Tensor field Graph-Cut for Image Segmentation: A Non-convex Perspective

Hu Zhu, *Member, IEEE,* Jieke Zhang, Guoxia Xu, *Member, IEEE,* and Lizhen Deng, *Member, IEEE,*

***Abstract*—Image segmentation is a key component of image analysis, which refers to the process of partitioning the image into multiple segments. Graph cut is widely used in image segmentation by constructing a graph that the minimal cut of this graph would lead to partition the corresponding pixels of the different objects. In this paper, we reconstruct the graph cut problem as a special non-convex optimization problem instead of the traditional maximum flow problem. We extend this non-convex problem to the hypergraph method and combine it with a tensor field based on a directional bilateral filter bank to achieve segmentation in grayscale images. Accordingly, an efficient minimization algorithm is proposed to solve this non-convex problem with global convergence. Furthermore, we have selected the data of BSDS300 and BSDS500 as tests. Experimental results and evaluation index tests further demonstrate the superiority of the proposed method.**

***Index Terms*—Image Segmentation, Tensor Field, Nonconvex Optimization, Hypergraph Cut.**

I. Introduction

Recently, the graph cut based segmentation methods have obtained lots of attention [1]–[5]. In brief, the graph cut algorithm maps the image pixels onto a graph, whose nodes represent the pixels and the edges represent similarity. The result of segmentation can be acquired by grouping the graph into two clusters through the knowledge of Graph Theory [6]. The corresponding clusters are explicitly referred to as background and foreground according to appearance and adjacency properties of nodes.

Graph cut based segmentation fully considers the boundary and region information to achieve the global optimal result [7]–[9]. The mainstream solution of min-cut in graph cut problem has a minimum cost, which is solving the equivalent maxflow problem by using Push-relabel method or Ford-Fulkerson method [10]. However, the traditional method may not get ideal segmented object due to the occluded objects and sometimes it costs large calculation to solve the equivalent maxflow problem [11]. To improve the performance or solve the problem in a more efficiently way, some works change the structure of original energy function [2]. One classical method is adding the regular term into energy function, which can incorporate the shape information of object into energy function and get ideal segmented object [3], [12]–[18], such as the star-shape regularizer [12], [13], connectivity regularizer [14], [15], or convexity-shape regularizer [3], [16] for energy function. Moreover, in [19], feature-aligned segmentation method is proposed, which is used for segmenting a mesh into patches whose boundaries are aligned with prominent ridge and valley lines of the shape. And in [20], they proposed a method that utilizes contrast prior, which used to be a dominant cue in none deep learning based SOD approaches, into CNNs-based architecture to enhance the depth information. These shape-based graph cut algorithms can work well with known prior information and descriptive shape. Some works also introduced the norm regularization that can be solved effectively, especially ℓ_1 norm regularization. Sinop and Grady [21] proposed a common framework that minimizes the energy in graph cut with respect to ℓ_1 norm. Their work shows the connection between graph cut problem and ℓ_1 norm. Then, for solving the problem in a more efficiently way, Bhusnurmath *et al.* [22] reformulated the graph cut problem as an unconstrained linear problem and introduced the ℓ_1 norm to measure it. Likewise, in [23], an ℓ_1-regularizer is adopted to promote sparse solution in a nonlinear embedding, which can be easily integrated into existed segmentation framework. Not only could these proposed algorithm be solved by many popular optimization methods but also they expose connections between graph cut and other related continuous optimization problems.

In this paper, we reformulate the graph cut problem as a special non-convex problem, which is seen as a matrix recovery problem of binary labeling and is also suitable for the solutions in the case of hypergraph model. Then a proximal minimization algorithm is proposed to tackle this challenging problem efficiently with global convergence and higher accuracy. Apart from that, we build a tensor field based on the graph and employ direction bilateral bank, which is able to capture edge information of the object. It makes our cut algorithm utilize more information when calculating the weight of the edge of graph in the boundary region and robust under the noise, such as Gaussian noise.

Our motivation can be summarized in two terms.

(1) In the graph model, it is popular for traditional methods that achieve the weight of edge by calculating the ’similarity’ between two corresponded pixels. However, it is inevitable that the high-level semantic graph clues are discarded under only considering piece-wise pixel information and spatial information for graph representation [1], [24], [25]. It is evident that the boundaries of objects can be considered as the discontinuation between pixel values. Here, we highly resort this discontinuation under high-level geometrical ob-

This work is supported by the National Natural Science Foundation of China under Grant 61701259. *(Corresponding author: Lizhen Deng) (E-mail: alicedenglzh@gmail.com)*

Hu Zhu and Jieke Zhang are with Jiangsu Province Key Lab on Image Processing and Image Communication, Nanjing University of Posts and Telecommunications, Nanjing 210003, China. Guoxia Xu is with Department of Computer Science, Norwegian University of Science and Technology, 2815 Gjovik, Norway. Lizhen Deng is with National Engineering Research Center of Communication and Network Technology, Nanjing University of Posts and Telecommunications, Nanjing, 210003, China.

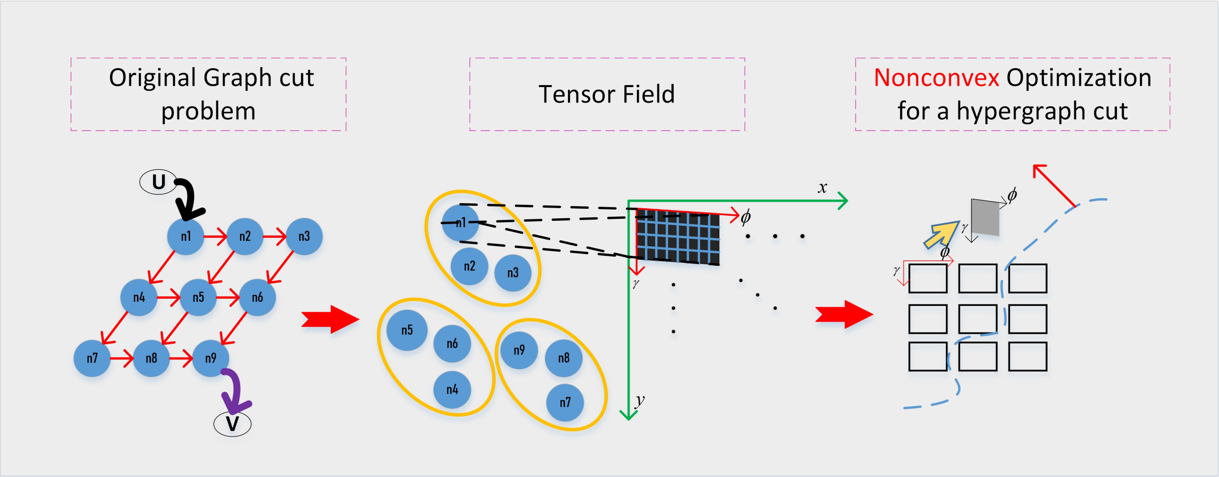
Original Graph cut problem
Tensor Field
Nonconvex Optimization for a hypergraph cut
U
V
n1
n2
n3
n4
n5
n6
n7
n8
n9
x
y
ϕ
γ

Fig. 1. The flowchart of the proposed method. First part is the classical graph model. Second part is the tensor field combined with hypergraph model. Third part is using our proposed algorithm to solve nonconvex problem in tensor filed.

servations of the hypergraph. Besides, there are other much more information that can be utilized, such as anisotropy, orientation, and gradient information. To provide the comprehensive representation for an image, some works construct the vector field [26], local structure tensor [27] or tensor field [28] for a image. Inspired by these works that the tensor is regarded as a multi-dimensional data set for larger capacity. Here we employ the directional bilateral filter to construct a tensor field for graph representation. There are mainly two reasons to introduce a directional bilateral bank for tensor field construction: First, it can extract the orientation and anisotropy at the edge of the object. Second, it keeps the original poverty of bilateral kernel function, which will increase the robustness of the algorithm under noise.

