2246, December 2007.

[28] T.-P. Wu, J. Jia, and C.-K. Tang. A closed-form solution to tensor voting for robust parameter estimation via expectation-maximization. Technical report, HKUST, 2009. `http://www.cse.ust.hk/~pang/papers/emtvTechnote.pdf`.

[29] Z. Zhang. A flexible new technique for camera calibration. *PAMI*, 22(11):1330–1334, November 2000.