Probabilistic Tensor Voting for Robust Perceptual Grouping

Dian Gong and Gérard Medioni
Institute for Robotics and Intelligent Systems, University of Southern California
Los Angeles, CA, 90089
{diangong|medioni}@usc.edu

Abstract

We address the problem of unsupervised segmentation and grouping in 2D and 3D space, where samples are corrupted by noise, and in the presence of outliers. The problem has attracted attention in previous research work, but non-parametric outlier filtering and inlier denoising are still challenging. Tensor voting is a non-parametric algorithm that can infer local data geometric structure. Standard tensor voting considers outlier noise explicitly, but may suffer from serious problems if the inlier data is also noisy. In this paper, we propose probabilistic Tensor Voting, a Bayesian extension of standard tensor voting, taking into consideration both probabilistic and geometric meaning. Probabilistic tensor voting explicitly considers both outlier and inlier noise, and can handle them simultaneously. In the new framework, the representation consists of a 2nd order symmetric tensor, a polarity vector, and a new type 2 polarity vector orthogonal to the first one. We give a theoretical interpretation of our framework. Experimental results show that our approach outperforms other methods, including standard tensor voting.

1. Introduction

A fundamental problem for low-level computer vision is how to process noisy data and extract meaningful features. This has applications in denoising, contour grouping, stereo matching and object detection. In general, there are two types of noise: outliers that are independent of the meaningful data, and inlier noise which refers to errors on the meaningful data. The left part of Fig. 1 gives an conceptual example; blue points that are far away from the underlying manifold (black) can be viewed as outliers, while red points close to the manifold can be viewed as samples with approximated Gaussian noise. Because (1) the underlying manifold is nonparametric (2) both inlier and outlier noise exist, the geometric structure inference becomes a challenging problem. The right part of Fig. 1 is a real example of contour detection on natural images. The results of the state-of-the-art boundary detector [8] still have many problems such as fragments, noisy false detection, which requires further robust grouping.

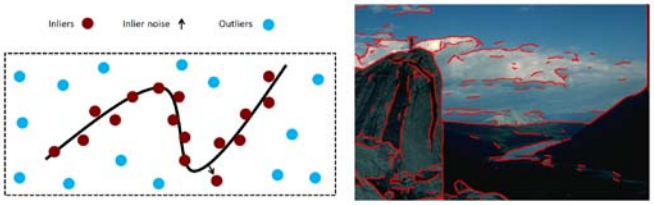

Figure 1. **Problem illustration.** Left: an example of inlier and outlier noise, right: a natural image and the boundary detection results by [8].

Tensor voting [12] is a geometric computational framework which has been successfully applied to many problems in computer vision [11, 20] and machine learning [13, 3]. In the tensor voting framework, local information is represented by a second-order tensor and a first order polarity vector. Tensor voting is quite robust with only one free parameter and can handle large amount of outliers. However, there are two potential problems, which are not solved by the standard framework. First, the standard tensor voting algorithm does not take into account inlier noise. Given two points lying on a manifold with errors, tensor information propagation from one point to the other is actually incorrect. The problem could be more serious when the data sampling density on the manifold is low, which means the error can not be mitigated by the law of large numbers. Second, when voter and receiver points are very close to each other, small numerical errors lead to large bias.

In this paper, Probabilistic Tensor Voting (PTV) is proposed to handle outliers and inlier noise simultaneously. The main idea is to apply the bias-variance tradeoff from machine learning to perceptual grouping by incorporating a Bayesian framework. In particular, we follow the original Tensor Voting approach, but change the voter's position from a fixed vector to a random vector with probabilistic distribution. The final voting results are the expectations under the probability measure. This process generates new de-

978-1-4673-1612-5/12/$31.00 ©2012 IEEE

terministic voting fields. These probabilistic votes are theoretically analyzed in section 4. In summary, the new voting method incorporates both a *geometric algorithm*, which is from standard tensor voting, and a *probabilistic inference*, which is used to handle inlier noise.

Contributions:

(1) By combining a Bayesian framework with the geometric structure inference algorithm, we gain the capability to reduce the learning variance for perceptual grouping.

(2) Our novel voting framework with one $2nd$ order tensor and two types of polarity vectors can represent and handle both outlier and inlier noise at the same time.

(3) As in standard tensor voting, our algorithm can process all types of geometric structures in a non-parametric way, including multi-manifold, manifold junctions, manifold endpoints, etc. Note that the computational cost of the new approach remains the same.

2. Related Work

We start with a review of the works along three axes. They are related work to tensor voting, outlier identification and non-parametric data denoising.

Tensor voting [12] is a general computational framework that can be applied to many areas in computer vision and machine learning. It includes figure completion [11], 3D mesh extraction, motion pattern analysis [20], dimension estimation, manifold learning [13], etc. Detailed comparisons with other methods in any one of these specific areas can be found in previous tensor voting papers.

For outliers identification, there is a large class of methods, called stochastic sample consensus. The representative work among them is the Random Sample Consensus, RANSAC [2]. The objective function to be maximized is the number of estimated inlier points, which are the data points lying within a given threshold of the estimated model. Related works include MLESAC, MAPSAC and M-estimator [4]. Recently, StaRSac is proposed to analyze the variance of the parameters (VoP), and it shows that there is a stable region of the estimated parameters which can solve the unknown uncertainty problem [1]. The main limitation of these methods is the need for specific parametric model of inlier data. Another type of popular work is voting based methods. In Hough transform [9], the optimal model parameters are selected as the parameters space cell which gets the maximum number of votes. Hough transform also requires a parametric model of inlier data, so it can't be applied to non-parametric case. Note that tensor voting can handle large amount of outliers without parametric model constraint, but the inlier noise is not taken into account in the original framework.

