Feature Space Analysis Using Low-order Tensor Voting

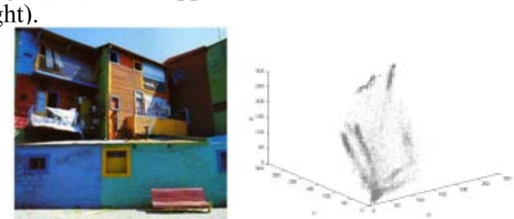
Jia Wang, Hanqing Lu, Qingshan Liu
National Laboratory of Pattern Recognition, Institute of Automation,
Chinese Academy of Sciences, P.O.Box 2728, Beijing 100080, China
Emails:{wangjia, luhq, qsliu}@nlpr.ia.ac.cn

Abstract

In this paper, low-order Tensor Voting, which was formerly used for structure inference from sparse data, is extended for feature space analysis. It is a nonparametric technique, because it does not have embedded assumptions. The methodology and possible applications are analyzed systematically. Its relation to Kernel Density Estimation and Mean Shift is also established, based on what the utilities for two fundamental analyses of feature space, density estimation and mode detection, are discussed. At last, two low-level vision tasks, image segmentation and motion analysis, are described as applications of the low-order Tensor Voting. Several experimental results illustrate its excellent performance.

1. Introduction

Feature space-based analysis is a paradigm which can provide a reliable representation of the input. A feature space is a mapping of the input obtained through the processing of the data in small subsets at a time. For each subset, a parametric representation of the feature of interest is obtained and the result is mapped into a point in the multidimensional space of the parameter. After the entire input is processed, significant features correspond to denser regions in the feature space, i.e., to clusters, and the goal of the analysis is the delineation of these clusters. Figure 1 shows an example in which all colors in an image (left) are mapped into the 3-D RGB color space (right).



(a) Color image (b) RGB color space

Figure 1. An example of feature space.

The analysis of the feature space is application independent. While there are a plethora of published clustering techniques, most of them are not adequate to analyze feature spaces derived from real data. Methods which rely upon a priori knowledge of the number of the clusters present (including those which use optimization of a global criterion to find this number), as well as methods which implicitly assume the same shape (most often elliptical) for all the clusters in the space, are not able to handle the complexity of a real feature space. For a recent survey of such methods, see [8, Section 8].

In this paper, the Tensor Voting theory, which was formerly used for structure inference from sparse data, is extended for feature space analysis. It becomes a nonparametric technique, because it does not have embedded assumptions. The number and delineation of the clusters will be a natural outcome of the tensor voting process.

Here, we will focus our mind on the methodology and application of low order Tensor Voting. Its relation to Kernel Density Estimation and Mean Shift is established, based on what the utilities for two fundamental analyses of feature space, density estimation and mode detection, are discussed. At last, two low-level vision tasks, image segmentation and motion analysis, are described as applications of the low-order Tensor Voting. Several experimental results illustrate its excellent performance.

2. Tensor voting

In 1997, *Tensor Voting* theory was first introduced by Guy and Medioni for structure inference from sparse data [1, 3]. The methodology of *tensor voting* can be grounded on two elements: *tensor calculus* for data representation, and *voting* for data communication.

The procedure is: In high-dimensional space, every point is encoded into a tensor. In the voting stage, tensors communicate their preferred information in a neighborhood through a predefined tensor field, and cast a tensor vote. Each site collects all the votes cast at its location and encodes them into a new tensor. After this refinement process, each tensor can describe a certain kind of information. As in Guy and Medioni's method, 2-order tensor can be used to generate descriptions of structure information (in terms of surfaces, curves, and junctions) from sparse and noisy data in 2-D or 3-D.

Here, we will analyze the methodology of tensor voting systematically and enumerate much more possible applications rather than structure inference.

2.1. Tensor calculus

An n-order tensor in m-dimensional space is a mathematical object[2] that has n indices and m^n components and obeys certain transformation rules. Each index of a tensor ranges over the number of dimensions of space. Tensors are generalizations of scalars (0-order, which have no indices), vectors (1-order, which have a single index), and matrices (2-order, which have two indices) to an arbitrary number of indices.

In this paper, scalars are denoted by lower case letters ($a, b, \ldots$), vectors by bold lower case letters ($\mathbf{a}, \mathbf{b} \ldots$), matrices by calligraphic upper-case letters ($\mathcal{A}, \mathcal{B} \ldots$), and higher-order tensors by bold upper-case letters ($\mathbf{A}, \mathbf{B} \ldots$).

