A Closed-Form Solution to Tensor Voting: Theory and Applications

Tai-Pang Wu, *Member*, *IEEE Computer Society*, Sai-Kit Yeung, *Member*, *IEEE Computer Society*, Jiaya Jia, *Senior Member*, *IEEE*, Chi-Keung Tang, *Senior Member*, *IEEE*, and Gérard Medioni, *Fellow*, *IEEE*

Abstract—We prove a closed-form solution to tensor voting (**CFTV**): Given a point set in any dimensions, our closed-form solution provides an exact, continuous, and efficient algorithm for computing a structure-aware tensor that simultaneously achieves salient structure detection and outlier attenuation. Using CFTV, we prove the convergence of tensor voting on a Markov random field (MRF), thus termed as **MRFTV**, where the structure-aware tensor at each input site reaches a stationary state upon convergence in structure propagation. We then embed structure-aware tensor into expectation maximization (EM) for optimizing a single linear structure to achieve efficient and robust parameter estimation. Specifically, our **EMTV** algorithm optimizes both the tensor and fitting parameters and does not require random sampling consensus typically used in existing robust statistical techniques. We performed quantitative evaluation on its accuracy and robustness, showing that EMTV performs better than the original TV and other state-of-the-art techniques in fundamental matrix estimation for multiview stereo matching. The extensions of CFTV and EMTV for extracting multiple and nonlinear structures are underway.

Index Terms—Tensor voting, closed-form solution, structure inference, parameter estimation, multiview stereo.

1 Introduction

This paper reinvents tensor voting (TV) [19] for robust computer vision by proving a closed-form solution to computing an *exact structure-aware tensor* after *data communication* in a feature space of any dimensions, where the goal is salient structure inference from noisy and corrupted data.

To infer structures from noisy data corrupted by outliers, in tensor voting, input points communicate among themselves subject to proximity and continuity constraints. Consequently, each point is aware of its structure saliency via a structure-aware tensor. Structure refers to surfaces, curves, or junctions if the feature space is three dimensional where a structure-aware tensor can be visualized as an ellipsoid: If a point belongs to a smooth surface, the resulting ellipsoid after data communication resembles a stick pointing along the surface normal; if a point lies on a curve the tensor resembles a plate where the curve tangent is perpendicular to the plate tensor; if it is a point junction where surfaces intersect, the tensor will be like a ball. An outlier is characterized by a set of inconsistent votes it receives after data communication.

We develop in this paper a closed-form solution to tensor voting (CFTV), which is applicable to the special as well as general theory of tensor voting. This paper focuses on the *special* theory, where the above data communication is data driven without using constraints other than proximity and continuity. The special theory, sometimes coined "first voting pass," is applied to process raw input data to detect structures and outliers. In addition to structure detection and outlier attenuation, in the *general* theory of tensor voting tensor votes are propagated along preferred directions to achieve data communication when such directions are available, typically after the first pass, such that useful tensor votes are reinforced, whereas irrelevant ones are suppressed.

Expressing tensor voting in a single and compact equation, or a closed-form solution, offers many advantages: Not only can an exact and efficient solution be achieved with less implementation effort for salient structure detection and outlier attenuation, formal and useful mathematical operations such as differential calculus can be applied, which is otherwise impossible using the original tensor voting procedure. Notably, we can prove the convergence of tensor voting on Markov random fields (MRFTV), where a structure-aware tensor at each input site achieves a stationary state upon convergence.

Using CFTV, we contribute a mathematical derivation based on expectation maximization (EM) that applies the exact tensor solution for extracting the most salient linear structure, despite the fact that the input data are highly corrupted. Our algorithm is called EMTV, which optimizes

- *T.-P. Wu is with the Enterprise and Consumer Electronics (ECE) Group, Hong Kong Applied Science and Technology Research Institute (ASTRI) Co. Ltd., 3/F, Bio-informatics Centre, 2 Science Park West Avenue, Hong Kong Science Park, Shatin, Hong Kong. E-mail: tpwu@astri.org.*
- *S.-K. Yeung is with the Pillar of Information Systems Technology and Design, The Singapore University of Technology and Design, 20 Dover Drive, Singapore 128805. E-mail: saikit@sutd.edu.sg.*
- *J. Jia is with the Department of Computer Science and Engineering, The Chinese University of Hong Kong, Shatin, Hong Kong. E-mail: leojia@cse.cuhk.edu.hk.*
- *C.-K. Tang is with the Department of Computer Science and Engineering, The Hong Kong University of Science and Technology, Clear Water Bay, Hong Kong. E-mail: cktang@cse.ust.hk.*
- *G. Medioni is with the Institute for Robotics and Intelligent Systems, University of Southern California, PHE 204, MC-2073, Los Angeles, CA 90089-0273. E-mail: medioni@iris.usc.edu.*

Manuscript received 13 July 2010; revised 31 Jan. 2011; accepted 6 Nov. 2011; published online 13 Dec. 2011.
Recommended for acceptance by C. Stewart.
For information on obtaining reprints of this article, please send e-mail to: tpami@computer.org, and reference IEEECS Log Number TPAMI-2010-07-0529.
Digital Object Identifier no. 10.1109/TPAMI.2011.250.

Published by the IEEE Computer Society

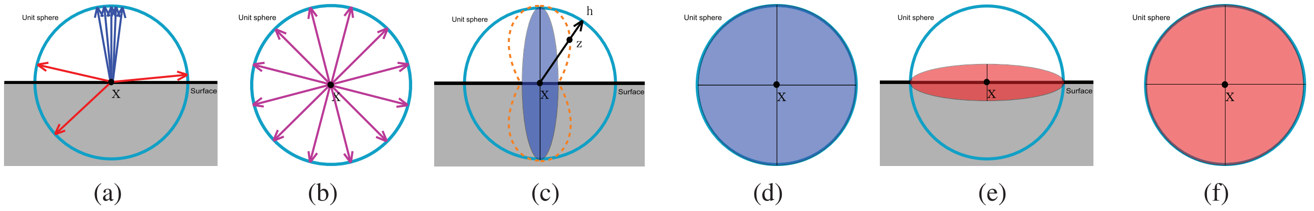


Fig. 1. Inlier/outlier and tensor inverse illustration. (a) The normal votes received at a surface point cast by points in $\mathbf{x}$'s neighborhood. Three salient outliers are present. (b) For a nonsurface point, there is no preference to any normals. (c) The structure-aware tensor induced by the normal observations in (a), which is represented by a d-D ellipsoid, where $d \geq 2$. The orange curve (dashed curve) represents the variance produced along all possible directions. (d) The structure-aware tensor collected after collecting the received votes in (b). (e) and (f) correspond to the inverse of (c) and (d), respectively.

both the tensor and fitting parameters upon convergence and does not require random sampling consensus (RANSAC) typical of existing robust statistical techniques. The extension to extract salient multiple and nonlinear structures is underway.

While the mathematical derivation may seem involved, our main results for CFTV, MRFTV, and EMTV, that is, (11), (12), (18), (19), (26), and (30), have rigorous mathematical foundations, are applicable to any dimensions, produce more robust and accurate results, as demonstrated in our qualitative and quantitative evaluation using challenging synthetic and real data, but, on the other hand, are easier to implement. The source codes accompanying this paper are available in the supplemental material, which can be found iat http://www.cs.ust.hi/!cktang/tv_code.zip.

2 Related Work

While this paper is mainly concerned with tensor voting, we provide a concise review on robust estimation and expectation maximization.

Robust estimators. Robust techniques are widely used and an excellent review of the theoretical foundations of robust methods in the context of computer vision can be found in [20].

The Hough transform [13] is a robust voting-based technique operating in a parameter space capable of extracting multiple models from noisy data. Statistical Hough transform [6] can be used for high-dimensional spaces with sparse observations. Mean shift [5] has been widely used since its introduction to computer vision for robust feature space analysis. The Adaptive Mean Shift [11] with variable bandwidth in high dimensions was introduced in texture classification and has since been applied to other vision tasks. Another popular robust method in computer vision is in the class of random sampling consensus procedures [7], which have spawned a lot of follow-up work (e.g., optimal randomized RANSAC [4]).

Like RANSAC [7], robust estimators, including the LMedS [22] and the M-estimator [14], adopted a statistical approach. The LMedS, RANSAC, and the Hough transform can be expressed as M-estimators with auxiliary scale [20]. The choice of scales and parameters related to the noise level are major issues. Existing works on robust scale estimation use random sampling [26] or operate on different assumptions (e.g., more than 50 percent of the data should be inliers [23]; inliers have a Gaussian distribution [15]). Among them, the Adaptive Scale Sample Consensus (ASSC) estimator [28] has shown the best performance where the estimation process requires no free parameter as input. Rather than using a Gaussian distribution to model inliers, the authors of [28] proposed to use a two-step scale estimator (TSSE) to refine the model scale: First, a non-Gaussian distribution is used to model inliers where local peaks of density are found by mean shift [5]; second, the scale parameter is estimated by a median scale estimator with the estimated peaks and valleys. On the other hand, the projection-based M-estimator (pbM) [3], an improvement made on the M-estimator, uses a Parzen window for scale estimation, so the scale parameter is automatically found by searching for the normal direction (projection direction) that maximizes the sharpest peak of the density. This does not require an input scale from the user. While these recent methods can tolerate more outliers, most of them still rely on or are based on RANSAC and a number of random sampling trials is required to achieve the desired robustness.

To reject outliers, a multipass method using L_∞-norms was proposed to successively detect outliers which are characterized by maximum errors [24].

Expectation maximization. EM has been used in handling missing data and identifying outliers in robust computer vision [8], and its convergence properties were studied [18]. In essence, EM consists of two steps [2], [18]:

1. **E-Step**. Computing an expected value for the complete data set using incomplete data and the current estimates of the parameters.
2. **M-Step**. Maximizing the complete data log likelihood using the expected value computed in the E-step.

EM is a powerful inference algorithm, but it is also well known from [8] that: 1) Initialization is an issue because EM can get stuck in poor local minima, and 2) treatment of data points with small expected weights requires great care. They should not be regarded as negligible, as their aggregate effect can be quite significant. In this paper, we initialize EMTV using structure-aware tensors obtained by CFTV. As we will demonstrate, such initialization not only allows the EMTV algorithm to converge quickly (typically within 20 iterations) but also produces accurate and robust solution in parameter estimation and outlier rejection.

3 Data Communication

In the tensor voting framework, a data point, or voter, communicates with another data point, or vote receiver, subject to proximity and continuity constraints, resulting in a tensor vote cast from the voter to the vote receiver (Fig. 1). In the following, we use $\mathbf{n}$ to denote a unit voting stick tensor, $\mathbf{v}$ to denote a stick tensor vote received. Stick tensor

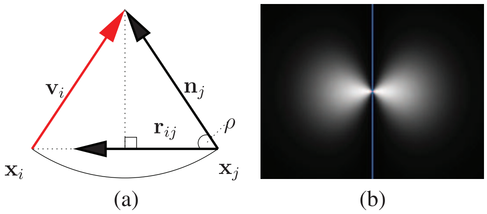


Fig. 2. (a) The normal vote $\mathbf{v}_i$ received at $\mathbf{x}_i$ using an arc of the osculating circle between $\mathbf{x}_i$ and $\mathbf{x}_j$, assuming the normal voter at $\mathbf{x}_j$ is $\mathbf{n}_j$, where $\mathbf{n}_j$, $\mathbf{r}_{ij}$, and $\mathbf{v}_i$ are unit vectors in this illustration. (b) Plot of (1) in 2D.

vote may not be unit vectors when they are multiplied by vote strength. These stick tensors are building elements of a structure-aware tensor vote.