(2) Some works show that there is a connection between graph cut problem the ℓ_1 norm optimization problem [22]. But it is known that ℓ_1 norm sometimes yields biased estimators and the solution is suboptimal [29]. It should be noted that some nonconvex functions are utilized to replace the linear regularization in statistic learning and image restoration for sparse estimation recently [30]–[32]. For example, in [30], a nonconvex penalty method is used for statistic estimation and showed a better solution compared to ℓ_1 penalty. And a family of nonconvex data fitting models are proposed in [33] and showed that the nonconvex methods could restore the image with neat edges compared with convex methods. Moreover, using the nonconvex functions onto the singular values of the square deal matrix of the tensor and matrix can achieve the better approximation in low-rank tensor [34] and matrix recovery [35], [36] respectively. To the best of our knowledge, this is the first work that shows how to optimize a nonconvex function for the graph cut problem. Consequently, motivated by these works, to solve the graph cut problem more effectively and obtain the optimal solution, we propose a nonconvex optimization approach for graph cut problem and extend it hypergraph model that overcomes the limitation of the graph.The contributions of this paper are presented as follows:

- Aiming at the calculation of edge weight in the graph and high similarity between the foreground object and background in some image, we construct a tensor field with a directional bilateral bank for capturing more edge information of the object. It makes better use of edge information of objects and segmentation algorithm more robust under various noise.
- In the basic of the tensor field, a special nonconvex optimization problem is proposed for solving graph cut problem, which is also suitable for the solution in the case of the hypergraph.
- we propose a proximal minimization algorithm to tackle this problem effectively with global convergence.

The organization of the rest of this paper is as follows: In section 2, we review the related basic work about our method. Section 3 shows the details of the proposed algorithm. Extensive experimental result and corresponded analysis are given in section 4. And conclusion is given in section 6.

II. Background

In this section, we show the formulation from the graph cut problem to a linear optimization problem.

Suppose that the $G = (N, E)$ is consisting of the node set $N = \{n_1, n_2, ..., n_m\}$ and edge set $E = \{e_1, e_2, ..., e_m\}$. A min-cut of Graph G is dividing the N into two different set U and V. Then the sum of the weights of the edges connecting these two sets are minimized.The min-cut problem in a graph is typically equivalent to the associated max-flow problem [2]. Thus, starting with the max-flow problem, the edges and associated weight can be viewed as pipes and capacities respectively. The associated problem can be shown as follows:

$$\begin{aligned} &\max_{v} \ f(v) = h^T v \\ &s.t. \ Bv = 0 \\ &-\omega \le v \le \omega \end{aligned} \tag{1}$$

This inner product represents $h^T v$ the total flow out of the U and constraint $Bv = 0$ represents that flow in each interior node is zero. Here the $v \in R^m$ is the vector that is undirected and representing the flow of each edge in the graph. And the $h \in R^m$ is a binary vector, the +1 value represents the edges adapted from the node set U and 0 value elsewhere. The $B \in R^{n\times m}$ is a matrix whose rows and columns correspond to the nodes and edges respectively in graph. In each column, an +1 value denotes the starting point of an arrow and an -1 value for the ending point. Attentively, for the reason that matrix does not contain rows of nodes U and V, the columns of edges starting from source node U or ending at sink node V only contain a single value. And the inequality $-\omega \leq v \leq \omega$ denotes the capacity constraints of each edge.

By adding the Lagrangians x and β corresponding to the flow constraint and capacity constraint respectively, the associated Lagrangian dual function is obtained, which is formulated as follow:

$$\begin{aligned} &\min_{x,\beta:\beta>0} \ \omega^T(\beta_+ + \beta_-) \\ &s.t.\ B^T x - h = (\beta_- - \beta_+) \end{aligned} \tag{2}$$

Because the minimum value of $(\beta_+ + \beta_-)$ attains to $\left|B^T x - h\right|$ when x is fixed, which is also described in [22]. Then the above problem can be reformulated as:

$$\min_x \sum_{j=1}^{m} \omega_j \left|(B^T x - h)_j\right| \tag{3}$$

Since the ω is a constant, the structure of this problem is always used as data fitting term in image restoration or matrix recovery. In [22], this problem is formulated as unconstrained ℓ_1 norm minimization:

$$\min_x \left\|diag(\omega)(B^T x - h)\right\|_1 \tag{4}$$

And it is solved by using interior point methods and show that variable x will converge to binary values.

III. The Proposed Nonconvex Model

In this section, we establish a tensor field based on the directional bilateral bank to extract more feature information in an image for segmentation. Then a nonconvex optimization problem is proposed for graph cut model, and we extend it to the hypergraph cut model. Finally, a proximal minimization (PM) is proposed for solving this challenging problem.

A. Tensor Field based on Directional Bilateral Filter

In (3), the ω is weight of the edge in graph. The assignment of the weight is very important, which will influence segmentation result. Normally, the weight of the edge connected two neighboring pixels (a, b) is calculated like in Boykov and Jolly method [1], [37]:

$$\omega = \exp(-\frac{(I_a - I_b)^2}{2\mu^2}) \cdot \frac{1}{dist(a,b)} \tag{5}$$

The parameter I_a and I_a can be the intensity of these two neighboring pixels. The parameter μ can be seen as 'camera noise'. And the $dist()$ represents the Euclidean distance. If the intensity of two neighboring pixel is very close, the value of the weight ω will be very large. It means that these two pixels are likely to have the same label after cut. However, this measurement only considers pixel intensity information and spatial information. In order to utilize more information at the edge of the object, we apply a directional bilateral bank to construct a tensor field for each pixel. In this way, this tensor representation will contain more information, such as edge, orientation, and anisotropy information.

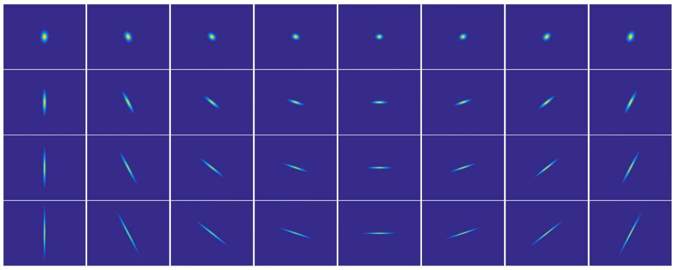

Fig. 2. Directional bilateral function with different scales and orientations.