For inlier data denoising, many parametric model based denoising methods have been proposed, such as linear and nonlinear regression. Here we only focus on the non-parametric data denoising. Principal Component Analysis (PCA) is one of the most successful denoising methods that has been applied to many aspects of computer vision [6]. While PCA is a linear subspace method, Kernel PCA is an extension by kernelizing the Gram matrix [10]. Robust PCA is a marriage between PCA and robust estimator to handle outliers [7], and the main limitation is the linearity constraint of the inlier data model. Also it calculates covariance matrix iteratively, which leads to large computational cost.

To handle non-parametric estimation, a manifold model is assumed to fit the data, which means there is a latent intrinsic structure embedded in the high dimensional space. Diffusion map method views denoising as reversing a diffusion process of which the normalized graph Laplacian matrix is the generator [5]. This approach tends to overly smooth the inputs and push data points to the mean curvature of the manifold. Moreover, outliers are not considered in the framework, and outliers are greatly harmful to the diffusion process. Our probabilistic tensor voting framework performs non-parametric estimation in the presence of both inlier and outlier noise.

3. Tensor Voting Review

Tensor Voting was originally developed in $2D$ for perceptual grouping, and later extended to $3D$ and ND space. Suppose we have a set of points $\boldsymbol{x_i}$ $(i = 1, 2, ...N)$ in 2D space.[1] Our objective is to infer the local geometric structure of $\boldsymbol{x_i}$, and use this to characterize $\boldsymbol{x_i}$. We use the structure tensor $\boldsymbol{T}$ and polarity $\boldsymbol{p}$ to describe the local information at point $\boldsymbol{x} \in R^2$. Here, $\boldsymbol{T}$ is a 2 by 2 symmetric and nonnegative definite matrix, and p is a 2 by 1 vector.

$$\begin{aligned} &\boldsymbol{T} = \lambda_1 \boldsymbol{e_1} \boldsymbol{e_1}^T + \lambda_2 \boldsymbol{e_2} \boldsymbol{e_2}^T = \\ &(\lambda_1 - \lambda_2)(\boldsymbol{e_1} \boldsymbol{e_1}^T) + \lambda_2 (\boldsymbol{e_1} \boldsymbol{e_1}^T + \boldsymbol{e_2} \boldsymbol{e_2}^T) \end{aligned} \tag{1}$$

The eigenvector $\boldsymbol{e_1}$ corresponding to the largest eigenvalue λ_1 represents the normal space of the local manifold, and eigenvector $\boldsymbol{e_2}$ represents the tangent space. The local manifold saliency is $\lambda_1 - \lambda_2$, indicating how clear the local structure is. Polarity vector p is used to detect endpoints which appear clearly as the local maxima of polarity $||\boldsymbol{p}||$. If the eigenvalues are both quite small, the point is classified as an outlier, and as a junction if both are large and almost equivalent. There is only one free parameter in the voting algorithm, which is the voting scale σ_v that controls the saliency decay. Details can be found in [12].

4. Probabilistic Tensor Voting

There are two distinguished features of probabilistic tensor voting. First, in the voting procedure, the voter's posi-

[1] Discussions are in 2D, experiments include 3D by naturally extending our algorithm.

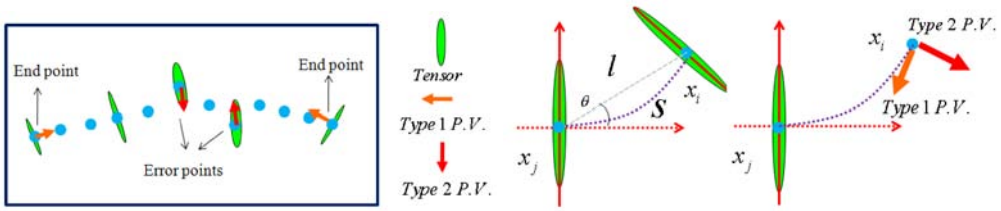

Figure 2. **Illustration and Visualization of Tensor Voting.** Left, the functions of the second order and two types of first order tensors; right, the second order stick vote and the first order stick vote with two polarity vectors orthogonal to each other

tion becomes a ***random vector*** with a given standard deviation. Second, in the representation part, a second type of polarity vector $\boldsymbol{p2}$ is introduced. As a result, inlier noise can be explicitly handled by the new framework.

4.1. Representation

The left part of Fig. 2 is an illustrative example, which shows how to use the second order tensor and two types of polarity vectors to represent the local geometric information and classify points.

Second order tensor: the function of the second order tensor $\boldsymbol{T}$ is almost the same as the standard voting, except that λ_2 represents the uncertainty of the local information. When λ_2 is relatively large, it can be inferred either the inlier point or its tensor suffers from noise, caused by error information propagation from the neighborhood.

Type 1 **polarity vector**: it is the same as in the standard voting, which uses $\boldsymbol{p1}$ to detect endpoints [12]. As in Fig. 2, endpoints can not be detected by using the second order tensor, but it can be detected by the local maximum of polarity $||\boldsymbol{p1}||$ in the tangent space $\boldsymbol{e_2}$. More generally, $\boldsymbol{p1}$ reflects the local data sampling asymmetry on the manifold. For example, if most of the neighborhood points of $\boldsymbol{x_j}$ lie on one side of its normal direction, then the polarity vector will point to this side. The extreme example is endpoints, which have the locally largest polarity $||\boldsymbol{p1}||$.