0-7803-8554-3/04/$20.00

0-order tensor: When a point in n-dimensional space is encoded into a 0-order tensor, an extra scalar [a] is assigned to the point. Then, 0-order tensor is represented by [a] together with the point's coordinates. For different purposes, this parameter [a] can be used to describe different properties. For example, as will be discussed later in Section 3, [a] can be used to represent "density" at its site.

1-order tensor: In this case, a vector $\mathbf{a}=[a_1\ a_2 \ldots a_n]^T$ is assigned to the point. In n-dimensional space, this vector can describe many properties such as velocity, acceleration, motion vector, gravitation, gradient, etc.

2-order tensor: 2-order tensor is a matrix:

$$A = [\mathbf{a_1} \quad \mathbf{a_2} \quad \ldots \quad \mathbf{a_n}] = \begin{bmatrix} a_{11} & a_{21} & \ldots & a_{n1} \\ a_{12} & a_{22} & \ldots & \ldots \\ \ldots & \ldots & \ldots & \ldots \\ a_{1n} & \ldots & \ldots & a_{nn} \end{bmatrix} \qquad (1)$$

Even more information can be encoded into this matrix. For instance, Guy and Medioni [1] used it to describe geometric structures: curves, surfaces and intersections.

High-order tensor could be used for more complicate analysis, i.e., M.A.O. Vasilescu and D. Terzopoulos used high-order tensor for multilinear analysis of ensembles of facial images[4].

2.2. Voting

Tensors communicate with each other in the voting process. There are two fundamental elements [3] in this step: *voting field* and *votes accumulation*.

Voting field is a potential field derived from a tensor, which characterize the influence of a tensor to its neighborhood. Voting field has the same order as the tensor who cast it. 0-order tensor cast a 0-order voting field, 1-order tensor cast a 1-order one, etc.

For a given tensor **A**, voting field $\mathbf{V_A}$ can be defined as either isotropic or anisotropic, based on the requirement of analysis. Both can be derived from a **Voting kernel** or several kernels. Generally, the influence of a tensor should decrease smoothly with distance. Figure 2 shows two gradually changed voting fields in 2-D.

Votes accumulation characterizes the co-influence to a site by several tensors. Every tensor casts a vote to a site. What does the total influence look like when these votes work together? All the votes will be accumulated based on certain rules to produce the final vote.

For given tensors $\mathbf{A_1},\mathbf{A_2},\ldots\mathbf{A_m}$, voting fields $\mathbf{V_{A1}}, \mathbf{V_{A2}},\ldots \mathbf{V_{Am}}$, the final vote V will be:

$$\mathbf{V} = \underset{i=1}{\overset{m}{\Omega}} \mathbf{V_{Ai}} \qquad (2)$$

Ω, rules of summation, define how the votes would be added together. Note that Ω do not change the order but could be nonlinear.

3. Low-order tensor voting toward feature space analysis

Having the definitions demonstrated in Section 2, tensor voting is prepared to solve more problems in high-dimensional feature space analysis. In this section, two examples will be shown: *0-order tensor voting for density estimation* and *1-order tensor voting for mode detection (clustering)*.

3.1. Density estimation

When 0-order tensor voting is used for density estimation, the scalar [a] characterizes density.

In n-dimensional space R^n, the *voting field* (vote casting from **x** to **x'**) is defined as

$$\begin{aligned} v_{\mathbf{x}\to\mathbf{x'}} &= \|H\|^{-1/2} k(\|H^{-1/2}.[d_1 \quad d_2 \quad \ldots \quad d_n]^T\|) \\ &= \|H\|^{-1/2} k(\|H^{-1/2}.(\mathbf{x'}-\mathbf{x})\|) = k_H(\mathbf{x'}-\mathbf{x}) \end{aligned} \qquad (3)$$

where d_i is the coordinates of (**x'-x**) projected to the ith dimension. H is a symmetric positive $n \times n$ bandwidth matrix. Let h_i be the bandwidth in the ith dimension, then

$$H = diag[h_1^2, h_2^2, \ldots h_n^2] \qquad (4)$$

In (3), voting kernel $k(x)$, together with H, produces the anisotropic voting field $v_\mathbf{x}$.

If $h_1=h_2=\ldots=h_n=h$, the voting field will be isotropic.

Figure 2 shows the two kinds of 0-Order voting fields in 2-D with color-coded strength.