The directional bilateral kernel incorporates both the orientation and the anisotropy information [38], which is defined as follows:

$$\begin{aligned} G^{DBF}_{a,b}((x_a, y_a), (x_b, y_b)) &= \exp(-\frac{\phi_1^2 m^2 + \phi_2^2 n^2}{2\sigma_d^2}) \\ \cdot \exp(-\frac{\left|I(x_a, y_a) - I(x_b, y_b)\right|^2}{2\sigma_r^2}) \end{aligned} \tag{6}$$

where the

$$m = (x_a - x_b)\cos\theta + (y_a - y_b)\sin\theta \tag{7}$$

$$n = -(x_a - x_b)\sin\theta + (y_a - y_b)\cos\theta \tag{8}$$

For an image I, the (x_a, x_b), (x_b, y_b) are the coordinates of pixels a and b respectively. And other parameters ϕ_1, ϕ_2 and θ determine the scales and orientations of the filter function. They also can obtain the orientation and anisotropy information when the filter smoothing along a certain direction. As shown in Figure 2, our model set the kernel function with four scales and eight directions.

According to the scale factor in [38], we have the $\phi_1 = 1/\varphi_2$, then we choose $\phi_2 \in \{1, 2, 3, 4\}$ and $\theta \in \left\{0, \frac{\pi}{8}, \frac{2\pi}{8}, \frac{3\pi}{8}, \frac{\pi}{2}, \frac{5\pi}{8}, \frac{6\pi}{8}, \frac{7\pi}{8}\right\}$. The construction for tensor field includes two steps:

Step 1) We use the directional bilateral filter with the four scales and eight orientations to convolve with the input image. Then we will get a tensor in $R^{M\times N\times 4\times 8}$ and the gray value of each pixel is embeded into the matrix representation:

$$[\mathbf{T}^{s,d}_{x,y}]_{S\times D} = [\begin{array}{ccc} G_{11}(y_{x,y}) & \cdots & G_{1D}(y_{x,y}) \\ \vdots & \vdots & \vdots \\ G_{S1}(y_{x,y}) & \cdots & G_{SD}(y_{x,y}) \end{array}] \tag{9}$$

The $\mathbf{T}^{s,d}_{x,y}$ represents the corresponding element in the matrix $\mathbf{T}$. And we have S=4, D=8, $G(\cdot)$ is the output generated by convolving the directional bilateral functions with the input image.

Step 2) We unfold the tensor T following the mode 1 and mode 2. Specifically, the every element of unfolding tensor field F is in the form of a matrix in $R^{4\times 8}$.

If the $\theta = 0$, the directional bilateral kernel will be equivalent to the bilateral filter. Thus, directional bilateral kernel can be used sufficiently and the filter result will contain edge information. In this way, every pixel of an image corresponds to a matrix which includes the edge, anisotropy and orientation information.

Then we calculate the weight in our tensor field by follow formulation:

$$\omega = \exp(-\frac{dist(T^{s,d}_{i,j}, T^{s,d}_{m,n})}{2\mu^2}) \cdot \frac{1}{dist((i,j),(m,n))} \tag{10}$$

B. Nonconvex Optimization Problem

In (4), it is well known that l_1-norm may yield biased estimators and can not be used to obtain best solution. And some works show that nonconvex penalty can achieve the better solution than l_1-norm or convex penalty. Here we reformualte it as a nonconvex optimization problem for graph cut:

$$Y(x) = \min \sum_{j=1}^{m} P(\omega_j \left|(B^T x - h)_j\right|) \tag{11}$$

where $P(x)$ is a concave function.

Theorem 1. [33] *For $R_+ \rightarrow R_+$, $P(x)$ is convex and twice continuously differentiable on R_+. In addition, for $x \in (0, +\infty)$, $P^{'}(x) > 0$, $P^{'}(0) = P^{'}(0_+) = \lim_{x\to 0_+} P^{'}(x)$, and $P^{''}(x) < 0$, $P^{''}(0_+) = \lim_{x\to 0_+} P^{''}(x)$.*

There are many functions satisfy the above poverty, we give two concave functions that are widely used. The first is Exponential-Type function:

$$P(x) = \frac{1 - \exp(-\beta x)}{1 - \exp(-\beta)} \tag{12}$$

The second is German function:

$$P(x) = \frac{x}{x + \beta} \tag{13}$$

Here the $\beta > 0$, which is used to control the concavity of above two functions. The above problem is solvable globally and accurately via proposed proximal minimization algorithm in section 3.

C. Hypergraph Cut Model

Since the hypergraph considers the relationship among more than two nodes, we consider extending the model to the case of hypergraph model. The difference between the hypergraph and graph is that several nodes can share one edge in hypergraph [39]. And the weight of hyperedge is defined as the sum of all pairwise distance of nodes. Suppose $G_{hyper} = (M, K)$ as a hypergraph that contains the node set $M = \{m_1, m_2, \cdot\cdot\cdot, m_n\}$ and hyperedge set $K = \{k_1, k_2, \cdot\cdot\cdot, k_m\}$, and the weight of each hyperedge is $\omega(k_i)$. Similar to the cut in graph, we have

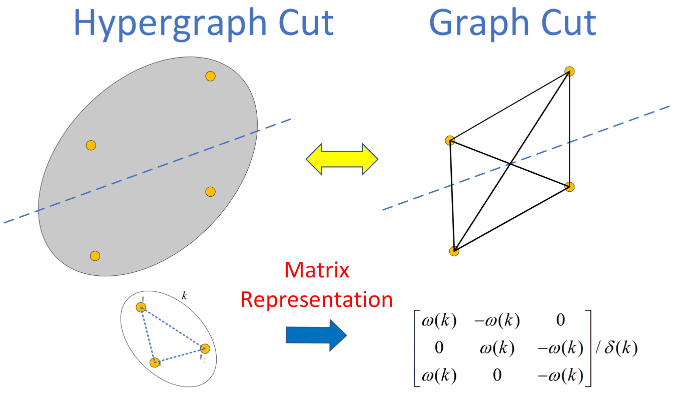
Hypergraph Cut
Graph Cut
k
m1
m2
m3
Matrix Representation
$\begin{bmatrix}\omega(k) & -\omega(k) & 0\\ 0 & \omega(k) & -\omega(k)\\ \omega(k) & 0 & -\omega(k)\end{bmatrix} / \delta(k)$

Fig. 3. Extend the model to hypergraph.

Definition 1. *A cut of hypergraph $G_{hyper} = (M, K)$ can divide the M into two parts L^c and L. The $\partial L := \{k \in V | k \cap L \neq \varnothing, k \cap L^c \neq \varnothing\}$ is defined as the set of hyperedge that is to be cut. In this way, a cut for the hypergraph could be defined as:*

$$Cut(L, L^c) = \sum_{k\in\partial L} \omega(k) \frac{|k \cap L|\,|k \cap L^c|}{\delta(k)} \tag{14}$$

The L is a node subset after cut and L^c is the compliment of L. $\delta(k)$ is the degree of a hyperedge.

As illustrated in [40], in above equation, each hyperedge can be regarded as a subgraph that contains several subedges and all subedges are assigned with the same weight $\omega(k)/\delta(k)$. Then one subedge is divided into two parts with the node set $|k \cap L|$ and $|k \cap L^c|$, and there are $|k \cap L|\,|k \cap L^c|$ subedges are cut.