Type 2 **polarity vector**: this is defined as the *orthogonal vector* (pointing to the local tangent space of the voter) of $\boldsymbol{p}_1$, as shown in Fig. 2. Error points are detected by the local maximum type 2 polarity, and the error vector is given by the combination of $\boldsymbol{e_1}$ and $\boldsymbol{p_2}$. The error vector gives the direction from the point to the manifold. More generally, the polarity $||\boldsymbol{p2}||$ and eigenvalue λ_2 reflect the uncertainty of the estimated normal (tangent) space, which is related to the error amount at this point. Furthermore, $\boldsymbol{p2}$ is helpful for junction localization, which we do not present due to lack of space. $\boldsymbol{p2}$ is related to the previous work [16], which estimates curvature.

4.2. Sparse Voting Procedure

Probabilistic tensor voting is a progressive algorithm, which includes three layers, sparse ball vote, sparse stick vote, and revised stick vote (it can have a few more layers for specific tasks). These progressive voting steps are the same as in the standard tensor voting [13]. The key idea is the position of each token $\boldsymbol{x_j}$ is changed from a deterministic vector to a ***random vector*** in R^2.

Ball Vote: there is no information of the tensor at each point $\boldsymbol{x_j}$. When token $\boldsymbol{x_j}$ votes to token $\boldsymbol{x_i}$, an isotropic *p.d.f* $p(\boldsymbol{x})$ is assumed for the voter's position $\boldsymbol{x_j}$. Under certain assumptions, the uncertainty distribution of the voter's position is in a $1D$ space, which reduces computational cost and avoids some numerical problems. The detailed derivation is given in section 4.1. Ball vote tensor at $\boldsymbol{x_i}$ can be calculated as follows,

$$\boldsymbol{B}(\boldsymbol{x_i}, \sigma_v, \sigma_n) = \sum_{j \neq i} [\int_{O_{ji}} \frac{1}{(2\pi)^{\frac{1}{2}} \sigma_n} e^{-\frac{\|\boldsymbol{x}-\boldsymbol{x_j}\|^2}{2\sigma_n^2}} \boldsymbol{T}(\boldsymbol{x}, \boldsymbol{x_i}, \sigma_v) d\boldsymbol{x}] \tag{2}$$

Here, $p(\boldsymbol{x})$ is assumed to be a $1D$ Gaussian distribution with mean $\boldsymbol{x_j}$ and variance σ_n^2 in the orthogonal space O_{ji}, which is a $1D$ line orthogonal to the straight line $\boldsymbol{x_i x_j}$. σ_v is the voting scale and σ_n is the noise voting scale. The function $T(\boldsymbol{x}, \boldsymbol{x_i}, \sigma_v)$ is standard ball vote with an analytic formulation as in [12],

$$\boldsymbol{T}(\boldsymbol{x}, \boldsymbol{x_i}, \sigma_v) = e^{-\frac{\|\boldsymbol{x}-\boldsymbol{x_i}\|^2}{\sigma_v^2}} (\boldsymbol{I} - \frac{(\boldsymbol{x}-\boldsymbol{x_i})(\boldsymbol{x}-\boldsymbol{x_i})^T}{\|\boldsymbol{x}-\boldsymbol{x_i}\|^2}) \tag{3}$$

Based on the theoretical analysis in the supplementary material, $\boldsymbol{B}(\boldsymbol{x_i}, \sigma_v, \sigma_n)$ actually has a straightforward formulation as follows,

$$\sum_{j \neq i} [\mu(||\boldsymbol{x_j}-\boldsymbol{x_i}||, \sigma_v, \sigma_n)(\frac{(\boldsymbol{x_j}-\boldsymbol{x_i})(\boldsymbol{x_j}-\boldsymbol{x_i})^T}{||\boldsymbol{x_j}-\boldsymbol{x_i}||^2}) + \lambda(||\boldsymbol{x_j}-\boldsymbol{x_i}||, \sigma_v, \sigma_n)(\boldsymbol{I} - \frac{(\boldsymbol{x_j}-\boldsymbol{x_i})(\boldsymbol{x_j}-\boldsymbol{x_i})^T}{||\boldsymbol{x}-\boldsymbol{x_i}||^2})] \tag{4}$$

$\lambda(.)$ is the kernel function for the normal direction propagation (vote saliency), and $\mu(.)$ is the kernel function for the uncertainty propagation (junction saliency). These two

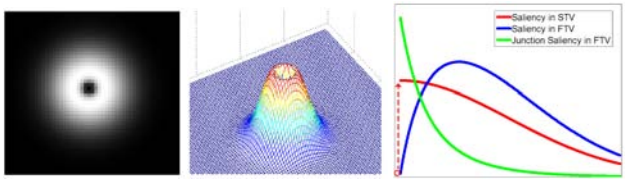

Figure 3. **Visualization and Comparisons of Ball Vote.** Left and middle, 2D ball vote saliency map; right, STV and PTV for ball vote saliency and junction saliency. The horizontal and vertical axes represent distance and saliency value respectively

kernels have analytic forms but involve complicated modified Bessel functions. In a degenerate case, $\sigma_n = 0$, $\lambda(.)$ becomes a Gaussian kernel and $\mu(.)$ becomes a 0 constant kernel, which is the same as eq. 3.

The $2D$ new ball field is visualized in Fig. 3. The voting saliency is a *mountain shaped* function of the voting distance s. When s small, the saliency is also small. More importantly, the new saliency function is *smooth* everywhere, while the standard one has a δ jump at $s = 0$.