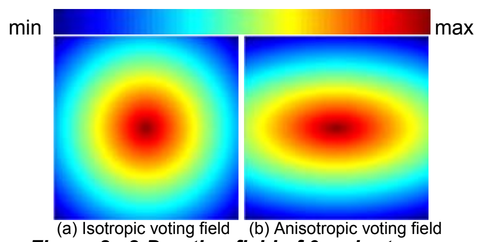


Figure 2. 2-D voting field of 0-order tensor.

If there are m points in the n-dimensional space, the *votes accumulation* is given by

$$v_\mathbf{x} = \underset{i=1}{\overset{m}{\Omega}} v_{\mathbf{x_m}\to\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m} v_{\mathbf{x_m}\to\mathbf{x}} = \frac{1}{m}\sum_{i=1}^{m} k_H(\mathbf{x}-\mathbf{x}_i) \qquad (5)$$

Here, we are surprised to find that the final expression (5) is just the same as the most popular density estimation method: **Kernel Density Estimation** [5,6]. So, it is further confirmed that 0-order tensor voting can be used for density estimation. Many conclusions of Kernel Density Estimation can be introduced.

In this paper, we only concern about such isotropic voting fields as

$$v_{\mathbf{x}\to\mathbf{x'}} = v_{\mathbf{x'}\to\mathbf{x}} = \frac{1}{h^n} k(\left\|\frac{\mathbf{x'}-\mathbf{x}}{h}\right\|^2) \qquad (6)$$

$$v_\mathbf{x} = \frac{1}{mh^n}\sum_{i=1}^{m} k(\left\|\frac{\mathbf{x}_i-\mathbf{x}}{h}\right\|^2) \qquad (7)$$

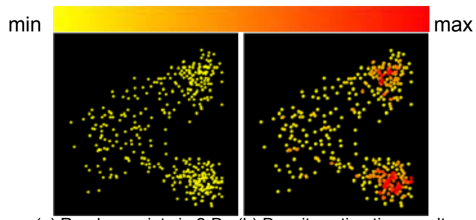
It has been approved by the use of kernel density estimator that they can suffice for most applications we are interested in. Finally, we get the density estimation

$$[a]_\mathbf{x} = v_\mathbf{x} = \frac{1}{mh^n}\sum_{i=1}^{m} k(\left\|\frac{\mathbf{x}_i-\mathbf{x}}{h}\right\|^2) \qquad (8)$$

Figure 3 is an example of density estimation in 2-D with kernel

$$k(x) = (2\pi)^{-1} \exp(-\frac{1}{2}x), \qquad h = 1 \tag{9}$$

Figure 3(b) shows the estimation result (with color-coded strength: redder color means higher density).



(a) Random points in 2-D (b) Density estimation result

Figure 3. Density estimation.

3.2. Mode detection (clustering)

Similar to 0-order tensor voting for density estimation, 1-order tensor voting can be used for density gradient estimation as well as mode (local maximum of estimated density) detection.

In 1-order tensor voting, a vector is assigned to the point in *n*-dimensional space, which describe the direction and magnitude of gradient.

1-oerder voting field can be expressed as

$$\mathbf{v}_{\mathbf{x}\to\mathbf{x}'} = \mathbf{k}(\mathbf{x}' - \mathbf{x}) \tag{10}$$

Considering the relationship between gradient and density, the *voting field* is given by

$$\mathbf{v}_{\mathbf{x}\to\mathbf{x}'} = \frac{2}{h^{n+2}}(\mathbf{x} - \mathbf{x}')g(\left\|\frac{\mathbf{x} - \mathbf{x}'}{h}\right\|^2) \tag{11}$$

$$g(x) = k'(x)$$

with the *votes accumulation*

$$\mathbf{v}_{\mathbf{x}} = \frac{2}{mh^{n+2}}\sum_{i=1}^{m}(\mathbf{x}_i - \mathbf{x})g(\left\|\frac{\mathbf{x}_i - \mathbf{x}}{h}\right\|^2) \tag{12}$$

Figure 4 shows the voting field of function (11).

Based on **Mean Shift** theory, we can define another *voting field* and *votes accumulation* as

$$\mathbf{v}_{\mathbf{x}\to\mathbf{x}'} = (\mathbf{x} - \mathbf{x}')g(\left\|\frac{\mathbf{x} - \mathbf{x}'}{h}\right\|^2) \tag{13}$$

$$\mathbf{v}_{\mathbf{x}} = \omega^{-1}\sum_{i=1}^{m}(\mathbf{x}_i - \mathbf{x})g(\left\|\frac{\mathbf{x}_i - \mathbf{x}}{h}\right\|^2), \tag{14}$$

where $\omega = \sum_{i=1}^{m} g(\left\|\frac{\mathbf{x}_i - \mathbf{x}}{h}\right\|^2)$.