Above all, we can regard the cut of a hyperedge as the cut of a subgraph that also can apply our model. Here we consider that a hyperedge is consist of a node with its at least two nearest neighbors: for a certain hyperedge, each node is connected with other nodes by a subedge. And the weight of each subedge in a hyperedge has the same weight $\omega(k)/\delta(k)$. As shown in Figure 3, each subgraph represents as a matrix that will have the same value of weight ω. And there are also two elements in each row of the matrix, and it represents that these two nodes belong to the one hyperedge. And the value of each element is the weight of one hypergraph. It also means that the partition of two nodes in hypergraph takes more cost than in graph .

D. A proximal minimization algorithm

Here we propose a proximal minimization algorithm (PM) to solve the problem (11) [35]. Since $\omega > 0$, we let $H_j = \omega_j * B^T_j$ and $b_j = \omega_j * h_j$. Then the problem in (11) is formulated as follows:

$$Y(x) = \min \sum_{j=1}^{m} P(|H_j x - b_j|) \tag{15}$$

Here we use the Taylor formulation to linearize the $Y(x)$ at x_i with an added proximal term:

$$\begin{aligned} Y(x|x_i) &= \sum_{j\in J}\{P(|H_j x_i - b_j|) \\ &+P'(|H_j x_i - b_j|)(|H_j x - b_j| - |H_j x_i - b_j|) \\ &+ \frac{\sigma}{2}\|x - x_i\|^2)\} \end{aligned} \tag{16}$$

When x_i is fixed, $P'(|H_j x_i - b_j|)\,|H_j x_i - b_j|$ is regarded as a constant. Because of the complexity of $F(x|x_i)$, we consider solving it inexactly. Then let:

$$\begin{aligned} x_{i+1} &\approx \arg\min_x Y(x|x_i) \\ &= \arg\min_x\{\sum_{j\in J} P'(|H_j x_i - b_j|)\,|H_j x - b_j| \\ &+ \frac{\sigma}{2}\|x - x_i\|^2\} \end{aligned} \tag{17}$$

The x_i also represents the ith step of PM algorithm. Until we find:

$$t_{i+1} \in \partial Y(x_{i+1}|x_i), \|t_{i+1}\| \le \sigma\varepsilon\,\|x_{i+1} - x_i\| \tag{18}$$

Here t is the subgradient of Y at x_{i+1}, ε is a constant that satisfies the $0 \le \varepsilon < \frac{1}{2}$. The whole step of PM is displayed in Algorithm 1.

```
Algorithm 1: PM algorithm for solving the (15)
  Step 0: Set given parameter $\sigma > 0$. Input $x_0$ and let
    $i = 0$.
  Step 1: Calculate $x_{i+1}$ by (17).
  Step 2: If the termination criterion is not met, let
    $i = i + 1$ and go back to step 1.
```

In (17), the sub-problem is solved by the alternating direction method of multiplier algorithm (ADMM) [41]. Introducing the Lagrangian multiplier λ and penalty parameter α, and we have $z_j = H_j x - b_j$. The augmented Lagrangian function is obtained:

$$\begin{aligned} L(z, x, \lambda) &= \sum_{j\in J} P^{'}(|H_j x_i - b_j|)\,|z_j| + \frac{\sigma}{2}\,\|x - x_i\| \\ &- \lambda^T(z_j - (H_j x - b_j)) + \frac{\alpha}{2}\|z_j - (H_j x - b_j)\|^2 \end{aligned} \tag{19}$$

And the iteration scheme of the ADMM is shown as follows:

$$z^{l+1} = \arg\min_z\{L(z, x^l, \lambda^l)\} \tag{20}$$

$$x^{l+1} = \arg\min_x\{L(z^{l+1}, x, \lambda^l)\} \tag{21}$$

$$\lambda^{l+1} = \lambda^l\text{-}\delta\alpha_1[z^{l+1} - (Hx^{l+1} - b)] \tag{22}$$

Finally, the z^{l+1} is obtained by:

$$\begin{aligned} z^{l+1} &= \mathrm{sgn}(Hx^l - b + \lambda^l/\alpha) \\ &* \max\{\left|Hx^l - b + \lambda^l/\alpha\right| - d/\alpha, 0\} \end{aligned} \tag{23}$$

and the x^{l+1} can be obtained by:

$$\begin{aligned} x^{l+1} &= [H^T(\alpha_1 z^{l+1} - \lambda_1^l) + \alpha_1 H^T b + \sigma x_i] \\ &/(\sigma I + \alpha_1 H^T H) \end{aligned} \tag{24}$$

The ADMM for solving the subproblem (17) is described in Algorithm 2.

```
Algorithm 2: ADMM algorithm for solving the (17)
  Step 0: Set given parameter $\sigma > 0$ and $\alpha > 0$. Initialize
    $z^0$, $x^0$, $\lambda^0$ and $l = 0$.
  Step 1: Calculate $z^{l+1}$ by (23).
  Step 2: Calculate $x^{l+1}$ by (24).
  Step 3: Update $\lambda^{l+1}$ by (22).
  Step 4: If the termination criterion is not met, let
    $l = l + 1$ and go back to step 1.
```

IV. Experiments

In this section, we do some experiments based on the BSDS500 and BSD300 database [42] by comparing with NACASH1 [43], NACASTV [43], One-Cut [4], SNC [44] and TR-Convexity [3]. The detail of these methods are displayed as follows:

Here, we use the hypergraph model that each hyperedge includes three nodes. The experiments are conducted on a computer with 8-GB random access memory and Intel(R) Core(TM) i5-7440HQ cpu, 2.80GHz processor, and the implementation is presented in Matlab 2016a.

A. Baseline Methods

Here in the experiments, we will start the comparisons between the NACASH1, NACASTV, One-Cut, SNC, TR-Convexity methods and our method.

- **NACASH1:** Normalized Cut based Segmentation Model with Adaptive Similarity and Spatial Regularization, proposed by Wang et al. in 2018. [43] on Normalized cut with adaptive similarity and spatial regularization.
- **NACASTV:** Normalized Cut based Segmentation Model with Adaptive Similarity and TV Regularization, proposed by Wang et al. in 2018. [43] on Normalized cut with adaptive similarity and spatial regularization.
- **One-Cut:** One-Cut method, proposed by Tang et al. in ICCV 2013. [4] on Grabcut in one cut.
- **SNC:** Scalable Normalized Cut, proposed by Chen et al. in 2017. [44] on Scalable Normalized Cut with Improved Spectral Rotation.
- **TR-Convexity:** shape convexity method for a new high-order regularization constraint, proposed by Gorelick et al. in 2017. [3] on Convexity shape prior for binary segmentation.

All the images are collected from the BSDS500 database. And we choose five salient images, the result of experiments as shown in the Fig. 4 and Fig. 5. From the first column to last column, it shows the result of NACASH1, NACASTV, One-Cut, SNC, TR-Convexity and our method respectively.