Stick Vote: the right part of Fig. 2 is an illustration of standard stick vote. There is a tensor $\boldsymbol{T_j}$ at each point $\boldsymbol{x_j}$ representing the local normal space. So, a 1D *p.d.f* $p_j(\boldsymbol{x})$ is enforced to $\boldsymbol{x_j}$, and the stick vote at $\boldsymbol{x_i}$ can be calculated as follows,

$$\boldsymbol{S}(\boldsymbol{x_i}, \sigma_v, \sigma_n) = \sum_{j \neq i} [\int_{N_j} \frac{1}{(2\pi)\sigma_n^2} e^{-\frac{\|\boldsymbol{x}-\boldsymbol{x_j}\|^2}{2\sigma_n^2}} \boldsymbol{T}(\boldsymbol{x}, \boldsymbol{x_i}, \boldsymbol{T_j}, \sigma_v) d\boldsymbol{x}] \quad (5)$$

Here N_j is the $1d$ normal space of $\boldsymbol{x_j}$, spanned by $\boldsymbol{e_{j_1}}$. $p_j(\boldsymbol{x})$ is a $1d$ Gaussian distribution with mean $\boldsymbol{x_j}$ and variance σ_n^2 in N_j. The function $T(\boldsymbol{x}, \boldsymbol{x_i}, \boldsymbol{T_j}, \sigma_v)$ is the standard stick vote [12] with the analytic formulation as,

$$\boldsymbol{T}(\boldsymbol{x}, \boldsymbol{x_i}, \boldsymbol{T_j}, \sigma_v) = e^{-\frac{s^2+ck^2}{\sigma_v^2}} \boldsymbol{R}(2\theta_{ji}) \boldsymbol{T_j} \boldsymbol{R}(2\theta_{ji})^T \quad (6)$$

θ_{ji} is the angle between the tangent direction $\boldsymbol{e_{j_2}}$ and vector $\boldsymbol{x_i x_j}$. Arc length $s = \frac{\theta_{ji} l}{sin(\theta_{ji})}$, curvature $k = \frac{2sin(\theta_{ji})}{l}$ and $l = ||\boldsymbol{x} - \boldsymbol{x_j}||$. $\boldsymbol{R}(2\theta_{ji})$ is the standard counter-clockwise rotation matrix with angle $2\theta_{ji}$. c is a constant defined in [12].

The intuition is that uncertainty can be locally decomposed into two directions, tangent and normal. The tangent direction represents the sampling uncertainty on the manifold, whereas the normal direction represents the uncertainty away from the manifold, indicating the inlier error in our probabilistic voting framework.

The stick field is visualized in Fig. 4. Compared with the standard field, the new field decreases more slowly as curvature increases, because of the uncertainty of the voter's

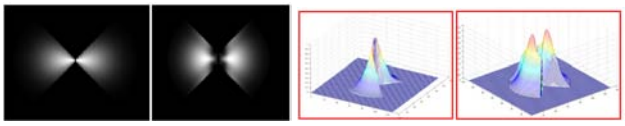

Figure 4. **Comparisons of Stick Vote.** Left, saliency comparison of the standard voting (left) and the new voting(right); right, 3d saliency map comparison of the standard voting (left) and the new voting (right)

position in the normal space. In the tangent direction, the saliency function is monotonically decreasing with the distance in the standard field, while it is a *mountain shaped* function in the new field.

Similar to the second order tensor, the polarity vector is the expectation of the random vote,

$$\boldsymbol{p1}(\boldsymbol{x_i}, \sigma_v, \sigma_n) = \sum_{j \neq i} [\int_{N_j} \frac{1}{(2\pi)\sigma_n^2} e^{-\frac{\|\boldsymbol{x}-\boldsymbol{x_j}\|^2}{2\sigma_n^2}} \boldsymbol{P1}(\boldsymbol{x}, \boldsymbol{x_i}, \boldsymbol{T_j}, \sigma_v) d\boldsymbol{x}] \quad (7)$$

$$\boldsymbol{p2}(\boldsymbol{x_i}, \sigma_v, \sigma_n) = \sum_{j \neq i} [\int_{N_j} \frac{1}{(2\pi)\sigma_n^2} e^{-\frac{\|\boldsymbol{x}-\boldsymbol{x_j}\|^2}{2\sigma_n^2}} \boldsymbol{P2}(\boldsymbol{x}, \boldsymbol{x_i}, \boldsymbol{T_j}, \sigma_v) d\boldsymbol{x}] \quad (8)$$

$\boldsymbol{p1}$ is the original polarity vector in the standard voting algorithm and $\boldsymbol{p2}$ is the type 2 polarity vector orthogonal to $\boldsymbol{p1}$ in each deterministic vote.

Revised Stick Vote: if we want to further reduce errors, a third pass of revised stick vote can be performed. The voting algorithm is the same as a stick vote, while the center of the distribution is moved from the voter's position to the error direction, which is provided by $\boldsymbol{p2}$.

Analysis: there are two parameters in the voting algorithm, the standard voting scale σ_v and the newly added noise voting scale σ_n, which indicates the variances of the voter's distribution. Both parameters are robust, and this is supported by empirical study in later sections. It is notable that both the ball and stick fields are identical to the original fields if $\sigma_n = 0$.

In the following part, we analyze the ball voting algorithm briefly and compare it to related approaches. Eq. 3 and eq. 4 focus on the normal representation, and the dual formulation focusing on the tangent representation can be obtained easily by omitting the identity matrix. Then, we can recognize they are special cases of the local linear tensor (LLT) model with different kernel functions (omitting the junction saliency part in probabilistic ball vote). There is another type of popular LLT model, Local Principal Component Analysis (LPCA), which serves as the foundation for

many nonlinear latent variable modeling works,

$$\boldsymbol{T}(\boldsymbol{x_i}, \sigma_v) = \sum_{\boldsymbol{x_j} \in N(\boldsymbol{x_i})} \| \boldsymbol{x_j} - \boldsymbol{x_i} \|^2 \left(\frac{(\boldsymbol{x_j} - \boldsymbol{x_i})(\boldsymbol{x_j} - \boldsymbol{x_i})^T}{\| \boldsymbol{x_j} - \boldsymbol{x_i} \|^2}\right) \tag{9}$$

$N(\boldsymbol{x_i})$ is the set of neighborhood points of $\boldsymbol{x_i}$ (K-NN or h-neighborhood). It is notable that we use $\boldsymbol{x_i}$ to approximate $\boldsymbol{x_m}$, which is the local mean estimation of $N(\boldsymbol{x_i})$. Compared with the quadric kernel (LPCA), Gaussian kernel (STV) is more suitable for nonlinear manifold and robust to outliers, while $\lambda(.)$ (eq. 4, PTV) is an advanced version of Gaussian kernel by considering inlier noise.