Here, we first define a decay function η to encode the proximity and smoothness constraints ((1) and (2)). While similar in effect to the decay function used in the original tensor voting and also to the one used in [9], where a vote attenuation function is defined to decouple proximity and curvature terms, our modified function, which also differs from that in [21], enables a closed-form solution for tensor voting without resorting to precomputed discrete voting fields.

Refer to Fig. 2. Consider two points $\mathbf{x}_i \in \mathbb{R}^d$ and $\mathbf{x}_j \in \mathbb{R}^d$ (where $d > 1$ is the dimension) that are connected by some smooth structure in the feature/solution space. Suppose that the unit normal $\mathbf{n}_j$ at $\mathbf{x}_j$ is known. We want to generate at $\mathbf{x}_i$ a normal (vote) $\mathbf{v}_i$ so that we can calculate $\mathbf{K}_i \in \mathbb{R}^d \times \mathbb{R}^d$, where $\mathbf{K}_i$ is the structure-aware tensor at $\mathbf{x}_i$ in the presence of $\mathbf{n}_j$ at $\mathbf{x}_j$. In tensor voting, a structure-aware tensor is a second-order symmetric tensor, which can be visualized as an ellipsoid.

While many possibilities exist, the unit direction $\mathbf{v}_i$ can be derived by fitting an arc of the osculating circle between the two points. Such an arc keeps the curvature constant along the hypothesized connection, thus encoding the smoothness constraint. $\mathbf{K}_i$ is then given by $\mathbf{v}_i\mathbf{v}_i^T$ multiplied by $\eta(\mathbf{x}_i, \mathbf{x}_j, \mathbf{n}_j)$ defined as

$$\eta(\mathbf{x}_i, \mathbf{x}_j, \mathbf{n}_j) = c_{ij}\left(1 - \left(\mathbf{r}_{ij}^T\mathbf{n}_j\right)^2\right), \tag{1}$$

where

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

is an exponential function using euclidean distance for attenuating the strength based on proximity. σ_d is the size of local neighborhood (or the scale parameter, the only free parameter in tensor voting).

In (1), $\mathbf{r}_{ij} \in \mathbb{R}^d$ is a unit vector at $\mathbf{x}_j$ pointing to $\mathbf{x}_i$, and $1 - (\mathbf{r}_{ij}^T\mathbf{n}_j)^2$ is a squared-sine function[1] for attenuating the contribution according to curvature. Similarly to the original tensor voting framework, (1) favors nearby neighbors that produce small-curvature connections, thus encoding the smoothness constraint. A plot of the 2D version of (1) is shown in Fig. 2b, where $\mathbf{x}_j$ is located at the center of the image and $\mathbf{n}_j$ is aligned with the blue line. The higher the intensity, the higher the value (1) produces at a given pixel location.

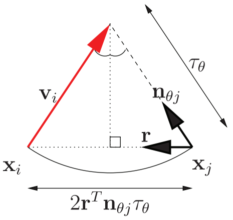


Fig. 3. Illustration of (5). Normal vote $\mathbf{v}_i = \mathbf{v}_\theta(\mathbf{x}_i, \mathbf{x}_j)$ received at $\mathbf{x}_i$ using the arc of the osculating circle between $\mathbf{x}_i$ and $\mathbf{x}_j$, considering one of the normal voters at $\mathbf{x}_j$ is $\mathbf{n}_{\theta j}$. Here, $\mathbf{n}_{\theta j}$ and $\mathbf{r}$ are unit vectors.

Next, consider the general case where the normal $\mathbf{n}_j$ at $\mathbf{x}_j$ is unavailable. Here, let $\mathbf{K}_j$ at $\mathbf{x}_j$ be any second-order symmetric tensor, which is typically initialized as an identity matrix if no normal information is available.

To compute $\mathbf{K}_i$ given $\mathbf{K}_j$, we consider equivalently the set of all possible unit normals $\{\mathbf{n}_{\theta j}\}$ associated with the corresponding length $\{\tau_{\theta j}\}$ which make up $\mathbf{K}_j$ at $\mathbf{x}_j$, where $\{\mathbf{n}_{\theta j}\}$ and $\{\tau_{\theta j}\}$ are, respectively, indexed by all possible directions θ. Each $\tau_{\theta j}\mathbf{n}_{\theta j}$ postulates a normal vote $\mathbf{v}_\theta(\mathbf{x}_i, \mathbf{x}_j)$ at $\mathbf{x}_i$ under the same smoothness constraint prescribed by the corresponding arc of the osculating circle as illustrated in Fig. 3.

Let $\mathbf{S}_{ij}$ be the second-order symmetric **tensor vote** obtained at $\mathbf{x}_i$ due to this complete set of normals at $\mathbf{x}_j$ defined above. We have

$$\mathbf{S}_{ij} = \int_{\mathbf{N}_{\theta j}\in\nu} \mathbf{v}_\theta(\mathbf{x}_i, \mathbf{x}_j)\mathbf{v}_\theta(\mathbf{x}_i, \mathbf{x}_j)^T \eta(\mathbf{x}_i, \mathbf{x}_j, \mathbf{n}_{\theta j}) d\mathbf{N}_{\theta j}, \tag{3}$$

where

$$\mathbf{N}_{\theta j} = \mathbf{n}_{\theta j}\mathbf{n}_{\theta j}^T, \tag{4}$$

and ν is the space containing all possible $\mathbf{N}_{\theta j}$. For example, if ν is 2D, the complete set of unit normals $\mathbf{n}_\theta$ describes a unit circle. If ν is 3D, the complete set of unit normals $\mathbf{n}_\theta$ describes a unit sphere.[2]

In a typical tensor voting implementation, (3) is precomputed as discrete voting fields (e.g., plate and ball voting fields in 3D tensor voting [19]): The integration is implemented by rotating and summing the contributions using matrix addition. Although precomputed once, such discrete approximations involve uniform and dense sampling of tensor votes $\mathbf{n}_\theta\mathbf{n}_\theta^T$ in higher dimensions where the number of dimensions depends on the problem. In the following section, we will prove a closed-form solution to (3), which provides an efficient and exact solution to computing $\mathbf{K}$ without resorting to discrete and dense sampling.

4 Closed-Form Solution

Theorem 1 (Closed-Form Solution to Tensor Voting). *The tensor vote at $\mathbf{x}_i$ induced by $\mathbf{K}_j$ located at $\mathbf{x}_j$ is given by the following closed-form solution:*

[1] $\sin^2\rho = 1 - \cos^2\rho$, where $\cos^2\rho = (\mathbf{r}_{ij}^T\mathbf{n}_j)^2$ and ρ is the angle between $\mathbf{r}_{ij}$ and $\mathbf{n}_j$.

[2] The domain of integration ν represents the space of stick tensors given by $\mathbf{n}_{\theta j}$. Note that $d > 1$; alternatively, it can be understood by expressing $\mathbf{N}_{\theta j}$ using polar coordinates and thus in N dimensions, $\theta = (\phi_1, \phi_2, \ldots, \phi_{n-1})$. It follows naturally that we do not use θ to define the integration domain, because rather than simply writing $\int_{\mathbf{N}_{\theta j}\in\nu}\cdots d\mathbf{N}_{\theta j}$, it would have been $\int_{\phi_1}\int_{\phi_2}\cdots\int_{\phi_{n-1}}\cdots d\phi_{n-1}d\phi_{n-2}\cdots d\phi_1$ making the derivation of the proof of Theorem 1 more complicated.

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

where $\mathbf{K}_j$ *is a second-order symmetric tensor,* $\mathbf{R}_{ij} = \mathbf{I} - 2\mathbf{r}_{ij}\mathbf{r}_{ij}^T$*,* $\mathbf{R}'_{ij} = (\mathbf{I} - \frac{1}{2}\mathbf{r}_{ij}\mathbf{r}_{ij}^T)\mathbf{R}_{ij}$*,* $\mathbf{I}$ *is an identity,* $\mathbf{r}_{ij}$ *is a unit vector pointing from* $\mathbf{x}_j$ *to* $\mathbf{x}_i$*, and* $c_{ij} = \exp(-\frac{\|\mathbf{x}_i - \mathbf{x}_j\|^2}{\sigma_d})$ *with* σ_d *as the scale parameter.*

Proof. For simplicity of notation, set $\mathbf{r} = \mathbf{r}_{ij}$, $\mathbf{n}_\theta = \mathbf{n}_{\theta j}$, and $\mathbf{N}_\theta = \mathbf{N}_{\theta j}$. Now, using the above-mentioned osculating arc connection, $\mathbf{v}_\theta(\mathbf{x}_i, \mathbf{x}_j)$ can be expressed as

$$\mathbf{v}_\theta(\mathbf{x}_i, \mathbf{x}_j) = \big(\mathbf{n}_\theta - 2\mathbf{r}(\mathbf{r}^T\mathbf{n}_\theta)\big)\tau_\theta. \tag{5}$$

Recall that $\mathbf{n}_\theta$ is the unit normal at $\mathbf{x}_j$ with direction θ, and that τ_θ is the length associated with the normal. This vector subtraction equation is shown in Fig. 3, where the roles of $\mathbf{v}_\theta$, $\mathbf{n}_\theta$, $\mathbf{r}$, and τ_θ are illustrated.

Let

$$\mathbf{R} = (\mathbf{I} - 2\mathbf{r}\mathbf{r}^T), \tag{6}$$

where $\mathbf{I}$ is an identity, we can rewrite (3) into the following form:

$$\mathbf{S}_{ij} = c_{ij}\int_{\mathbf{N}_\theta \in \nu} \tau_\theta^2 \mathbf{R}\mathbf{n}_\theta\mathbf{n}_\theta^T\mathbf{R}^T\big(1 - (\mathbf{n}_\theta^T\mathbf{r})^2\big)d\mathbf{N}_\theta. \tag{7}$$

Following the derivation:

$$\begin{aligned}
\mathbf{S}_{ij} &= c_{ij}\int_{\mathbf{N}_\theta \in \nu} \tau_\theta^2 \mathbf{R}\mathbf{n}_\theta\big(1 - (\mathbf{n}_\theta^T\mathbf{r})^2\big)\mathbf{n}_\theta^T\mathbf{R}^T d\mathbf{N}_\theta \\
&= c_{ij}\mathbf{R}\bigg(\int_{\mathbf{N}_\theta \in \nu} \tau_\theta^2 \mathbf{n}_\theta\big(1 - \mathbf{n}_\theta^T\mathbf{r}\mathbf{r}^T\mathbf{n}_\theta\big)\mathbf{n}_\theta^T d\mathbf{N}_\theta\bigg)\mathbf{R}^T \\
&= c_{ij}\mathbf{R}\bigg(\int_{\mathbf{N}_\theta \in \nu} \tau_\theta^2 \mathbf{N}_\theta - \tau_\theta^2\mathbf{N}_\theta\mathbf{r}\mathbf{r}^T\mathbf{N}_\theta d\mathbf{N}_\theta\bigg)\mathbf{R}^T \\
&= c_{ij}\mathbf{R}\bigg(\mathbf{K}_j - \int_{\mathbf{N}_\theta \in \nu} \tau_\theta^2\mathbf{N}_\theta\mathbf{r}\mathbf{r}^T\mathbf{N}_\theta d\mathbf{N}_\theta\bigg)\mathbf{R}^T.
\end{aligned} \tag{8}$$