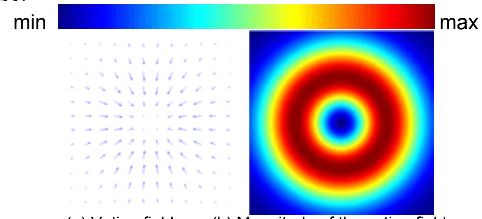
The 1-order tensor voting procedure for mode detection and clustering is:

1. For every point in the feature space, $\mathbf{v}_{\mathbf{x}}$ define a path leading to a nearby mode.

2. All the modes can be detected by the convergence of the paths.

3. A cluster is formed by the detected mode together with a group of points on the path lead to it.

Note the shape of the cluster can be arbitrary adapting to the structure of feature space, and the number of the clusters will be a natural outcome of the mode seeking process.



(a) Voting field (b) Magnitude of the voting field

Figure 4. 2-D voting field of 1-order tensor.

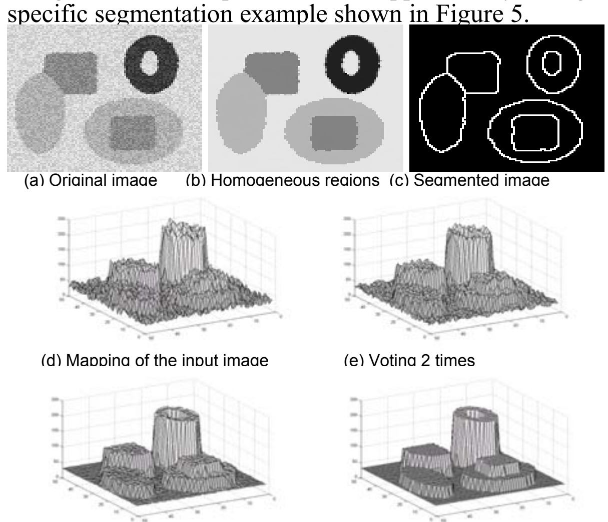
4. Applications

Two related applications using tensor voting are discussed in the sequel: *image segmentation* and *motion analysis.*

4.1. Image segmentation

Image segmentation, decomposition of a gray level or color image into homogeneous regions, is an important low-level vision task. Homogeneity is usually defined as similarity in pixel values, i.e., a piecewise constant model is enforced over the image.

In this section, an image segmentation method using 1-order tensor voting is presented. For illustration purpose, we give the description of our approach by using a specific segmentation example shown in Figure 5.



(a) Original image (b) Homogeneous regions (c) Segmented image

(d) Mapping of the input image (e) Voting 2 times

(f) Voting 4 times (g) Result after voting 6 times

Figure 5. Image segmentation.

In Figure5, for a 8-bit gray-level image (Fig. 5a), every pixel together with its value is mapped to a 3-D space (Fig. 5d) and encoded into a 1-order 3-D tensor with initial vector

$$\mathbf{a}_0 = [0 \quad 0 \quad 0]^T \tag{15}$$

Then, tensor voting procedure is performed between every two points with voting field (13) and kernel (9).

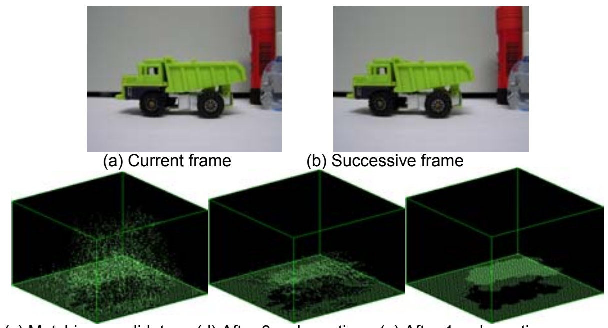
Let $\mathbf{x}$=[x y z]^T be the initial coordinates for a arbitrary point. After votes accumulation

$$\mathbf{a} = \mathbf{a}_0 + \mathbf{v} = \mathbf{v} \tag{16}$$

Vector **a** characterizes the influence of all other points toward **x**. It is a kind of gravity that forces the point to move from **x** to **x**+**a**. When the point finished moving, **a** will go back to his initial value $\mathbf{a}_0$.