4.3. Dense Voting

After the sparse vote, a dense vote can be performed to explore the underlying data structure if needed, such as the out-of-sample extension manifold. The related data space (such as the convex hull of the data) $\boldsymbol{y}$ is quantized on a grid by precision requirement, and all grid elements collect vote information from the input data $\boldsymbol{x}$. The vote equations of dense vote are slightly different from the stick vote.

$$\boldsymbol{S}(\boldsymbol{y_i}, \sigma_v, \sigma_n) = \sum_j [\int_{N_j^*} p_j^*(\boldsymbol{x}) T(\boldsymbol{x}, \boldsymbol{y_i}, \boldsymbol{T_j}, \sigma_v) d\boldsymbol{x}] \tag{10}$$

Here N_j^* is the half range of $1d$ normal direction which has cross angle less than $\frac{\pi}{2}$ with $\boldsymbol{p2}$. $p_j^*(x)$ is a unilateral Gaussian distribution with mean x_j and isotropic co-variance σ_n^2 in N_j^*. The same changes also take place for the polarity vote. After getting geometric information for all points on this grid, non-maximum suppression can be used to find the underlying manifold [18].

5. Theoretical Justification

Mathematics Preliminary: let $\boldsymbol{M}$ be a smooth submanifold (in differential sense) in Euclidean space R^2 of intrinsic dimensionality 1, then for any point $\boldsymbol{q} \in \boldsymbol{M}$, there are 1D normal space and tangent space, denoted by $\boldsymbol{N_p}$ and $\boldsymbol{T_p}$. For inlier noise point $\boldsymbol{q}$, we can write $\boldsymbol{q} = \boldsymbol{p} + \boldsymbol{e}$ where $\boldsymbol{p} \in \boldsymbol{M}$ and $\boldsymbol{e}$ is the error vector. Then, the normal space of $\boldsymbol{q}$ is defined by the normal space of $\boldsymbol{p}$.

5.1. Probabilistic Ball Voting

The key assumption of the ball voting is as follows: the uncertainty of the voter's position can be modeled as the isotropic distribution in the orthogonal space, which is $1d$ and orthogonal to the $1d$ line between voter and receiver. This can be explained by the following theorem.

Ball Voting Theorem: *The receiver's position is $\boldsymbol{x}$ in R^2. The first voter's position is a random vector $\boldsymbol{y}$ isotropically distributed in R^2, with mean $\boldsymbol{0}$. The second voter's position is a random vector $\boldsymbol{z}$ with the same isotropic distribution in space O_x, which is $1d$ and orthogonal to the line $\boldsymbol{0x}$. Then, the ball vote tensor $B_{\boldsymbol{xy}}$ has the same eigenvectors as $B_{\boldsymbol{xz}}$.*

Basically, these two votes' results have the same eigenspace, and the corresponding eigenvalues have the same ranking. The first advantage of the assumption is the computational cost can be reduced, since the dimension of the uncertainty space is reduced. Secondly, there is no risk of hole integration in the second formulation, since the support of the distribution of the voter does not include receiver, no matter what the distribution is. For the first formulation, if the support includes the receiver, then no definition exists in standard tensor voting. It can only be solved by Cauchy principle value method, while this problem does not exist for the second formulation. Due to space limitations, the proof is in the supplementary material.

5.2. Probabilistic Stick Voting

The key assumption of stick voting is as follows: the uncertainty of the voter's position can be modeled as a normal distribution in the voter's normal space.

Tubular Neighborhood Theorem: *let $\boldsymbol{M}$ be a smooth $1D$ manifold (in differential sense) in Euclidean space R^2. Then, there exists $\epsilon > 0$ such that for any point $\boldsymbol{q}$ close to $\boldsymbol{M}$ with distance at most ϵ, there is a unique expression $\boldsymbol{q} = \boldsymbol{p} + \boldsymbol{v}$, where $\boldsymbol{p} \in \boldsymbol{M}$ and $\boldsymbol{v}$ is the normal space at point $\boldsymbol{p}$.*

The proof of this theorem can be found in any classical differential geometry book. This theorem states that for any point close to the manifold $\boldsymbol{M}$, there is a corresponding projection point $\boldsymbol{p}$ at $\boldsymbol{M}$. q can be moved to $\boldsymbol{p}$ lying in its normal space N_q. We already know that standard tensor voting gives precise normal space estimation for points on the manifold, and the approximate normal space estimation for points with small inlier errors. Based on the estimated normal space information, probabilistic stick vote allows points to have uncertainty in their normal space, so the new vote can get more precise tensor estimation results. More analysis is given in the supplementary material.