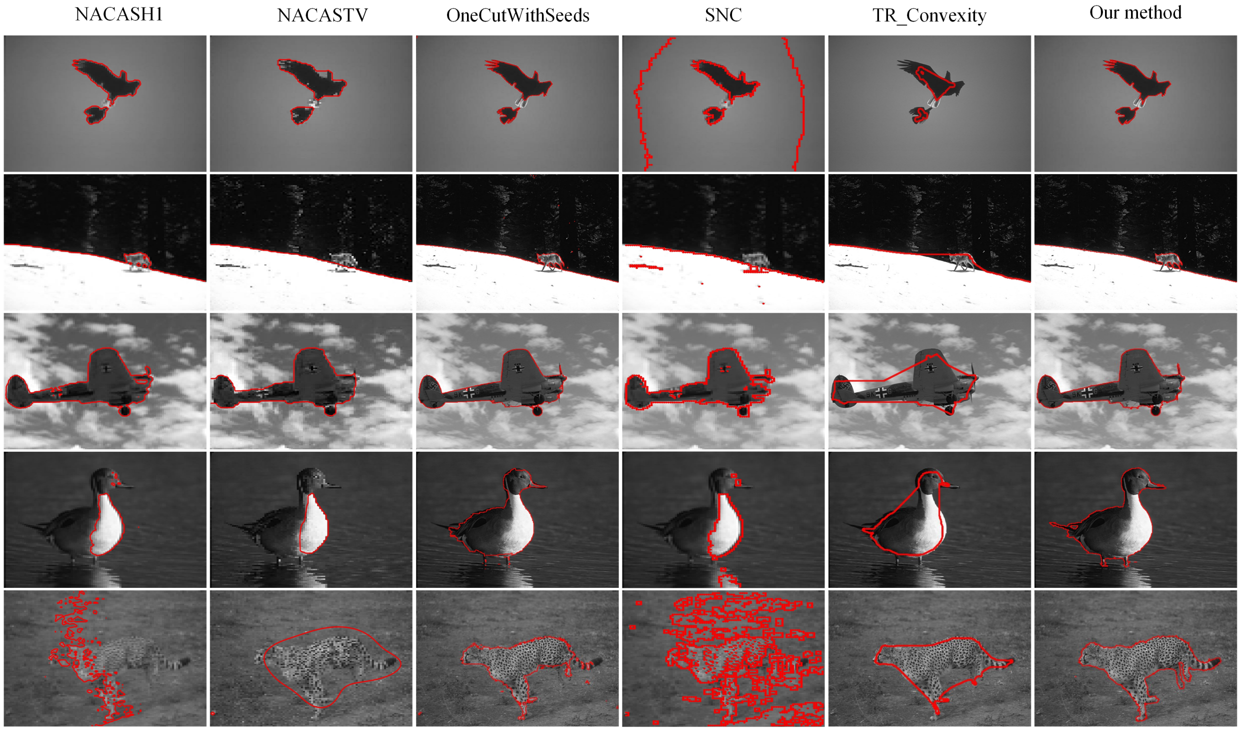

Fig. 4. The comparative experiment between the NACASH1, NACASTV, One-Cut and TR-Convexity methods and our method.

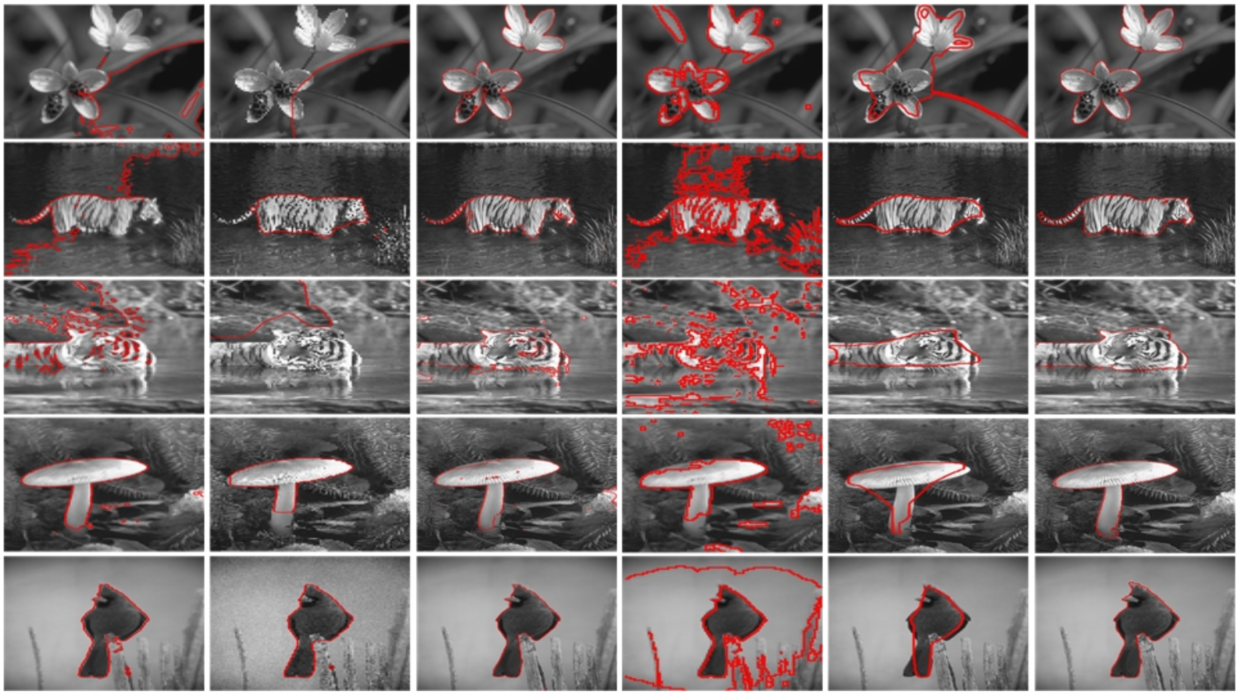

Fig. 5. The comparative experiment between the NACASH1, NACASTV, One-Cut and TR-Convexity methods and our method.

TABLE I
THE EVALUATION OF DIFFERENT METHODS FOR FIG.4

	Image 1	Image 2	Image 3	Image 4	Image 5
	VI/RI	VI/RI	VI/RI	VI/RI	VI/RI
NACASH1	0.1045 / 0.9856	0.1610 / 0/9835	0.1942 / 0.9702	0.7701 / 0.7688	1.5718 / 0.5011
NACASTV	0.1385 / 0.9798	0.1869 / 0.9779	0.2135 / 0.9691	0.7519 / 0.7617	0.8435 / 0.8043
One-Cut	0.1668 / 0.9811	0.2078 / 0.9757	0.5652 / 0.8847	0.3479 / 0.9323	0.3912 / 0.9341
SNC	1.1398 / 0.6095	0.3266 / 0.9482	0.3097 / 0.9517	1.3062 / 0.6324	1.3920 / 0.5968
TR-Convexity	0.2574 / 0.9600	0.2179 / 0.9747	0.5080 / 0.8204	0.5265 / 0.8923	0.4818 / 0.9179
Our Method	0.1355 / 0.9671	**0.1190 / 0.9882**	**0.1903 / 0.9846**	**0.2294 / 0.9523**	**0.1956 / 0.9746**