This tensor voting procedure is an iterative process, which will not stop until all the **v** (or **a**) in the space is smaller than predefined threshold. After tensor voting, pixels will be clustered into homogeneous regions with the same value characterizing their homogeneity, see (Fig. 5b) and (Fig. 5g). The segmentation result is shown in (Fig. 5c).

4.2. Motion analysis



(a) Current frame (b) Successive frame

(c) Matching candidates. (d) After 0-order voting. (e) After 1-order voting.

Figure 6. Motion analysis.

Layered models provide a natural way to estimate motion when there are several regions having different velocities. But one weakness of existing layered representations is that they assign pixels to layers independently of pixels at neighboring locations. In doing so, their underlying generative model does not manifest the constraint that most physical objects are spatially coherent and have boundaries, nor does it represent relative depths and occlusion.

In this paper, we use 0-order and 1-order tensor voting to extract the motion layers.

In [7], M. Nicolescu and G. Medioni have employed a 2-order tensor voting procedure for the motion analysis. Yet, by experiments, we found that 0-order tensor voting is power enough to provide a satisfying result, and more importantly, its computation complexity is reduced. 1-order voting is an extra selective process, which can produce a more smooth motion layer.

The procedure of motion analysis is:

First, two successive frames that involve general motion are taken as input. For every pixel in the first frame, a normalized cross-correlation procedure is used to produce candidate matches in the second image, where all peaks of correlation are retained as candidates. When a peak is found, its position is also adjusted for sub-pixel precision according to the correlation values of its neighbors. Finally, each candidate match is represented as a (x,y,v_x,v_y) point in the 4-D space of image coordinates and pixel velocities, with respect to the first image.

Then, in 4-D space, several points may correspond to one pixel in the image. Each point describes a possible velocity for the pixel.

Second, 0-order tensor voting is performed in the 4-D space to infer the density information. This is based on the intuition that correct velocity estimation should have a higher probability. After voting, for a pixel in the image, only the velocity with the most density (probability) is retained.

Finally, 1-order tensor voting is performed to extract the motion layer. The voting process in this step is similar to that for image segmentation.

Figure 6 illustrates the recovered v_x velocities within layers. Higher velocity has a higher altitude in the axis box.

5. Conclusions

In this paper, Tensor Voting theory, which was formerly used for structure inference from sparse data, is extended for feature space analysis. It is a nonparametric technique for the analysis of complex multi-dimensional feature space and can provide reliable solutions for many vision tasks. We focus our mind on the theory and application of low-order Tensor Voting. Its relation to Kernel Density Estimation and Mean Shift is established. Two low-level vision tasks, image segmentation and motion analysis, are described as applications of the low-order Tensor Voting. Several experimental results illustrate its excellent performance.

6. Acknowledgements

This research is supported by France Telecom R&D Division, and the National Natural Science Foundation of China (Grant No. 60135020 and 60121302).

7. References

[1] G. Guy, G. Medioni, “Inference of Surfaces, 3-D Curves, and Junctions from Sparse, Noisy 3-D Data”, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Nov.1997, vol.19, no.11, pp.1265-1277.

[2] R. Abraham, J.E. Marsden and T.S.Ratiu, *Manifolds, Tensor Analysis, and Applications, 2nd ed.* New York: Springer-Verlag, 1991.

[3] G. Medioni, M.S. Lee and C.K. Tang, *A Computational Framework for Feature Extraction and Segmentation*, Elsevier, 1999.

[4] M.A.O. Vasilescu, D. Terzopoulos, “Multilinear analysis of image ensembles: TensorFaces”, *ECCV 2002*, Copenhagen, Denmark, May.2002, pp.447-460.

[5] D. Comaniciu, P. Meer. “Mean shift: A robust approach toward feature space analysis”. *IEEE Trans. Pattern Analysis and Machine Intelligence*, May.2002, vol.24, no.5, pp.603-619.

[6] R.O. Duda, P.E. Hart, *Pattern Classification and Scene Analysis*, Wiley, 1973

[7] M. Nicolescu, G..Medioni, “Motion Segmentation with Accurate Boundaries-A Tensor Voting Approach”, *CVPR2003*, Madison, Wisconsin, June 2003.

[8] A.K. Jain, R.P.W. Duin and J. Mao, “Statistical Pattern Recognition: A Review”, *IEEE Trans. Pattern Analysis and Machine Intelligence*, Jan.2000, vol.22, no.1, pp.4-37.