6. Experiments

6.1. Evaluation Methodology and Implementation

We compare our probabilistic tensor voting (PTV) with other non-parametric methods, including standard tensor voting (STV) [12], PCA [6] and diffusion map based manifold denoising (MD) [5]. Parametric methods, such as RANSAC, are not considered here. We examine the quality of algorithms with endpoint completion, tangent space estimation, data denoising and image contour grouping. PTV and STV are applied on all tasks while PCA and MD are designed for denoising task only. Several data sets are used in our experiments, including synthetic data, the logo of POCV, 3D Stanford Bunny data, and the BSDS image

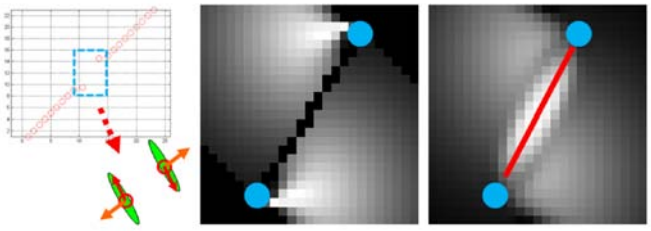

Figure 5. **Comparison between STV and PTV for endpoints completion.** Left: input tokens; middle: endpoints completion results by STV; right: endpoint completion results by PTV

database [8]. We chose them for their different characteristics in intrinsic structure (linear or nonlinear), application domain, multi or single manifold, curvatures (small or large), number of samples (varies from a few hundred to thousands), etc. For 3D data, we naturally extend PTV algorithm to 3D as the same way in [12].

We also used Local PCA (LPCA) [17] for tangent space estimation. Preliminary results did not support this algorithm as a viable option, because LPCA is sensitive to data sampling density on the manifold and outliers. So we omit those results. To further reduce the computational cost, vote only takes place in a neighborhood with a distance threshold s_{max}, which is calculated as $exp(-(\frac{s_{max}^2}{\sigma_v^2})) = 0.001$. The number of voting is reduced from $O(N^2)$ to $O(NM)$, making the computational cost a linear function of the input data size N. Here M is the average size of neighborhood depending on σ_v and σ_n.

6.2. Results

Endpoint completion: Fig. 5 shows the comparison between STV and PTV for endpoints completion. Synthetic error (0.5 shift) is added to the endpoints of two line segments. These two line segments should be connected by a straight line, which is missing in this figure. The target of the completion is to recover this missing line, and similar work has been done in [11], [19], where there are no errors in location. At the sparse vote stage, both STV and PTV use two pass votes to estimate the tensor at each point. Then the tensors at two endpoints are used to generate dense saliency map by one pass dense vote. Since STV gets incorrect tensors at those two endpoints, the missing line can not be recovered. For PTV, $\boldsymbol{p2}$ points in the direction of the endpoints error. By moving the endpoints' positions aligned to $\boldsymbol{p2}$ (unilateral Gaussian distribution), the missing line is perfectly recovered by dense vote ($\sigma_v = 30$, $\sigma_n = 0.5$).

Manifold tangent space estimation: correctly estimating the tangent (normal) space of manifolds under noisy conditions is an important step for many applications, including contour grouping, manifold denoising, mesh reconstruction and smoothing, etc. In this section, we intensively investigate the robustness of STV and PTV for this target by adding different levels of inlier and outlier noise. The logo of POCV, sphere data and Stanford Bunny data are used here. The comparison metric we use is the average error of the estimated tangent space on inlier points by two pass votes,

$$e = [\sum_{i=1}^{Num_I} cos^{-1}(|\boldsymbol{N_i}^T \boldsymbol{G_i}|)]/Num_I \qquad (11)$$

Num_I is the number of inliers, Num_O is the number of outliers and outlier ratio (OR) is Num_O/Num_I. $\boldsymbol{N_i}$ is the estimated tangent (normal) space and $\boldsymbol{G_i}$ is the ground-truth at $\boldsymbol{x_i}$. It is clear that PTV produces higher errors as the noise level increases. The point here is to demonstrate that PTV is *consistently better* than STV in diversified noise environments, even in *extremely noisy conditions*.

(1), the tangent space ground-truth of POCV logo can be calculated analytically. 11 levels of inlier noise (standard deviation (SD) from 0 to 2) with 11 levels of outliers (OR from 0% to 300%) are added to the data. Estimated error by two pass votes is reported in Table 1.[2] It shows STV is quite robust as OR increases but sensitive to SD. PTV is consistently better than STV, in a large range of noise, and decreases the error by 25% on average. For instance, for data with $SD = 0.6$ and $OR = 300\%$, PTV achieves only 8 degree error, while it is even difficult for humans to recognize POCV from the noisy environments. By setting $\sigma_v = 10$ for all experiments, we show that PTV algorithm is quite robust. Furthermore, we fix σ_v and observe the performance via different choices of σ_n. Results show that PTV is not sensitive to σ_n. For instance, for data with $SD = 0.6$ and $OR = 300\%$, estimated errors with large range of σ_n are reported in Table 2 and the performance is almost a constant to different values of σ_n ($\sigma_v = 10$). (2), for the sphere data, we also add different levels of Gaussian noise and uniformly distributed outlier noise to the data (number of points from 1600 to 6400). We found that PTV is consistently better than STV and achieves 10% error reduction on average. (3), for Stanford Bunny data (Fig. 6, reconstruction results), the ground-truth is calculated by using STV on noiseless points cloud. Since outliers bring difficulties for reconstruction, we only added different levels of Gaussian inlier noise. The mesh reconstruction (1889 vertexes and 3851 faces) is based on the noiseless point could. In this case, PTV is also consistently better than STV and reduces error by 12% on average.[3]

Data denoising: we compare our PTV algorithm with PCA and MD, which are two popular non-parametric denoising methods. Points are sampled from an arc curve with small Gaussian noise and uniformly distributed outliers (50%). From Fig. 7, we can see that PCA finds the global principal

[2] When SD is large (≥ 0.4), performance is dominated by inlier noise.

[3] In 3D, we use one pass vote, and the detailed comparison is omitted.