The integration can be solved by integration by parts. Let $f(\theta) = \tau_\theta^2\mathbf{N}_\theta$, $f'(\theta) = \tau_\theta^2\mathbf{I}$, $g(\theta) = \frac{1}{2}\mathbf{r}\mathbf{r}^T\mathbf{N}_\theta^2$, and $g'(\theta) = \mathbf{r}\mathbf{r}^T\mathbf{N}_\theta$, and note that $\mathbf{N}_\theta^q = \mathbf{N}_\theta$[3] for all $q \in \mathbb{Z}^+$, and $\mathbf{K}_j$, in the most general form, can be expressed as a generic tensor $\int_{\mathbf{N}_\theta \in \nu} \tau_\theta^2 \mathbf{N}_\theta d\mathbf{N}_\theta$. So we have

$$\begin{aligned}
&\int_{\mathbf{N}_\theta \in \nu} \tau_\theta^2\mathbf{N}_\theta\mathbf{r}\mathbf{r}^T\mathbf{N}_\theta d\mathbf{N}_\theta \\
&= [f(\theta)g(\theta)]_{\mathbf{N}_\theta \in \nu} - \int_{\mathbf{N}_\theta \in \nu} f'(\theta)g(\theta)d\mathbf{N}_\theta \\
&= \bigg[\frac{1}{2}\tau_\theta^2\mathbf{N}_\theta\mathbf{r}\mathbf{r}^T\mathbf{N}_\theta^2\bigg]_{\mathbf{N}_\theta \in \nu} - \frac{1}{2}\int_{\mathbf{N}_\theta \in \nu} \tau_\theta^2\mathbf{r}\mathbf{r}^T\mathbf{N}_\theta d\mathbf{N}_\theta \\
&= \frac{1}{2}\int_{\mathbf{N}_\theta \in \nu}\bigg(\tau_\theta^2\frac{d}{d\mathbf{N}_\theta}[\mathbf{N}_\theta]\mathbf{r}\mathbf{r}^T\mathbf{N}_\theta^2 + \tau_\theta^2\mathbf{N}_\theta\frac{d}{d\mathbf{N}_\theta}\big[\mathbf{r}\mathbf{r}^T\mathbf{N}_\theta^2\big]\bigg)d\mathbf{N}_\theta \\
&\quad - \frac{1}{2}\mathbf{r}\mathbf{r}^T\mathbf{K}_j.^4
\end{aligned}$$

Finally, we apply the fact that $\mathbf{N}_\theta^q = \mathbf{N}_\theta$ (for all $q \in \mathbb{Z}^+$) to convert $\frac{d}{d\mathbf{N}_\theta}[\mathbf{r}\mathbf{r}^T\mathbf{N}_\theta^2]$ into $\frac{d}{d\mathbf{N}_\theta}[\mathbf{r}\mathbf{r}^T\mathbf{N}_\theta]$. We obtain

$$\begin{aligned}
&\frac{1}{2}\int_{\mathbf{N}_\theta \in \nu}\big(\tau_\theta^2\mathbf{r}\mathbf{r}^T\mathbf{N}_\theta^2 + \tau_\theta^2\mathbf{N}_\theta\mathbf{r}\mathbf{r}^T\big)d\mathbf{N}_\theta - \frac{1}{2}\mathbf{r}\mathbf{r}^T\mathbf{K}_j \\
&= \frac{1}{2}\big(\mathbf{r}\mathbf{r}^T\mathbf{K}_j + \mathbf{K}_j\mathbf{r}\mathbf{r}^T - \mathbf{r}\mathbf{r}^T\mathbf{K}_j\big) \\
&= \frac{1}{2}\mathbf{K}_j\mathbf{r}\mathbf{r}^T.
\end{aligned} \tag{9}$$

By substituting (9) back to (8), we obtain the result as follows:

$$\mathbf{S}_{ij} = c_{ij}\mathbf{R}\mathbf{K}_j\bigg(\mathbf{I} - \frac{1}{2}\mathbf{r}\mathbf{r}^T\bigg)\mathbf{R}^T. \tag{10}$$

Replace $\mathbf{r}$ by $\mathbf{r}_{ij}$ such that $\mathbf{R}_{ij} = \mathbf{I} - 2\mathbf{r}_{ij}\mathbf{r}_{ij}^T$ and let $\mathbf{R}'_{ij} = (\mathbf{I} - \frac{1}{2}\mathbf{r}_{ij}\mathbf{r}_{ij}^T)\mathbf{R}_{ij}$, we obtain

$$\mathbf{S}_{ij} = c_{ij}\mathbf{R}_{ij}\mathbf{K}_j\mathbf{R}'_{ij}. \tag{11}$$

□

A *structure-aware tensor* $\mathbf{K}_i = \sum_j \mathbf{S}_{ij}$ can thus be assigned at each site $\mathbf{x}_i$. This tensor sum considers both geometric proximity and smoothness constraints in the presence of neighbors $\mathbf{x}_j$ under the chosen scale of analysis. Note also that (11) is an exact equivalent of (3), or (7), that is, the first principle. Since the first principle produces a positive semidefinite matrix, (11) still produces a positive semidefinite matrix.

In tensor voting, eigen-decomposition is applied to a structure-aware tensor. In three dimensions, the eigensystem has eigenvalues $\lambda_1 \geq \lambda_2 \geq \lambda_3 \geq 0$ with the corresponding eigenvectors $\hat{e}_1$, $\hat{e}_2$, and $\hat{e}_3$. $\lambda_1 - \lambda_2$ denotes surface saliency with normal direction indicated by $\hat{e}_1$; $\lambda_2 - \lambda_3$ denotes curve saliency with tangent direction indicated by $\hat{e}_3$; junction saliency is indicated by λ_3.

While it may be difficult to observe any geometric intuition directly from this closed-form solution, the geometric meaning of the closed-form solution has been described by (3) (or (7), the first principle) since (11) is equivalent to (3). Note that our solution is different from, for instance, [21], where the N-D formulation is approached from a more geometric point of view.

As will be shown in the next section on EMTV, the inverse of $\mathbf{K}_j$ is used. In case of a perfect stick tensor, which can be equivalently represented as a rank-1 matrix, does not have an inverse. Similar in spirit where a Gaussian function can be interpreted as an impulse function associated with a spread representing uncertainty, a similar statistical approach is adopted here in characterizing our tensor inverse. Specifically, the uncertainty is incorporated using a ball tensor, where $\epsilon\mathbf{I}$ is added to $\mathbf{K}_j$, ϵ is a small positive constant (0.001), and $\mathbf{I}$ an identity matrix. Fig. 1 shows a tensor and its inverse for some selected cases. The following corollary regarding the inverse of $\mathbf{S}_{ij}$ is useful:

Corollary 1. *Let* $\mathbf{R}''_{ij} = \mathbf{R}_{ij}(\mathbf{I} + \mathbf{r}_{ij}\mathbf{r}_{ij}^T)$ *and also note that* $\mathbf{R}_{ij}^{-1} = \mathbf{R}_{ij}$*, the corresponding inverse of* $\mathbf{S}_{ij}$ *is*

$$\mathbf{S}'_{ij} = c_{ij}^{-1}\mathbf{R}''_{ij}\mathbf{K}_j^{-1}\mathbf{R}_{ij}. \tag{12}$$

[3] The derivation is as follows: $\mathbf{N}_\theta^q = \mathbf{n}_\theta\mathbf{n}_\theta^T\mathbf{n}_\theta\mathbf{n}_\theta^T \cdots \mathbf{n}_\theta\mathbf{n}_\theta^T = \mathbf{n}_\theta \cdot 1 \cdot 1 \cdots 1 \cdot \mathbf{n}_\theta^T = \mathbf{n}_\theta\mathbf{n}_\theta^T = \mathbf{N}_\theta$.

[4] Here, we rewrite the first term by the product rule for derivative and the fundamental theorem of calculus and then express part of the second term by a generic tensor. We obtain:

$$\frac{1}{2}\int_{\mathbf{N}_\theta \in \nu}\frac{d}{d\mathbf{N}_\theta}\big[\tau_\theta^2\mathbf{N}_\theta\mathbf{r}\mathbf{r}^T\mathbf{N}_\theta^2\big]d\mathbf{N}_\theta - \frac{1}{2}\mathbf{r}\mathbf{r}^T\mathbf{K}_j.$$

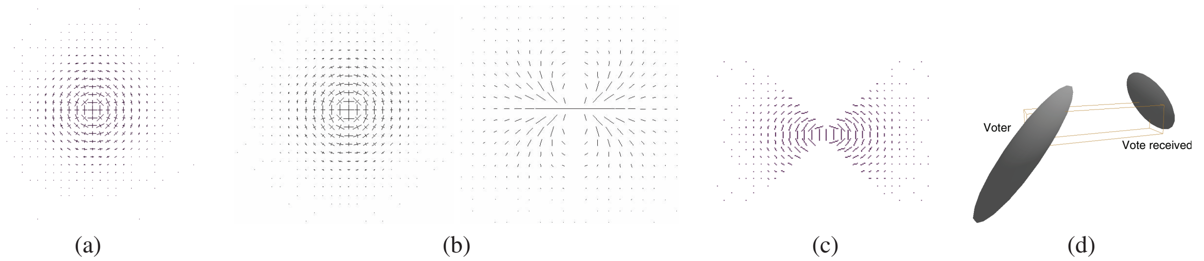


Fig. 4. (a) **3D ball voting field**. A slice generated using the closed-form solution (11), which has similar tensor orientations (but different tensor strengths) as the ball voting field in [19]. (b) **3D plate voting field**. Left: A cut of the voting field (direction of $\hat{e}_3$ normal to the page). Right: A cut of the same voting field, showing the $(\lambda_2 - \lambda_3)\hat{e}_3$ component (i.e., component parallel to the tangent direction). The field is generated by using (11), showing similar tensor orientations as the plate voting field in [19]. (c) **3D stick voting field**. A slice after zeroing out votes lying in the 45-degree zone as done in [19]. The stick tensor orientations shown in the figure are identical to those in the 3D stick voting field in [19]. (d) Vote computation using the closed-form solution in one single step by (11).

Proof. This corollary can simply be proved by applying inverse to (11). □

Note the initial $\mathbf{K}_j$ can be either derived when input direction is available, or simply assigned as an identity matrix otherwise.

4.1 Examples

Using (11), given any input $\mathbf{K}_j$ at site j which is a second-order symmetric tensor, the output tensor $\mathbf{S}_{ij}$ can be computed directly. Note that $\mathbf{R}_{ij}$ is a $d \times d$ matrix of the same dimensionality d as $\mathbf{K}_j$. To verify our closed-form solution, we perform the following to compare with the voting fields used by the original tensor voting framework (Fig. 4):

1. Set $\mathbf{K}$ to be an identity (ball tensor) in (11) and compute all votes $\mathbf{S}$ in a neighborhood. This procedure generates the ball voting field, Fig. 4a.
2. Set $\mathbf{K}$ to be a plate tensor
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$
in (11) and compute all votes $\mathbf{S}$ in a neighborhood. This procedure generates the plate voting field, Fig. 4b.
3. Set $\mathbf{K}$ to be a stick tensor
$$\begin{bmatrix} 1 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$
in (11) and compute all votes $\mathbf{S}$ in a neighborhood. This procedure generates the stick voting field, Fig. 4c.
4. Set $\mathbf{K}$ to be *any* generic second-order tensor in (11) to compute a tensor vote $\mathbf{S}$ at a given site. We do not need a voting field, or the somewhat complex procedure described in [21]. In one single step using the closed-form solution (11), we obtain $\mathbf{S}$ as shown in Fig. 4d.