TABLE II
THE EVALUATION OF DIFFERENT METHODS FOR FIG.5

	Image 1	Image 2	Image 3	Image 4	Image 5
	GCE/LCE	GCE/LCE	GCE/LCE	GCE/LCE	GCE/LCE
NACASH1	0.4589 / 0.4426	0.1899 / 0.1932	0.4615 / 0.4434	0.2291 / 0.2367	0.1618 / 0.1694
NACASTV	0.4335 / 0.4116	0.1869 / 0.1779	0.4136 / 0.4389	0.1941 / 0.1629	0.1753 / 0.1811
One-Cut	0.1867 / 0.2018	0.1739 / 0.1839	0.1652 / 0.1836	0.1562 / 0.1489	0.1722 / 0.1822
SNC	0.2443 / 0.2551	0.4305 / 0.4249	0.4024 / 0.4296	0.2062 / 0.2345	0.3321 / 0.3490
TR-Convexity	0.2774 / 0.2614	0.2281 / 0.2069	0.1679 / 0.1743	0.1891 / 0.1902	0.2089 / 0.2379
Our Method	**0.1749 / 0.1917**	**0.1513 / 0.1438**	**0.1321 / 0.1335**	**0.1331 / 0.1411**	**0.1561 / 0.1676**

B. Evaluation Index

These results show apparently that our method can get better performance compared to other methods. In order to evaluate the results of different methods, we employ two quantitative criteria to measure the similarity between the segmentation results and ground truth. They are Variation of information (VI) [45] and Rand Index (RI) [46]. The Variation of information is the distance of shared information between the two variables, which is used for clustering comparisons. The definition of it is given as follow:

$$VI(T_R, T_G) = H(T_R) + H(T_G) - 2I(T_R, T_G) \tag{25}$$

where H denotes the entropy and I represents the mutual information between the segmentation result T_R and ground truth T_G.

The Rand Index is used to calculate the similarity between the two clustering results. The definition of RI is an index that tracks the performance of segmentation that determined by the true predicted labels compared with groundtruth.

$$\begin{aligned} RI\{T_R, \{T_G\}\} = \frac{1}{M} \sum_{m<n} [s_{mn} p_{mn} \\ + (1 - s_{mn})(1 - p_{mn})] \end{aligned} \tag{26}$$

Here, the p_{mn} represents the probability of two pixels m and n with the same label, and s_{mn} represents the event that result and groundtruth with same label. M is the amount of pair of pixels. The value of the VI and RI are given in Table 1, which indicates our method has better performance. In table 1, it is obvious that the result of our method has smaller VI value and larger RI value. The overstrking number indicates that our method get the best result.

Moreover, we also use Global Consistency Error(GCE) and Local Consistency Error(LCE) to evaluate our method. For a given pixel p_i consider the segments in S_1 and S_2 that contain that pixel. The segments are sets of pixels. If one segment is a proper subset of the other, then the pixel lies in an area of refinement, and the local error should be zero. If there is no subset relationship, then the two regions overlap in an inconsistent manner. In this case, the local error should be non-zero. Let $\setminus$ denote set difference, and $|x|$ the cardinality of set x. If $R(S, p_i)$ is the set of pixels corresponding to the region in segmentation S that contains pixel p_i, the local refinement error is defined as:

$$E(S_1, S_2, p_i) = \frac{|R(S_1, p_i) \setminus R(S_2, p_i)|}{|R(S_1, p_i)|} \tag{27}$$

Given this local refinement error in each direction at each pixel, there are two natural ways to combine the values into a error measure for the entire image. Global Consistency Error (GCE) forces all local refinements to be in the same direction. Local Consistency Error (LCE) allows refinement in different directions in different parts of the image. Let n be the numbers of pixel:

$$GCE(S_1, S_2) = \frac{1}{n} \min \left\{ \sum_i E(S_1, S_2, p_i), \sum_i E(S_2, S_1, p_i) \right\} \tag{28}$$

$$LCE(S_1, S_2) = \frac{1}{n} \sum_i \min \{E(S_1, S_2, p_i), E(S_2, S_1, p_i)\} \tag{29}$$

C. Quantitative Analysis

The smaller value of GCE and LCE is, the result is more closer to groundtruth. Fig.5 is the test image and Table. II.

From the figures, we can obverse clearly that some methods fail to meet the requirements of distinguishing background and object, while some methods do, but the contour is zigzag and very rough. Our method not only accurately distinguishes the background and object from a picture, but also makes the contour as smooth as possible. At the same time, combined with the table, our indexes is the lowest of all comparison methods, so through these two indexes, we can also show the superiority of our method.

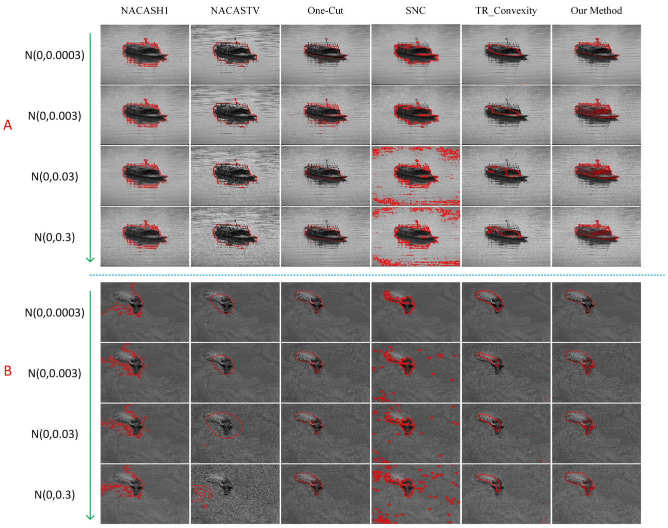

Fig. 6. Different methods under the gaussian noise with the variance of 0,0.0003,0.003,0.03 respectively.

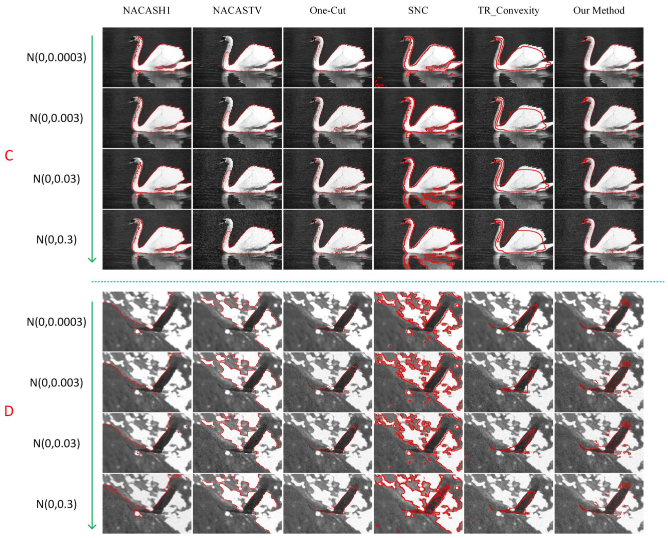

Fig. 7. Different methods under the gaussian noise with the variance of 0,0.0003,0.003,0.03 respectively.

D. The Test on Different Noise Levels

In this part, we show the contribution of the tensor field based on bilateral filter. We test the robustness of our algorithm under the gaussian noise with different intensity. The test data is also taken from the BSDS300 database. The original image is corrupted by gaussian noise with level N(0,0), N(0,0.003), N(0,0.003), N(0,0.03) respectively.