Table 1. **Tangent space estimation on POCV data.** PTV results are bold

OR/SD	0.4	0.8	1.2	1.6	2.0
0%	0.158/**0.105**	0.316/**0.224**	0.344/**0.247**	0.407/**0.299**	0.479/**0.336**
150%	0.150/**0.128**	0.293/**0.206**	0.346/**0.214**	0.428/**0.298**	0.466/**0.328**
300%	0.168/**0.130**	0.258/**0.207**	0.340/**0.287**	0.432/**0.371**	0.473/**0.391**

Table 2. **PTV is robust to σ_n.** σ_n v.s. error of the estimated tangent space

σ_n	0.2	0.4	0.6	0.8	1.0	1.2	1.4	1.6	1.8	2.0
Error	0.148	0.140	0.133	0.130	0.137	0.141	0.141	0.139	0.140	0.138

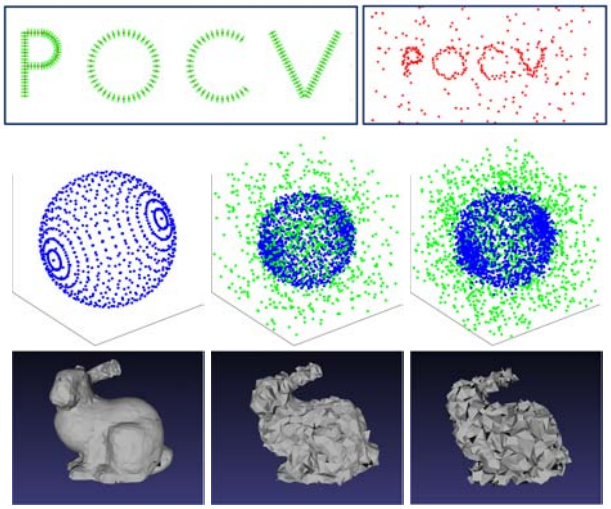

Figure 6. **Tangent space estimation.** The first row, noiseless POCV logo with ground truth of normal directions and $SD = 1$ with 200% OR. The second row, noiseless sphere, $SD = 0.5$ with 50% OR and $SD = 1$ with 100% OR. The third row, noiseless Bunny, $SD = 0.5$, $SD = 1$

direction of the data, which is incorrect due to the nonlinear intrinsic structure and outliers. MD can smooth inlier points after several iterations but can not recognize outliers and it attracts close outliers to diffusion process if we keep running the algorithm. In contrast, PTV can filter out outliers and get information from inlier data simultaneously by two pass votes, and inlier data can be further smoothed by moving points in their normal direction. One may argue that MD or PCA could be combined with some outlier detection methods, like clustering. In fact, the motivation here is to demonstrate PTV is robust to outliers for nonlinear data denoising, while MD or PCA not. Since PTV is a purely local method, it can also be strengthened by combining with other approaches, like the local to global alignment method [17].

Image Contour Grouping: we applied our algorithm to BSDS image database [8]. The problem we consider is how to group edgels in natural images. Edgel grouping results

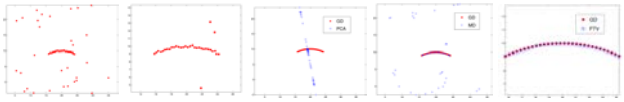

Figure 7. **Data denoising.** From left to right: original data, inlier data, PCA results, MD results and PTV results (the scale of the last one is adapted to only visualize inliers).

should be connected, smooth, and not sensitive to image noise, illumination change and contour occlusion. It is different from object contour detection [15], which involves high-level knowledge. This problem is also associated with multi-region image segmentation [14]. We start from edge detection results in [8].[4] As shown in Fig. 8, the initially detected edgels results clearly suffer from noise, which requires robust grouping. For instance, the edge in the first figure of the second row misses a part (left part of the endpoint is missing), because the boundary contrast is not high enough to infer an edge locally.

In the first row, the slope line shaped edge looks like a step edge because of textures, and it induces errors for STV, especially for $\boldsymbol{p1}$ (red vector). The first and the fourth points at each step are detected as endpoints. On the other hand, PTV gets tensors which are more likely to be orthogonal to the edge direction, with more precise $\boldsymbol{p1}$. Endpoints are extremely important for image contour grouping, and reducing error for $\boldsymbol{p1}$ is crucial for endpoints detection. In the second row, PTV provides precise $\boldsymbol{T}$ and $\boldsymbol{p2}$ (red vector) on endpoints, which is important for further endpoint completion. On the right of the second row, there is a comparison between STV (left) and PTV (right) on a T-junction. $\boldsymbol{p2}$ (red vector) in PTV points to the location of the L-junction, which is helpful for precise junction localization. In the last row, given a horizontal edge with a small jump, STV produces false alarms of endpoints detection, while PTV does not.

Based on the local grouping results by PTV, local to global contour completion is performed by connected

[4]The probability boundary detector provides a $1/0$ edge map with 8 filter responses for each edgel. Initial tensor is generated by summing up 8 filter responses together, then tensor voting is performed on the edgels.