Note that the stick voting field generation is the same as the closed-form solution given by the arc of an osculating circle. On the other hand, since the closed-form solution does not remove votes lying beyond the 45-degree zone as done in the original framework, it is useful to compare the ball voting field generated using the CFTV and the original framework. Fig. 5 shows the close ups of the ball voting fields generated using the original framework and CFTV. As anticipated, the tensor orientations are almost the same (with the maximum angular deviation at 4.531 degrees), while the tensor strength is different due to the use of different decay functions. The new computation results in perfect vote orientations which are radial, and the angular discrepancies are due to the discrete approximations in the original solution.

While the above illustrates the usage of (11) in three dimensions, the equation applies to any dimensions d. All of the $\mathbf{S}$s returned by (11) are second-order symmetric tensors and can be decomposed using eigen-decomposition. The implementation of (11) is a matter of a few lines of C++ code.

Our "voting without voting fields" method is uniform to any input tensors $\mathbf{K}_j$ that are second-order symmetric tensor in its closed-form expressed by (11), where formal mathematical operation can be applied on this compact equation, which is otherwise difficult on the algorithmic procedure described in previous tensor voting papers. Notably, using the closed-form solution, we are now able to prove mathematically the convergence of tensor voting in the next section.

4.2 Time Complexity

Akin to the original tensor voting formalism, each site (input or noninput) communicates with each other on a Markov random field in a broad sense, where the number of edges depends on the scale of analysis, parameterized by σ_d in (2). In our implementation, we use an efficient data structure such as ANN tree [1] to access a constant number of neighbors $\mathbf{x}_j$ of each $\mathbf{x}_i$. It should be noted that under a large scale of analysis

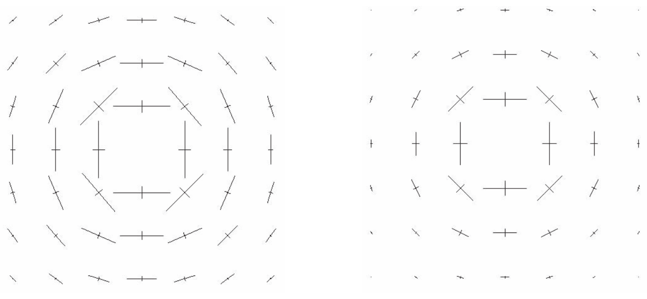


Fig. 5. Close-ups of ball voting fields generated using the original tensor voting framework (left) and CFTV (right).

where the number of neighbors is sufficiently large, a similar number of neighbors are accessed in ours and the original tensor voting implementation.

The speed of accessing nearest neighbors can be greatly increased (polylogarithmic) by using ANN, thus making the computation of a structure-aware tensor efficient. Note that the running time for this implementation of the closed-from solution is $O(d^3)$, while the running time for the original tensor voting is $O(u^{d-1})$, where d is the dimension of the space and u is the number of sampling directions for a given dimension. Because of this, a typical TV implementation precomputes and stores the dense tensor fields. For example, when $d = 3$ and $u = 180$ for high accuracy, our method requires 27 operation units, while a typical TV implementation requires 32,400 operation units. Given 1,980 points and the same number of neighbors, the time to compute a structure-aware tensor using our method is about 0.0001 second; it takes about 0.1 second for a typical TV implementation to output the corresponding tensor. The measurement was performed on a computer running on a core duo 2 GHz CPU with 2 GB RAM.

Note that the asymptotic running time for the improved TV in [21] is $O(d\gamma^2)$ since it applies the Gramm-Schmidt process to perform component decomposition, where γ is the number of linearly independent set of the tensors. In most of the cases, $\gamma = d$. So, the running time for our method is comparable to [21]. However, their approach does not have a precise mathematical solution.

5 MRFTV

We have proven CFTV for the special theory of tensor voting, or the "first voting pass" for structure inference. Conventionally, tensor voting was done in two passes, where the second pass was used for structure propagation in the preferred direction after disabling the ball component in the structure-aware tensor. What happens if more tensor voting passes are applied? This has never been answered properly.

In this section, we provide a convergence proof for tensor voting based on CFTV: The structure-aware tensor obtained at each site achieves a stationary state upon convergence. Our convergence proof makes use of Markov random fields, thus termed as MRFTV.

It should be noted that the original tensor voting formulation is also constructed on an MRF according to the broad definition, since random variables (that is, the tensors after voting) are defined on the nodes of an undirected graph in which each node is connected to all neighbors within a fixed distance. On the other hand, without CFTV, it was previously difficult to write down an objective function and to prove the convergence. One caveat to note in the following is that we do not disable the ball component in each iteration, which will be addressed in the future in developing the general theory of tensor voting in structure propagation. As we will demonstrate, MRFTV does not smooth out important features (Fig. 6) and still possesses high outlier rejection ability (Fig. 13).

Recall in MRF, a Markov network is a graph consisting of two types of nodes—a set of hidden variables $\mathbf{E}$ and a set of observed variables $\mathbf{O}$, where the edges of the graph are described by the following posterior probability $\mathrm{P}(\mathbf{E}|\mathbf{O})$ with standard Bayesian framework:

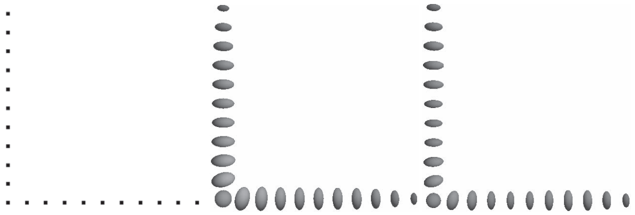


Fig. 6. Convergence of MRF-TV. From left to right: Input points, result after two passes, result after convergence (10 iterations). Visually, this simple example distinguishes tensor voting from smoothing as the sharp orientation discontinuity is preserved upon convergence.

$$\mathrm{P}(\mathbf{E}|\mathbf{O}) \propto \mathrm{P}(\mathbf{O}|\mathbf{E})\mathrm{P}(\mathbf{E}). \qquad (13)$$

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

$$\mathrm{P}(\mathbf{O}|\mathbf{E}) \propto \prod_i \mathrm{p}(\tilde{\mathbf{K}}_i|\mathbf{K}_i) = \prod_i e^{-\frac{\|\mathbf{K}_i - \tilde{\mathbf{K}}_i\|_F^2}{\sigma_h}}, \qquad (14)$$

$$\mathrm{P}(\mathbf{E}) \propto \prod_i \prod_{j \in \mathcal{N}(i)} \mathrm{p}(\mathbf{S}_{ij}|\mathbf{K}_i) \qquad (15)$$

$$= \prod_i \prod_{j \in \mathcal{N}(i)} e^{-\frac{\|\mathbf{K}_i - \mathbf{S}_{ij}\|_F^2}{\sigma_s}}, \qquad (16)$$

where $\|\cdot\|_F$ is the 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.

Note that we use the Frobenius norm to encode tensor orientation consistency as well as to reflect the necessary vote saliency information including distance and continuity attenuation. For example, suppose we have a unit stick tensor at $\mathbf{x}_i$ and a stick vote (received at $\mathbf{x}_i$) which is parallel to it but with magnitude equal to 0.8. In another scenario, $\mathbf{x}_i$ receives from a voter farther away a stick vote with the same orientation but magnitude being equal to 0.2. The Frobenius norm reflects the difference in saliency despite the perfect orientation consistency in both cases. Notwithstanding, it is arguable that the Frobenius norm may not be the perfect solution to encode orientation consistency constraint in the pertinent equations, while this current form works acceptably well in our experiments in practice.

By taking the logarithm of (13), we obtain the following energy function:

$$E(\mathbf{E}) = \sum_i \|\mathbf{K}_i - \tilde{\mathbf{K}}_i\|_F^2 + g \sum_i \sum_{j \in \mathcal{N}(i)} \|\mathbf{K}_i - \mathbf{S}_{ij}\|_F^2, \qquad (17)$$

where $g = \frac{\sigma_h}{\sigma_s}$. Theoretically, this *quadratic* energy function can be directly solved 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 (17) (w.r.t. to $\mathbf{K}_i$), the following update rule is obtained:

TABLE 1
Comparison with Ground Truth for the Example in Fig. 6

	Tensor orientation angular error (in deg)		
	min	max	avg
2 passes	0.000000	0.115430	0.027264
converged	0.000000	0.045051	0.010683

$$\mathbf{K}_i^* = \left(\tilde{\mathbf{K}}_i + 2g\sum_{j\in\mathcal{N}(i)}\mathbf{S}_{ij}\right)\left(\mathbf{I} + g\sum_{j\in\mathcal{N}(i)}\left(\mathbf{I} + c_{ij}^2{\mathbf{R}'_{ij}}^2\right)\right)^{-1}, \tag{18}$$

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

$$\mathbf{K}_i^{(m+1)} = (1-q)\mathbf{K}_i^{(m)} + q\mathbf{K}_i^*, \tag{19}$$

where $1 < q < 2$ is the SOR weight and m is the iteration number. After each iteration, we normalize $\mathbf{K}_i$ such that the eigenvalues of the corresponding eigensystem are within the range $(0, 1]$.

The above proof on convergence of MRF-TV shows that structure-aware tensors achieve stationary states after a finite number Gauss-Seidel iterations in the above formulation. It also dispels a common pitfall that tensor voting is similar in effect to smoothing. Using the same scale of analysis (that is, in (2)) and the same σ_h, σ_s in each iteration, tensor saliency and orientation will both converge. We observe that the converged tensor orientation is, in fact, similar to that obtained after two voting passes using the original framework, where the orientations at curve junctions are not smoothed out. See Fig. 6 for an example where sharp orientation discontinuity is not smoothed out when tensor voting converges. Here, λ_1 of each structure-aware tensor is not normalized to 1 for visualizing its structure saliency after convergence. Table 1 summarizes the quantitative comparison with the ground-truth orientation.

6 EMTV

Previously, while tensor voting was capable of rejecting outliers, it fell short of producing accurate parameter estimation, explaining the use of RANSAC in the final parameter estimation step after outlier rejection [27].

This section describes the EMTV algorithm for optimizing 1) the structure-aware tensor $\mathbf{K}$ at each input site, and 2) the parameters of a single plane $\mathbf{h}$ of *any* dimensionality containing the inliers. This algorithm will be applied to stereo matching.

We first formulate the three constraints to be used in EMTV. These constraints are not mutually exclusive, where knowing the values satisfying one constraint will help computing the values of the others. However, in our case, they are all unknowns, so EM is particularly suitable for their optimization since the expectation calculation and parameter estimation are solved alternately.

6.1 Constraints

Data constraint. Suppose we have a set of clean data. One necessary objective is to minimize the following for all $\mathbf{x}_i \in \mathbb{R}^d$ with $d > 1$:

$$\left\|\mathbf{x}_i^T\mathbf{h}\right\|, \tag{20}$$

where $\mathbf{h} \in \mathbb{R}^d$ is a unit vector representing the plane (or the model) to be estimated.[5] This is a typical data term that measures the faithfulness of the input data to the fitting plane.