In Fig. 6(A), with the density of noise increasing, our method classifies the less noise point as background than other methods. Because the bilateral function can smooth the noise at the edge, even when the density increased to 0.03, our method still can segment well at the boundary of the object. In Fig. 6(B), the background of the input image is much more complicated than other images. But our image can get higher accuracy than other methods. In Fig. 7(C), when the density of noise is 0, our method achieves the good performance at detail part of the boat, such as boat pole. With the density increasing, although some points of the object are classified as background, our algorithm can keep the segmentation accuracy at the boundary of the boat, which show the tensor field make full use of orientation information at the boundary of objects . Similarly, In Fig. 7(D), the edge of buffalo the object and are weak, but our method also acquires the better result of segmentation. Correspondingly, Fig. 8 shows the segmentation accuracy of above experiments under different noise. It is clear that our accuracy is higher than other methods and can keep steady with the noise increasing, which shows the robustness of our algorithm in another view.

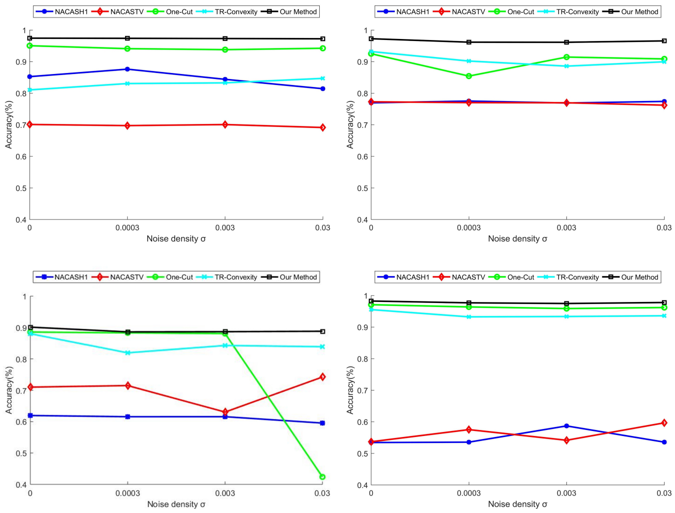

Fig. 8. The true accuracy of experiment 1, experiment 2(first row), experiment 3 and experiment 4(second row).

E. The Contribution of Tensorfield and Hypergraph

It can be seen clearly that the contour of the object after only Graph cut is not complete, and some parts are still inseparable from the background. By using Tensor field and Graph cut, the results show that the physical outline has been improved compared with the previous version, but the edge is rough and the serration is strong. This is not the perfect way. Our approach combines Tensor field, hypergraph and nonconvex. It can be seen intuitively that the contour of the object is not only separated from the background accurately, but also the overall contour edge is very smooth. This ablation analysis is demonstrated that the each component has greatly improved based on the baseline compared with the previous methods. Our method gets higher accuracy and achieves the good performance in some detail parts.

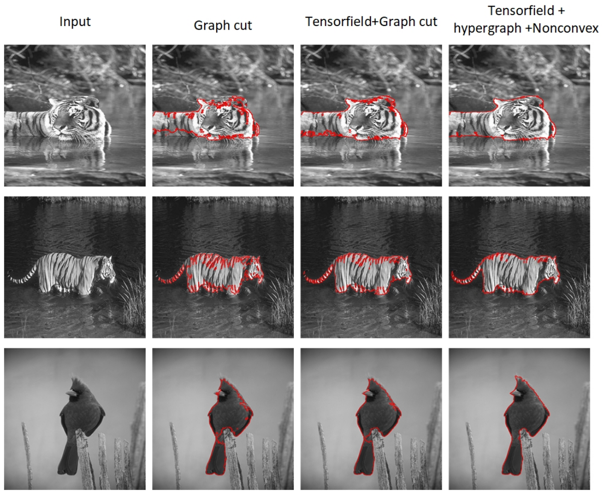

Fig. 9. The comparative experiment between the NACASH1, NACASTV, One-Cut and TR-Convexity methods and our method.

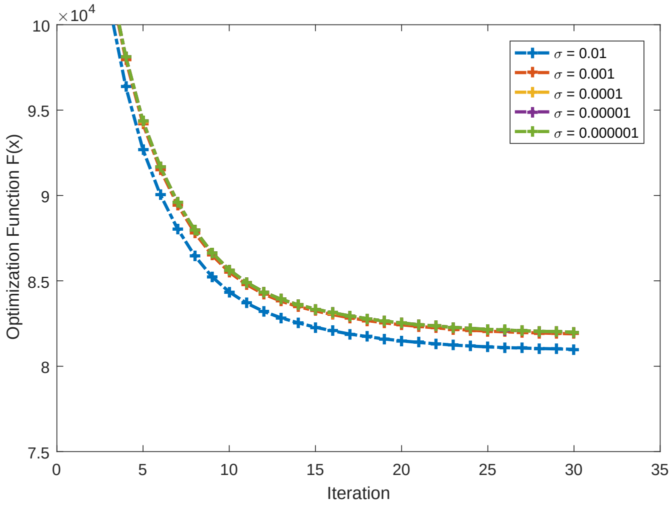

Fig. 10. The value of the $F(x)$ with different outer iteration steps under different σ.

F. Parameter Analysis

Here, we give the test under different value of β and γ. The result is shown in Fig. 12. It can be seen from the figure that when the value of b is constant, y is too small and the red circle is large. When the y value takes an appropriate value, the b value is too small, and the red circle is also small. When the red circle is too large, the red circle is in the background, and the separation of the object from the background fails. By analyzing the parameters, we can know that the convergence is excessive. When the red circle is too small and the red circle is in the object, the separation of the object from the background is also a failure. By analyzing the parameters, we can know that the convergence rate is too slow and the convergence state is not reached at this time.

G. Convergence Analysis

In the test, we use the $\beta \in \{1, 3, 5, 7, 11, 13, 15\}$ and $\gamma \in \{5, 10, 15, 20, 25\}$ for ET function. And we set $\alpha_1 = 1$, step length $\delta \in \{0.02, 0.05, 0.1, 0.2, 0.3, 0.5\}$ for inner ADMM iteration for all experiments. For σ, it can be seen in Fig. 10, our algorithm is not sensitivity to the value of σ. Here, we set $\sigma = 10^{-5}$. The terminal condition for inner ADMM iteration is:

$$\|x_{i+1} - x_i\| \,/\, \|x_i\| \leq 10^{-3} \tag{30}$$

where the relative error of iterations is set to be no more than 10^{-3}. Moreover, the maximum outer iteration and inner iteration is 13 and 350 respectively. And the proof of the golbal convergence is listed in the appendix. After setting parameters, we display the convergence curve and results under different iteration segmentation. As shown in Fig. 10 and Fig. 11, with the outer iterations steps increasing, the energy function $F(x)$ is gradually approaching to the convergence value and curve of segmentation is also becoming closer to the object.