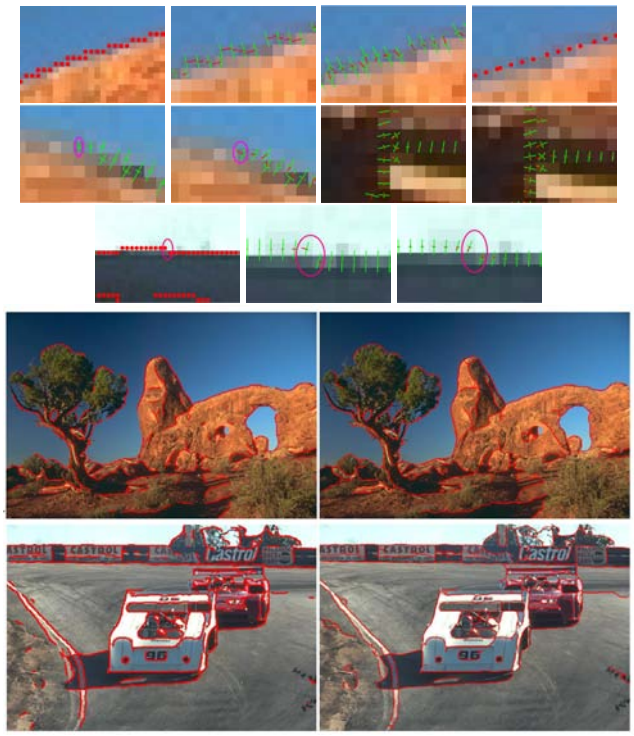

Figure 8. **PTV performs better than STV for edgel grouping.** First three rows: local contour grouping results; last two rows: initial contour detection (left) and contour completion results (right).

boundary with detected key points, i.e., endpoints and junctions. There are three types of completion; endpoint to boundary, endpoint to endpoint and junctions to boundary. As a necessary step, we cluster detected key points into groups by local graph clustering. Details are omitted because of limited space. The last two rows of Fig. 8 illustrate the contour completion results. In each row, the left one is the boundary detection results by [8] and the right one is completion results. After completion, the quality of contours is improved since disconnected boundary fragments are grouped and false detections are removed.

7. Conclusion

In this paper, probabilistic Tensor Voting is proposed to process data with both outlier and inlier noise. The advantages of this framework are: it's non-parametric, robust to both inlier and outlier noise, can handle complicated structures, and has the same computational cost as STV. From the experimental results, it is clear that our framework outperforms other candidate methods. When data suffers from both types of noise, the improvement of our framework is significant.

Acknowledgements

This work was supported in part by NIH Grant EY016093. The authors would like to thank Philippos Mordohai and Fei Sha for their helpful discussions.

References

[1] J. Choi and G. Medioni. Starsac: Stable random sample consensus for parameter estimation. In *Proc. CVPR*, pages 675–682, 2009. 2

[2] M. Fischler and R. Bolles. Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography. *Comm. of the ACM*, 24:381–395, 1981. 2

[3] D. Gong and G. Medioni. Dynamic manifold warping for view invariant action recognition. In *Proc. ICCV*, pages 571–578, 2011. 1

[4] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2nd edition edition, 2004. 2

[5] M. Hein and M. Maier. Manifold denoising. In B. Schölkopf, J. Platt, and T. Hoffman, editors, *Advances in Neural Information Processing Systems 19*, pages 561–568. MIT Press, Cambridge, MA, 2007. 2, 5

[6] I. T. Jolliffe. Principal component analysis. *Springer, 2nd edition*, 2002. 2, 5

[7] F. D. la Torre and M. J. Black. A framework for robust subspace learning. *Int. J. of Computer Vision*, 54:117–142, 2003. 2

[8] D. Martin, C. Fowlkes, and J. Malik. Learning to detect natural image boundaries using local brightness, color, and texture cues. *IEEE Trans. on PAMI*, 26:530–549, 2004. 1, 6, 7, 8

[9] J. Matas, C. Galambos, and J. Kittler. Robust detection of lines using progressive probabilistic hough transform. *CVIU*, 78:119–137, 2000. 2

[10] S. Mika, B. Schölkopf, A. J. Smola, K. R. Müller, M. Scholz, and G. Rätsch. Kernel pca and de-noising in feature spaces. In *NIPS 11*, pages 536–542. MIT Press, Cambridge, MA, 1999. 2

[11] P. Mordohai and G. Medioni. Junction inference and classification for figure completion using tensor voting. In *Fourth Workshop on Perceptual Organization in Computer Vision*, 2004. 1, 2, 6

[12] P. Mordohai and G. Medioni. *Tensor Voting: A Perceptual Organization Approach to Computer Vision and Machine Learning*. Morgan and Claypool Publishers, 2007. 1, 2, 3, 4, 5, 6

[13] P. Mordohai and G. Medioni. Dimensionality estimation, manifold learning and function approximation using tensor voting. *Journal of Machine Learning Research*, 11:411–450, 2010. 1, 2, 3

[14] T. Pock, A. Chambolle, D. Cremers, and H. Bischof. A convex relaxation approach for computing minimal partitions. In *Proc. CVPR*, pages 810–817, 2009. 7

[15] P. Srinivasan, L. Wang, and J. Shi. Grouping contours via a related image. In D. Koller, D. Schuurmans, Y. Bengio, and L. Bottou, editors, *Advances in Neural Information Processing Systems 21*, pages 1553–1560. 2009. 7

[16] C. Tang and G. Medioni. Curvature-augmented tensor voting for shape inference from noisy 3d data. *IEEE Trans. on PAMI*, 24:858–864, 2002. 3

[17] Y. W. Teh and S. Roweis. Automatic alignment of local representations. In M. Kearns, S. Solla, and D. Cohn, editors, *Advances in Neural Information Processing Systems 15*, pages 841–848. MIT Press, Cambridge, MA, 2003. 6, 7

[18] W. Tong, C. Tang, P. Mordohai, and G. Medioni. First order augmentation to tensor voting for boundary inference and multiscale analysis in 3d. *IEEE Trans. on PAMI*, 26:594–611, 2004. 5

[19] L. R. Williams and K. K. Thornber. A comparison of measures for detecting natural shapes in cluttered backgrounds. *IJCV*, 34(2-3):81–96, 2000. 6

[20] X. Zhao and G. Medioni. Robust unsupervised motion pattern inference from video and applications. In *Proc. ICCV*, pages 715–722, 2011. 1, 2