Orientation consistency. The plane being estimated is defined by the vector $\mathbf{h}$. Since the tensor $\mathbf{K}_i \in \mathbb{R}^d \times \mathbb{R}^d$ encodes structure awareness, if $\mathbf{x}_i$ is an inlier, the orientation information encoded by $\mathbf{K}_i$ and $\mathbf{h}$ has to be consistent. That is, the variance $\mathbf{h}^T\mathbf{K}_i^{-1}\mathbf{h}$ produced by $\mathbf{h}$ should be minimal. Otherwise, $\mathbf{x}_i$ might be generated by other models even if it minimizes (20). Mathematically, we minimize

$$\left\|\mathbf{h}^T\mathbf{K}_i^{-1}\mathbf{h}\right\|. \tag{21}$$

Neighborhood consistency. While the estimated $\mathbf{K}_i$ helps to indicate inlier/outlier information, $\mathbf{K}_i$ has to be consistent with the local structure imposed by its neighbors (when they are known). If $\mathbf{K}_i$ is consistent with $\mathbf{h}$ but not the local neighborhood, either $\mathbf{h}$ or $\mathbf{K}_i$ is wrong. In practice, we minimize the following Frobenius norm as in (14)-(16):

$$\left\|\mathbf{K}_i^{-1} - \mathbf{S}'_{ij}\right\|_F. \tag{22}$$

In the spirit of MRF, $\mathbf{S}'_{ij}$ encodes the tensor information within $\mathbf{x}_i$'s neighborhood, thus a natural choice for defining the term for measuring neighborhood orientation consistency. This is also useful, as we will see, to the M-step of EMTV which makes the MRF assumption.

The above three constraints will interact with each other in the proposed EM algorithm.

6.2 Objective Function

Define $\mathbf{O} = \{o_i = \mathbf{x}_i | i = 1, \ldots, N\}$ to be the set of observations. Our goal is to optimize $\mathbf{h}$ and $\mathbf{K}_i^{-1}$ given $\mathbf{O}$. Mathematically, we solve the objective function:

$$\Lambda^* = \arg\max_{\Lambda} P(\mathbf{O}, \mathbf{R}|\Lambda), \tag{23}$$

where $P(\mathbf{O}, \mathbf{R}|\Lambda)$ is the complete-data likelihood to be maximized, $\mathbf{R} = \{r_i\}$ is a set of hidden states indicating if observation o_i is an outlier ($r_i = 0$) or inlier ($r_i = 1$), and $\Lambda = \{\{\mathbf{K}_i^{-1}\}, \mathbf{h}, \alpha, \sigma, \sigma_1, \sigma_2\}$ is a set of parameters to be estimated. α, σ, σ_1, and σ_2 are parameters imposed by some distributions, which will be explained shortly by using an equation to be introduced.[6] Our EM algorithm estimates an optimal Λ^* by finding the value of the complete-data log likelihood with respect to $\mathbf{R}$ given $\mathbf{O}$ and the current estimated parameters Λ':

$$Q(\Lambda, \Lambda') = \sum_{\mathbf{R}\in\psi} \log P(\mathbf{O}, \mathbf{R}|\Lambda) P(\mathbf{R}|\mathbf{O}, \Lambda'), \tag{24}$$

[5] Note that, in some cases, the underlying model is represented in this form $\mathbf{x}_i^T\mathbf{h} - z_i$, where we can rearrange it into the form given by (20). For example, expand $\mathbf{x}_i^T\mathbf{h} - z_i$ into $ax_i + by_i + 1z_i = 0$, which can written in the form of (20).

[6] See the M-step in (30).

where ψ is a space containing all possible configurations of $\mathbf{R}$ of size N. Although EM does not guarantee a global optimal solution theoretically, because CFTV provides good initialization we will demonstrate empirically that reasonable results can be obtained.

6.3 Expectation (E-Step)

In this section, the marginal distribution $p(r_i|o_i, \Lambda')$ will be defined so that we can maximize the parameters in the next step (M-Step) given the current parameters.

If $r_i = 1$, the observation o_i is an inlier and therefore minimizes the first two conditions ((20) and (21)) in Section 6.1, that is, the data and orientation constraints. In both cases, we assume that inliers follow a Gaussian distribution which explains the use of $\mathbf{K}_i^{-1}$ instead of $\mathbf{K}_i$.[7] We model $\mathrm{p}(o_i|r_i, \Lambda')$ as

$$\propto \begin{cases} \exp\Big(-\frac{\|\mathbf{x}_i^T\mathbf{h}\|^2}{2\sigma^2}\Big)\exp\Big(-\frac{\|\mathbf{h}^T\mathbf{K}_i^{-1}\mathbf{h}\|}{2\sigma_1^2}\Big), & \text{if } r_i = 1; \\ \dfrac{1}{C}, & \text{if } r_i = 0. \end{cases} \tag{25}$$

We assume that outliers follow uniform distribution, where C is a constant that models the distribution. Let C_m be the maximum dimension of the bounding box of the input. In practice, $C_m \le C \le 2C_m$ produces similar results.

Since we have no prior information on a point being an inlier or outlier, we may assume that the mixture probability of the observations $p(r_i = 1) = p(r_i = 0)$ equals a constant $\alpha = 0.5$ such that we have no bias to either category (inlier/outlier). For generality in the following, we will include α in the derivation.

Define $w_i = p(r_i|o_i, \Lambda')$ to be the probability of o_i being an inlier. Then,

$$\begin{aligned} w_i &= \mathrm{p}(r_i = 1|o_i, \Lambda') = \frac{\mathrm{p}(o_i, r_i = 1|\Lambda')}{\mathrm{p}(o_i|\Lambda')} \\ &= \frac{\alpha\beta\exp\Big(-\frac{\|\mathbf{x}_i^T\mathbf{h}\|^2}{2\sigma^2}\Big)\exp\Big(-\frac{\|\mathbf{h}^T\mathbf{K}_i^{-1}\mathbf{h}\|}{2\sigma_1^2}\Big)}{\alpha\beta\exp\Big(-\frac{\|\mathbf{x}_i^T\mathbf{h}\|^2}{2\sigma^2}\Big)\exp\Big(-\frac{\|\mathbf{h}^T\mathbf{K}_i^{-1}\mathbf{h}\|}{2\sigma_1^2}\Big) + \frac{1-\alpha}{C}}, \end{aligned} \tag{26}$$

where $\beta = \frac{1}{2\sigma\sigma_1\pi}$ is the normalization term.

6.4 Maximization (M-Step)

In the M-Step, we maximize (24) using w_i obtained from the E-Step. Since neighborhood information is considered, we model $P(\mathbf{O}, \mathbf{R}|\Lambda)$ as an MRF:

$$P(\mathbf{O}, \mathbf{R}|\Lambda) = \prod_i \prod_{j\in\mathcal{G}(i)} p(r_i|r_j, \Lambda)p(o_i|r_i, \Lambda), \tag{27}$$

where $\mathcal{G}(i)$ is the set of neighbors of i. In theory, $\mathcal{G}(i)$ contains all the input points except i since c_{ij} in (2) is always nonzero (because of the long tail of the Gaussian distribution). In practice, we can prune away the points in $\mathcal{G}(i)$ where the values of c_{ij} are negligible. This can greatly reduce the size of the neighborhood. Again, using ANN tree [1], the speed of searching for nearest neighbors can be greatly increased.

Let us examine the two terms in (27). $p(o_i|r_i, \Lambda)$ has been defined in (25). We define $p(r_i|r_j, \Lambda)$ here. Using the third condition mentioned in (22), we have

$$p(r_i|r_j, \Lambda) = \exp\left(-\frac{\left\|\mathbf{K}_i^{-1} - \mathbf{S}'_{ij}\right\|_F^2}{2\sigma_2^2}\right). \tag{28}$$

We are now ready to expand (24). Since r_i can only assume two values (0 or 1), we can rewrite $Q(\Lambda, \Lambda')$ in (24) into the following form:

$$\sum_{t\in\{0,1\}} \log\left(\prod_i \prod_{j\in\mathcal{G}(i)} p(r_i = t|r_j, \Lambda)p(o_i|r_i = t, \Lambda)\right) P(\mathbf{R}|\mathbf{O}, \Lambda').$$

After expansion,

$$\begin{aligned} Q(\Lambda, \Lambda') = &\sum_i \log\left(\alpha\frac{1}{\sigma\sqrt{2\pi}}\exp\left(-\frac{\|\mathbf{x}_i^T\mathbf{h}\|^2}{2\sigma^2}\right)\right)w_i \\ &+ \sum_i \log\left(\frac{1}{\sigma_1\sqrt{2\pi}}\exp\left(-\frac{\|\mathbf{h}^T\mathbf{K}_i^{-1}\mathbf{h}\|}{2\sigma_1^2}\right)\right)w_i \\ &+ \sum_i \log\left(\exp\left(-\frac{\|\mathbf{K}_i^{-1} - \mathbf{S}'_{ij}\|_F^2}{2\sigma_2^2}\right)\right)w_i w_j \\ &+ \sum_i \log\left(\frac{1-\alpha}{C}\right)(1 - w_i). \end{aligned} \tag{29}$$

To maximize (29), we set the first derivative of Q with respect to $\mathbf{K}_i^{-1}$, $\mathbf{h}$, α, σ, σ_1, and σ_2 to zero, respectively, to obtain the following set of update rules:

$$\begin{aligned} \alpha &= \frac{1}{N}\sum_i w_i, \\ \mathbf{K}_i^{-1} &= \frac{1}{\sum_{j\in\mathcal{G}(i)} w_j}\left(\sum_{j\in\mathcal{G}(i)} \mathbf{S}'_{ij}w_j - \frac{\sigma_2^2}{2\sigma_1^2}\mathbf{h}\mathbf{h}^T w_i\right), \\ \min\|\mathbf{M}\mathbf{h}\| &\qquad \text{subject to } \|\mathbf{h}\| = 1 \\ \sigma^2 &= \frac{\sum_i \left\|\mathbf{x}_i^T\mathbf{h}\right\|^2 w_i}{\sum_i w_i}, \\ \sigma_1^2 &= \frac{\sum_i \left\|\mathbf{h}^T\mathbf{K}_i^{-1}\mathbf{h}\right\| w_i}{\sum_i w_i}, \\ \sigma_2^2 &= \frac{\sum_i \sum_{j\in\mathcal{G}(i)} \left\|\mathbf{K}_i^{-1} - \mathbf{S}'_{ij}\right\|_F^2 w_i w_j}{\sum_i w_i}, \end{aligned} \tag{30}$$

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. Equation (30) constitutes the set of update rules for the M-step.

In each iteration, after the update rules have been executed, we normalize $\mathbf{K}_i^{-1}$ onto the feasible solution space by normalization, that is, the eigenvalues of the corresponding eigensystem are within the range $(0, 1]$. Also, $\mathbf{S}'_{ij}$ will be updated with the newly estimated $\mathbf{K}_i^{-1}$.

6.5 Implementation and Initialization

In summary, (12), (26), and (30) are all the equations needed to implement EMTV and therefore the implementation is straightforward.

Noting that initialization is important to an EM algorithm, to initialize EMTV we set σ_1 to be a very large value,

[7] Although a linear structure is being optimized here, the inliers together may describe a structure that does not necessarily follow any particular model. Each inlier may not exactly lie on this structure where the misalignment follows the Gaussian distribution.

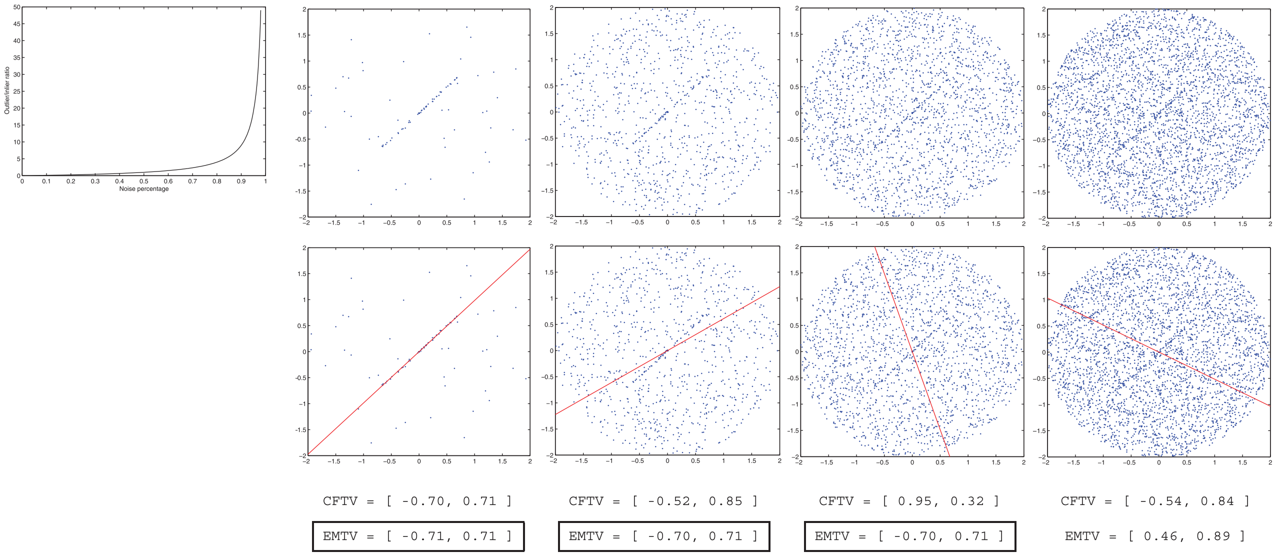


Fig. 7. The top-left subfigure shows the plot of $\frac{R}{R+1}$. The four 2D data sets shown here have OI ratios $[1, 20, 45, 80]$, respectively, which correspond to outlier percentages [50, 95, 98, 99 percent]. Our EMTV can tolerate OI ratios ≤ 51 in this example. The original input, the estimated line after the first EMTV iteration using CFTV to initialize the algorithm, and the line parameters after the first EMTV iteration and final EMTV convergence were shown. The ground-truth parameter is $[-0.71, 0.71]$.

$\mathbf{K}_i = \mathbf{I}$ and $w_i = 1$ for all i. $\mathbf{S}'_{ij}$ is initialized to be the inverse of $\mathbf{S}_{ij}$, computed using the closed-form solution presented in the previous section. These initialization values mean that at the beginning we have no preference for the surface orientation. So, all the input points are initially considered as inliers. With such initialization, we execute the first and the second rules in (30) in sequence. Note that when the first rule is being executed, the term involving $\mathbf{h}$ is ignored because of the large σ_1; thus we can obtain $\mathbf{K}_i^{-1}$ for the second rule. After that, we can start executing the algorithm from the E-step. This initialization procedure is used in all the experiments in the following sections. Fig. 7 shows the result after the first EMTV iteration on an example; note in particular that even though the initialization is at times not close to the solution, our EMTV algorithm can still converge to the desired ground-truth solution.

7 Experimental Results

First, quantitative comparison will be studied to evaluate EMTV with well-known algorithms: RANSAC [7], ASSC [28], and TV [19]. In addition, we also provide the result using the least squares method as a baseline comparison. Second, we apply our method to real data with synthetic outliers and/or noise where the ground truth is available, and perform comparison. Third, more experiments on multiview stereo matching on real images are performed.

As we will show, EMTV performed the best in highly corrupted data because it is designed to seek one linear structure of known type (as opposed to multiple, potentially nonlinear structures of unknown type). The use of orientation constraints, in addition to position constraints, makes EMTV superior to the random sampling methods as well.

Outlier/inlier (OI) ratio. We will use the *outlier/inlier ratio* to characterize the outlier level, which is related to the outlier percentage $Z \in [0, 1]$:

$$Z = \frac{R}{R+1}, \tag{31}$$

where R is the OI ratio. Fig. 7 shows a plot of $Z = \frac{R}{R+1}$, indicating that it is much more difficult for a given method to handle the same percentage increase in outliers as the value of Z increases. Note the rapid increase in the number of outliers as Z increases from 50 to 99 percent. That is, it is more difficult for a given method to tolerate an addition of, say, 20 percent outliers when Z is increased from 70 to 90 percent than from 50 to 70 percent. Thus, the OI ratio gives more insight in studying an algorithm's performance on severely corrupted data.

7.1 Robustness

We generate a set of 2D synthetic data to evaluate the performance on line fitting, by randomly sampling 44 points from a line within the range $[-1, -1] \times [1, 1]$ where the locations of the points are contaminated by Gaussian noise of 0.1 standard deviation. Random outliers were added to the data with different OI ratios.

The data set is then partitioned into two:

- SET 1: OI ratio $\in [0.1, 1]$ with step size 0.1,
- SET 2: OI ratio $\in [1, 100]$ with step size 1.

In other words, the partition is done at 50 percent outliers. Note from the plot in Fig. 7 that the number of outliers increases rapidly after 50 percent outliers. Sample data sets with different OI ratios are shown at the top of Fig. 7. Outliers were added within a bounding circle of radius 2. In particular, the bottom of Fig. 7 shows the result of the first EMTV iteration upon initialization using CFTV.

The input scale which is used in RANSAC, TV and EMTV was estimated automatically by TSSE proposed in [28]. Note, in principle these scales are not the same because TSSE estimates the scales of residuals in the normal space. Therefore, the scales estimated by TSSE used in TV and EMTV are only approximations. As we will demonstrate

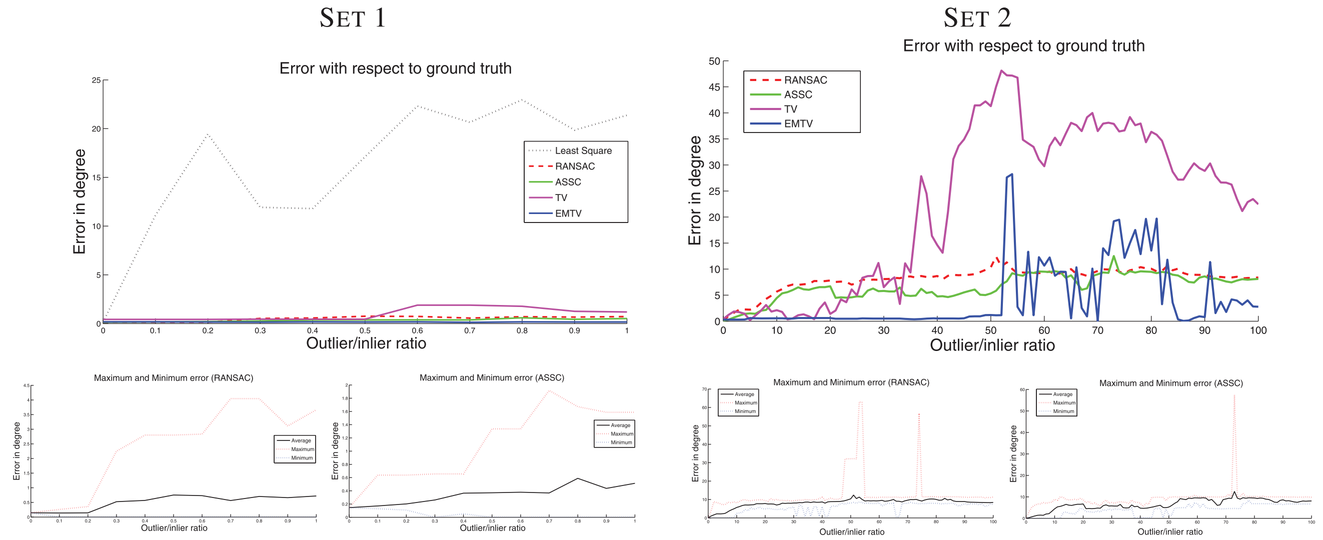


Fig. 8. Error plots for SET 1 (OI ratio $= [0.1, 1]$, up to 50 percent outliers) and SET 2 (OI ratio $= [1, 100]$, $\geq 50\%$ outliers). Left: For SET 1, all the tested methods except for the least-squares demonstrated reliable results. EMTV is deterministic and converges quickly, capable of correcting Gaussian noise inherent in the inliers and rejecting spurious outliers, and resulting in the almost-zero error curve. Right: For SET 2, EMTV still has an almost-zero error curve up to an OI ratio of 51 ($\simeq 98.1\%$ outliers). We ran 100 trials in RANSAC and ASSC and averaged the results. The maximum and minimum errors of RANSAC and ASSC are shown below each error plot.

below, even with such rough approximations, EMTV still performs very well, showing that it is not sensitive to scale inaccuracy, a nice property of tensor voting which will be shown in an experiment to be detailed shortly. Note that ASSC [28] does not require any input scale.

SET 1—Refer to the *left* of Fig. 8, which shows the error produced by various methods tested on SET 1. The error is measured by the angle between the estimated line and the ground truth. Except for the least squares method, we observe that all the tested methods (RANSAC, ASSC, TV, and EMTV) performed very well with OI ratios ≤ 1. For RANSAC and ASSC, all the detected inliers were finally used in parameter estimation. Note that the errors measured for RANSAC and ASSC were the average errors in 100 executions.[8] Fig. 8 also shows the maximum and minimum errors of the two methods after running 100 trials. EMTV does not have such maximum and minimum error plots because it is deterministic.

Observe that the errors produced by our method are almost zero in SET 1. EMTV is deterministic and converges quickly, capable of correcting Gaussian noise inherent in the inliers and rejecting spurious outliers, and resulting in the almost-zero error curve. RANSAC and ASSC have error < 0.6 degree , which is still very acceptable.

SET 2—Refer to the *right* of Fig. 8, which shows the result for SET 2, from which we can distinguish the performance of the methods. TV breaks down at OI ratios ≥ 20. After that, the performance of TV is unpredictable. EMTV breaks down at OI ratios ≥ 51, showing greater robustness than TV in this experiment due to the EM parameter fitting procedure.

The performances of RANSAC and ASSC were quite stable where the average errors are within 4 and 7 degrees over the whole spectrum of OI ratios considered. The maximum and minimum errors are shown in the bottom of Fig. 8, which shows that they can be very large at times. EMTV produces almost zero errors with OI ratio ≤ 51, but then breaks down with unpredictable performance. From the experiments on SET 1 and SET 2 we conclude that EMTV is robust up to an OI ratio of 51 ($\simeq 98.1\%$ outliers).

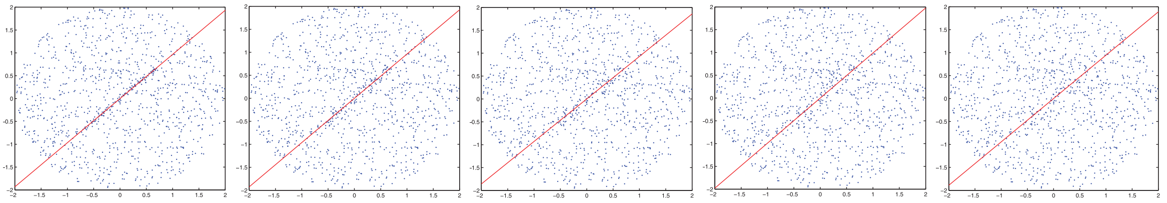
Insensitivity to choice of scale. We studied the errors produced by EMTV with different scales σ_d ((2)), given OI ratio of 10 ($\simeq 91\%$ outliers). Even in the presence of many outliers, EMTV broke down only when $\sigma_d \simeq 0.7$ (the ground truth σ_d is 0.1), which indicates that our method is not sensitive to large deviations of scale. Note that the scale parameter can sometimes be automatically estimated (e.g., by modifying the original TSSE to handle tangent space) as was done in the previous experiment.

Large measurement errors. In this experiment, we increased the measurement error by increasing the standard deviation (s.d.) from 0.01 to 0.29, while keeping the OI ratio equal to 10 and the location of the outliers fixed. Some of the input data sets are depicted in Fig. 9, showing that the inliers are less salient as the standard deviation (s.d.) increases. A similar experiment was also performed in [20]. Again, we compared our method with RANSAC, ASSC, and TV.

According to the error plot at the *top* of Fig. 10, TV is very sensitive to the change of s.d.: When the s.d. is greater than 0.03, the performance is unpredictable. With increasing s.d., the performances of RANSAC and ASSC degrade gracefully, while ASSC always outperforms RANSAC. The *bottom* of Fig. 10 shows the corresponding maximum and minimum error in 100 executions.

On the other hand, we observe the performance of EMTV (with $\sigma_d = 0.05$) is extremely steady and accurate when s.d. < 0.15. After that, although its error plot exhibits some perturbation, the errors produced are still small and the performance is quite stable compared with other methods.

[8] We executed the algorithm 100 times. In each execution, iterative random sampling was done where the desired probability of choosing at least one sample free from outliers was set to 0.99 (default value).



s.d. = 0.03 s.d. = 0.06 s.d. = 0.15 s.d. = 0.2 s.d. = 0.25

Fig. 9. Inputs containing various measurement errors, with $\mathrm{OI\ ratio} = 10$ and fixed outliers location. The estimated models (depicted by the red lines) obtained using EMTV are overlayed on the inputs. Notice the line cluster becomes less salient when $\mathrm{s.d.} = 0.25$.

7.2 Fundamental Matrix Estimation

Given an image pair with $p \geq 8$ correspondences $\mathcal{P} = \{(\mathbf{u}_i, \mathbf{u'}_i) | 8 \leq i \leq p\}$, the goal is to estimate the 3×3 fundamental matrix $\mathbf{F} = [f]_{a,b}$, where $a, b \in \{1, 2, 3\}$, such that

$$\mathbf{u'}_i^T \mathbf{F} \mathbf{u}_i = 0, \tag{32}$$

for all i. $\mathbf{F}$ is of rank 2. Letting $\mathbf{u} = (u, v, 1)^T$ and $\mathbf{u'} = (u', v', 1)$, (32) can be rewritten into

$$\mathbf{U}_i^T \mathbf{h} = 0, \tag{33}$$

where

$$\mathbf{U} = (uu', uv', u, vu', vv'v, u', v', 1)^T,$$
$$\mathbf{v} = (f_{11}, f_{21}, f_{31}, f_{12}, f_{22}, f_{32}, f_{13}, f_{23}, f_{33})^T.$$

Noting that (33) is a simple plane equation, if we can detect and handle noise and outliers in the feature space, (33) should enable us to produce a good estimation. Finally, we apply [12] to obtain a rank-2 fundamental matrix. Data normalization is done similarly as in [12] before the optimization.

We evaluate the results by estimating the fundamental matrix of the data set *Corridor*, which is available at www.robots.ox.ac.uk/~vgg/data.html. The matches of feature points (Harris corners) are available. Random outliers were added in the feature space.

Fig. 11 shows the plot of RMS error, which is computed by summing up and averaging $\sqrt{\frac{1}{p}\sum_i \|\mathbf{U}_i^T \hat{\mathbf{h}}\|^2}$ over all pairs, where $\mathbf{U}_i$ is the set of clean data and $\hat{\mathbf{h}}$ is the 9D vector produced from the rank-2 fundamental matrices estimated by various methods. Note that all the images available in the *Corridor* data set are used, that is, all C_2^{11} pairs were tested. It can be observed that RANSAC breaks down at an OI ratio $\simeq 20$, or 95.23 percent outliers. ASSC is very stable with RMS error < 0.15. TV breaks down at an OI ratio $\simeq 10$. EMTV has negligible RMS error before it starts to break down at an OI ratio $\simeq 40$. This finding echoes that of Hartley [12] that linear solution is sufficient when outliers are properly handled.

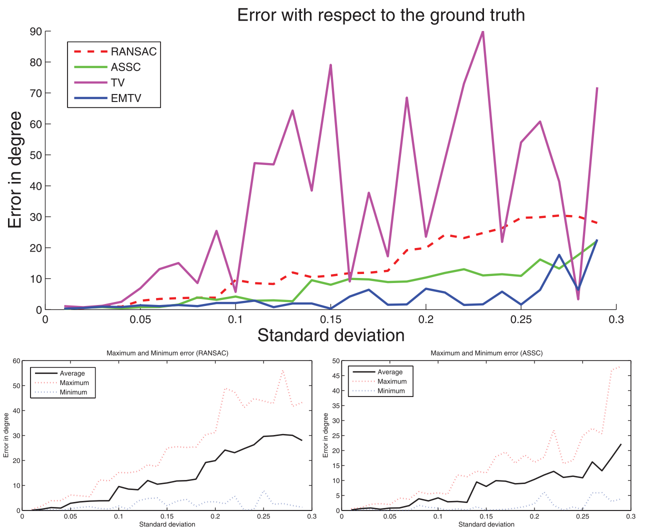


Fig. 10. Measurement error: Standard deviation varies from 0.01 to 0.29 with OI ratio at 10.

7.3 Matching

In the uncalibrated scenario, EMTV estimates parameter accurately by employing CFTV, and effectively discards epipolar geometries induced by wrong matches. Typically, a camera calibration is performed using nonlinear least-squares minimization and bundle adjustment [16] which requires good matches as input. In this experiment, candidate matches are generated by comparing the resulting 128D SIFT feature vectors [17], so many 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 (see (26)) of a keypoint pair i being an inlier. In the experiment, we assume keypoint pair i is an inlier if $w_i > 0.8$. Fig. 12 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 results produced with `KeyMatchFull` [25], `linear_match`, `assc_match` [28], and `ransac_match`. Note in particular that only `emtv_match` recovers the overall geometry of the teapot, whereas the other methods can only recover one side of the teapot.

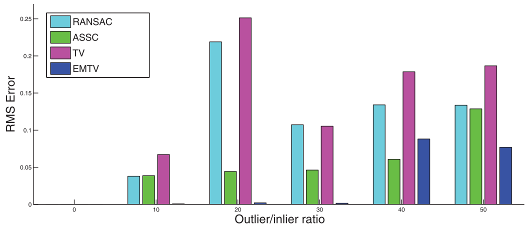


Fig. 11. *Corridor*. RMS error plot of various methods.

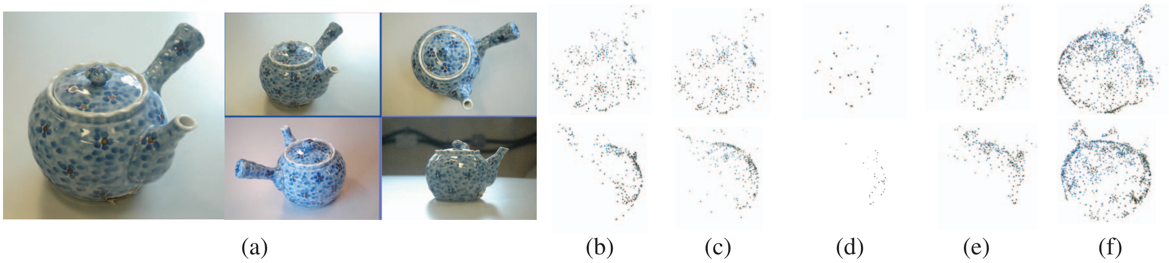


Fig. 12. *Teapot*: (a) Four images (one in enlarged view) from the input image set consisting of 30 images captured around the object in a casual manner. (b)-(f) show two views of the sparse reconstruction generated by using KeyMatchFull (398 points), linear_match (493 points), ransac_match (37 points), assc_match (208 points), and emtv_match (2,152 points). The candidate matches returned by SIFT are extremely noisy due to the ambiguous patchy patterns. On average, 17,404 trials were run in ransac_match. It is time consuming to run more trials on this noisy and large input where an image pair can have as many as 5,000 similar matches and similarly for assc_match, where additional running time is needed to estimate the scale parameter in each iteration. On the other hand, emtv_match does not require any random sampling.

This example is challenging because the teapot's shape is quite symmetric and the patchy patterns look identical everywhere. As was done in [25], each photo was paired, respectively, with a number of photos with camera poses satisfying certain basic criteria conducive to matching or making the numerical process stable (e.g., wide-baseline stereo). We can regard this pair-up process as one of computing connected components. If the fundamental matrix between any successive images is incorrectly estimated, the corresponding components will no longer be connected, resulting in the situation where only one side or part of the object can be recovered.

Since KeyMatchFull and linear_match use simple distance measure for finding matches, the coverage of the corresponding connected components tends to be small. It is interesting to note that the worst result is produced by using ransac_match. This can be attributed to three reasons: 1) The fundamental matrix is of rank 2, which implies that $\mathbf{h}$ spans a subspace $\leq$ 8D rather than a 9D hyperplane; 2) the input matches contain too many outliers for some image pairs; 3) it is not feasible to fine-tune the scale parameter for every possible image pair and so we used a single value for all of the images. A slight improvement could be found from ASSC. However, it still suffers from problems (1) and (2) and so the result is not very good even compared with KeyMatchFull and linear_match.

On the other hand, emtv_match utilizes the epipolar geometry constraint by computing the fundamental matrix in a data driven manner. Since the outliers are effectively filtered out, the estimated fundamental matrices are sufficiently accurate to pair up all of the images into a single connected component. Thus, the overall 3D geometry can be recovered from all the available views.

7.4 Multiview Stereo Reconstruction

This section outlines how CFTV and EMTV are applied to improve the match-propagate-filter pipeline in multiview stereo. 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 [10] has produced some of the best results to date.

We observe that PMVS had not fully utilized 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 [10], data communication should not be done by smoothing, but the lack of communication will cause perturbed surface normals and patch outliers during the propagation stage. In [29], we proposed tensor-based multiview stereo (TMVS) and used 3D structure-aware tensors which communicate among each other via CFTV. 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.

We captured 179 photos around a building which were first calibrated as described in Section 7.3. 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-baseline stereo. TMVS was run to obtain the quasi-dense reconstruction, where MRFTV was used to filter outliers as shown in Fig. 13. Fig. 14 shows the 3D reconstruction, which is faithful to the real building. Readers are referred to [29] for more detail and experimental evaluation of TMVS.

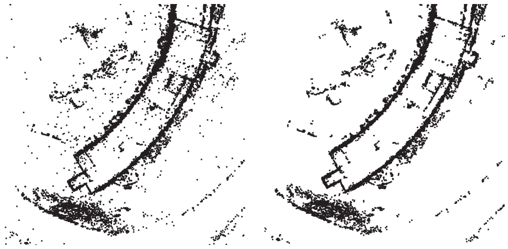


Fig. 13. Results before and after filtering of *Hall 3* (images shown in Fig. 14). All salient 3D structures are retained in the filtered result, including the bushes near the left facade and planters near the right facade in this top view of the building.

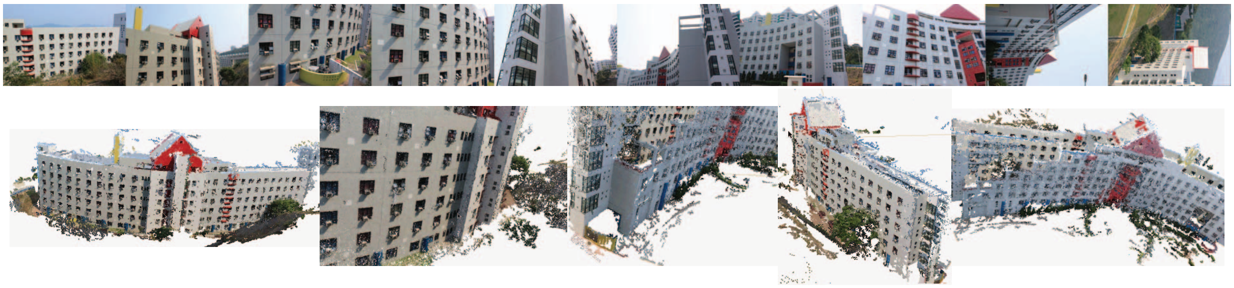


Fig. 14. The *Hall 3* reconstruction: 10 of the input images (top) and five views of the quasi dense 3D reconstruction (bottom).

8 Conclusions

A closed-form solution is proven for the special theory of tensor voting (CFTV) for computing an exact structure-aware tensor in any dimensions. For structure propagation, we derive a quadratic energy for MRFTV, thus providing a convergence proof for tensor voting which is impossible to prove using the original tensor voting procedure. Then, we derive EMTV for optimizing both the tensor and model parameters for robust parameter estimation. We performed quantitative and qualitative evaluation using challenging synthetic and real data sets. In the future, we will develop a closed-form solution for the general theory of tensor voting, and extend EMTV to extract multiple and nonlinear structures. We have provided C++ source code, but it is straightforward to implement (11), (12), (26), (30), (18), and (19). We demonstrated promising results in multiview stereo, and will apply our closed-form solution to address important computer vision problems.

Acknowledgments

The authors would like to thank the Associate Editor and all of the anonymous reviewers. Special thanks to Reviewer 1 for his/her helpful and detailed comments throughout the review cycle. S.-K. Yeung was supported by the Start-up Research Grant: SRG ISTD 2011 016. J. Jia was supported by the Hong Kong Research Grant Council (RGC, grant no. 412911). C.-K. Tang was supported by RGC (grant no. 619711) and the Google Faculty Award on quasi dense 3D reconstruction.

References

[1] S. Arya and D.M. Mount, "Approximate Nearest Neighbor Searching," *Proc. ACM-SIAM Symp. Discrete Algorithms,* pp. 271-280, 2003.

[2] J. Bilmes, "A Gentle Tutorial on the EM Algorithm and Its Application to Parameter Estimation for Gaussian Mixture and Hidden Markov Models," Technical Report ICSI-TR-97-021, ICSI, 1997.

[3] H. Chen and P. Meer, "Robust Regression with Projection Based m-Estimators," *Proc. Ninth IEEE Int'l Conf. Computer Vision,* vol. 2, pp. 878-885, 2003.

[4] O. Chum and J. Matas, "Optimal Randomized Ransac," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 30, no. 8, pp. 1472-1482, Aug. 2008.

[5] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach toward Feature Space Analysis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 24, no. 5, pp. 603-619, May 2002.

[6] R. Dahyot, "Statistical Hough Transform," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 31, no. 8, pp. 1502-1509, Aug. 2009.

[7] M.A. Fischler and R.C. Bolles, "Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography," *Comm. ACM,* vol. 24, pp. 381-395, 1981.

[8] D. Forsyth and J. Ponce, *Computer Vision: A Modern Approach.* Prentice Hall, 2003.

[9] E. Franken, M. van Almsick, P. Rongen, L. Florack, and B. ter Haar Romeny, "An Efficient Method for Tensor Voting Using Steerable Filters," *Proc. Ninth European Conf. Computer Vision,* vol. 4, pp. 228-240, 2006.

[10] Y. Furukawa and J. Ponce, "Accurate, Dense, and Robust Multiview Stereopsis," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 32, no. 8, pp. 1362-1376, Aug. 2010.

[11] B. Georgescu, I. Shimshoni, and P. Meer, "Mean Shift Based Clustering in High Dimensions: A Texture Classification Example," *Proc. Ninth IEEE Int'l Conf. Computer Vision,* pp. 456-463, 2003.

[12] R. Hartley, "In Defense of the Eight-Point Algorithm," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 19, no. 6, pp. 580-593, June 1997.

[13] P. Hough, "Machine Analysis of Bubble Chamber Pictures," *Proc. Int'l Conf. High Energy Accelerators and Instrumentation,* 1959.

[14] P.J. Huber, *Robust Statistics.* John Wiley & Sons, 1981.

[15] K. Lee, P. Meer, and R. Park, "Robust Adaptive Segmentation of Range Images," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 20, no. 2, pp. 200-205, Feb. 1998.

[16] M. Lourakis and A. Argyros, "The Design and Implementation of a Generic Sparse Bundle Adjustment Software Package Based on the Levenberg-Marquardt Algorithm," Technical Report 340, ICSFORTH, 2004.

[17] D. Lowe, "Distinctive Image Features from Scale-Invariant Keypoints," *Int'l J. Computer Vision,* vol. 60, no. 2, pp. 91-110, Nov. 2004.

[18] G.J. McLachlan and T. Krishnan, *EM Algorithms and Extension.* Elsevier, 1997.

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

[20] P. Meer, "Robust Techniques for Computer Vision," *Emerging Topics in Computer Vision,* chapter 4, Prentice Hall, 2004.

[21] P. Mordohai and G. Medioni, "Dimensionality Estimation, Manifold Learning and Function Approximation Using Tensor Voting," *J. Machine Learning Research,* vol. 11, pp. 411-450, Jan. 2010.

[22] P. Rousseeuw, "Least Median of Squares Regression," *J. Am. Statistics Assoc.,* vol. 79, pp. 871-880, 1984.

[23] P. Rousseeuw, *Robust Regression and Outlier Detection.* Wiley, 1987.

[24] K. Sim and R. Hartley, "Removing Outliers Using the Linfty Norm," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* vol. 1, pp. 485-494, 2006.

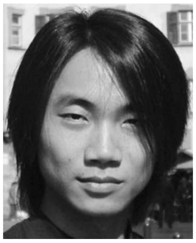
[25] N. Snavely, S. Seitz, and R. Szeliski, "Modeling the World from Internet Photo Collections," *Int'l J. Computer Vision,* vol. 80, no. 2, pp. 189-210, Nov. 2008.

[26] R. Subarao and P. Meer, "Beyond Ransac: User Independent Robust Regression," *Proc. Workshop 25 Years of Random Sample Consensus,* June 2006.

[27] W.S. Tong, C.K. Tang, and G. Medioni, "Simultaneous Two-View Epipolar Geometry Estimation and Motion Segmentation by 4D Tensor Voting," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 9, pp. 1167-1184, Sept. 2004.

[28] H. Wang and D. Suter, "Robust Adaptive-Scale Parametric Model Estimation for Computer Vision," *IEEE Trans. Pattern Analysis and Machine Intelligence,* vol. 26, no. 11, pp. 1459-1474, Nov. 2004.

[29] T.P. Wu, S.K. Yeung, J. Jia, and C.K. Tang, "Quasi-Dense 3D Reconstruction Using Tensor-Based Multiview Stereo," *Proc. IEEE Conf. Computer Vision and Pattern Recognition,* 2010.



Tai-Pang Wu received the PhD degree in computer science from the Hong Kong University of Science and Technology (HKUST) in 2007. He was awarded the Microsoft Fellowship in 2006. He has been employed as a postdoctoral fellow at Microsoft Research Asia and the Chinese University of Hong Kong. He is currently a manager of the Enterprise and Consumer Electronics Group at the Hong Kong Applied Science and Technology Research Institute (ASTRI). He is a member of the IEEE Computer Society.

Sai-Kit Yeung received the BEng degree (First Class Honors) in computer engineering and the MPhil degree in bioengineering from the Hong Kong University of Science and Technology (HKUST) in 2003 and 2005, respectively, and the PhD degree in electronic and computer engineering from HKUST in 2009. He was a visiting student at the Image Sciences Institute, University Medical Center Utrecht, The Netherlands, in 2007 and the Image Processing Research Group at the University of California Los Angeles (UCLA) in 2008. He is currently an assistant professor at the Singapore University of Technology and Design (SUTD). Prior to joining SUTD, he was a postdoctoral scholar in the Department of Mathematics, UCLA in 2010. His research interests include computer vision, computer graphics, computational photography, image/video processing, and medical imaging. He is a member of the IEEE Computer Society.

Jiaya Jia received the PhD degree in computer science from the Hong Kong University of Science and Technology in 2004 and is currently an associate professor in the Department of Computer Science and Engineering at the Chinese University of Hong Kong (CUHK). He was a visiting scholar at Microsoft Research Asia from March 2004 to August 2005 and conducted collaborative research at Adobe Systems in 2007. He leads the research group at CUHK, focusing on computational photography, 3D reconstruction, practical optimization, and motion estimation. He serves as an associate editor for the *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* and as an area chair for ICCV 2011. He was on the program committees of several major conferences, including ICCV, ECCV, and CVPR, and cochaired the Workshop on Interactive Computer Vision in conjunction with ICCV 2007. He received the Young Researcher Award 2008 and Research Excellence Award 2009 from CUHK. He is a senior member of the IEEE and a member of the IEEE Computer Society.

Chi-Keung Tang received the MSc and PhD degrees in computer science from the University of Southern California (USC), Los Angeles, in 1999 and 2000, respectively. Since 2000, he has been with the Department of Computer Science at the Hong Kong University of Science and Technology (HKUST), where he is currently a professor. He is an adjunct researcher in the Visual Computing Group of Microsoft Research Asia. His research areas are computer vision, computer graphics, and human-computer interaction. He is an associate editor of the *IEEE Transactions on Pattern Analysis and Machine Intelligence (TPAMI)* and on the editorial board of the *International Journal of Computer Vision (IJCV)*. He served as an area chair for ACCV 2006 (Hyderabad), ICCV 2007 (Rio de Janeiro), ICCV 2009 (Kyoto), ICCV 2011 (Barcelona), and as a technical papers committee member for the inaugural SIGGRAPH Asia 2008 (Singapore), SIGGRAPH 2011 (Vancouver), SIGGRAPH Asia 2011 (Hong Kong), and SIGGRAPH 2012 (Los Angeles). He is a senior member of the IEEE and a member of the IEEE Computer Society.

Gérard Medioni received the Diplôme d'Ingénieur from the École Nationale Supérieure des Télécommunications (ENST), Paris, in 1977 and the MS and PhD degrees from the University of Southern California (USC) in 1980 and 1983, respectively. He has been with USC since then and is currently a professor of computer science and electrical engineering, a codirector of the Institute for Robotics and Intelligent Systems (IRIS), and a codirector of the USC Games Institute. He served as the chairman of the Computer Science Department from 2001 to 2007. He has made significant contributions to the field of computer vision. His research covers a broad spectrum of the field, such as edge detection, stereo and motion analysis, shape inference and description, and system integration. He has published three books, more than 50 journal papers, and 150 conference articles. He is the holder of eight international patents. He is an associate editor of the *Image and Vision Computing Journal*, *Pattern Recognition and Image Analysis Journal*, and the *International Journal of Image and Video Processing*. He served as a program cochair of the 1991 IEEE Computer Vision and Pattern Recognition (CVPR) Conference and the 1995 IEEE International Symposium on Computer Vision, a general cochair of the 1997 IEEE CVPR Conference, a conference cochair of the 1998 International Conference on Pattern Recognition, a general cochair of the 2001 IEEE CVPR Conference, a general cochair of the 2007 IEEE CVPR Conference, and a general cochair of the 2009 IEEE CVPR Conference. He is a fellow of the IEEE, IAPR, and AAAI and a member of the IEEE Computer Society.