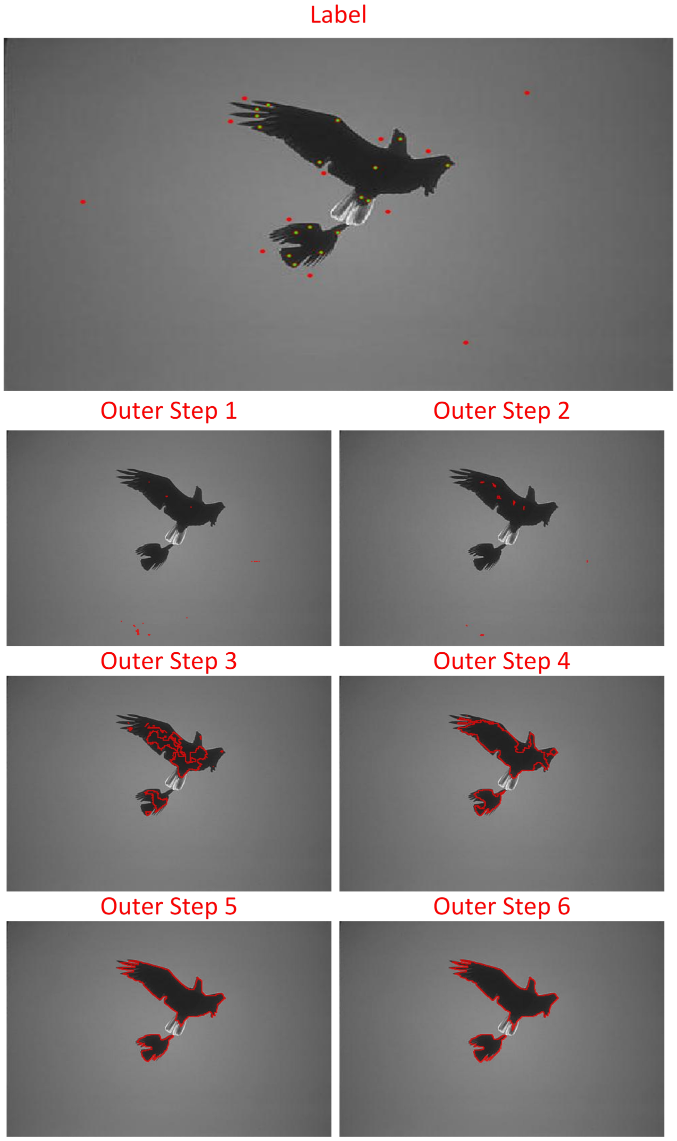

Fig. 11. The corresponded result of different outer iteration steps.

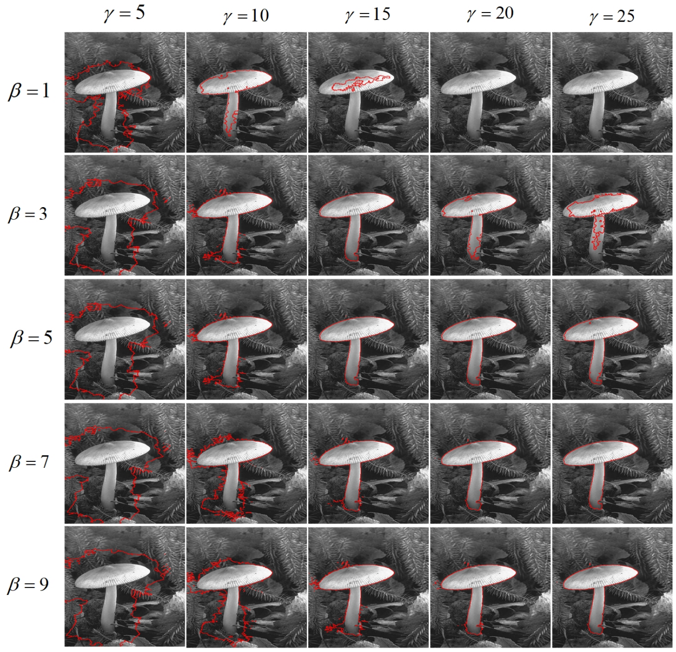

Fig. 12. The results under different values of β and γ parameters.

TABLE III
The Average Time Consumption Through Different Methods

	NACASHI	NACASTV	One-Cut	SNC	TRConvexity	Our Method
Time/s	4.13	3.35	2.32	0.50	1.67	10.74

H. The Computational Complexity

Here, we briefly discuss the complexity of our method. As shown in Fig. 1, we can find that the computational of our algorithm mainly consists of three parts: 1) The tensorfield construction, 2) hypergraph model construction and 3)using proposed proximal minimization algorithm to solve the model. We define that the size of image is $M \times N$. For the tensorfield construction, the computational complexity is $O\left(M \times N \times K\right)$. For the hypergraph construction, the computational complexity is $O\left(M \times N\right)$. Finally, for using proposed proximal minimization algorithm to solve the model, the computational complexity is $O\left(M^2 \times N^2\right)$. In this way, the total computational complexity is $O\left(M^2 \times N^2 + M \times N \times K + M \times N\right)$.The algorithm complexity and average computational time for images in Fig. 5 with different methods are shown in Table. III

V. Conclusion

We presented a tensor field graph-cut algorithm for image segmentation. In contrast to traditional graph cut methods, our algorithm utilizes more information in segmentation by building a tensor field based on directional bilateral filter, which also make the results robust under the noise. Moreover, the graph cut problem is formulated as nonconvex problem and we extended it to the case in hypergraph model that overcomes the limitation of the graph. To achieve the solution with high quality, a proximal minimization algorithm is proposed. Above all, we have presented a powerful nonconvex optimization algorithm for graph cut in image segmentation, which is confirmed by extensive experiments.

References

[1] Y. Y. Boykov and M.-P. Jolly, “Interactive graph cuts for optimal boundary & region segmentation of objects in nd images,” in *IEEE International Conference on Computer Vision*, vol. 1. IEEE, 2001, pp. 105–112.

[2] V. Kolmogorov and R. Zabih, “What energy functions can be minimizedvia graph cuts ?” *IEEE Transactions on Pattern Analysis Machine Intelligence*, no. 2, pp. 147–159, 2004.

[3] L. Gorelick, O. Veksler, Y. Boykov, and C. Nieuwenhuis, “Convexity shape prior for binary segmentation,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 2, pp. 258–271, 2017.

[4] M. Tang, L. Gorelick, O. Veksler, and Y. Boykov, “Grabcut in one cut,” in *IEEE International Conference on Computer Vision*, 2013, pp. 1769–1776.

[5] Y. Zhou, T. Zhang, S. Huo, C. Hou, and S. Kung, “Adaptive irregular graph construction based salient object detection,” *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2019.

[6] Z. Wu and R. Leahy, “An optimal graph theoretic approach to data clustering: Theory and its application to image segmentation,” *IEEE Transactions on Pattern Analysis Machine Intelligence*, no. 11, pp. 1101–1113, 1993.

[7] Y. Boykov, O. Veksler, and R. Zabih, “Fast approximate energy minimization via graph cuts,” in *IEEE International Conference on Computer Vision*, vol. 1. IEEE, 1999, pp. 377–384.

[8] Y. Liu, J. Han, Q. Zhang, and L. Wang, “Salient object detection via two-stage graphs,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 29, no. 4, pp. 1023–1037, April 2019.

[9] W. Xia, C. Domokos, L. Cheong, and S. Yan, “Background context augmented hypothesis graph for object segmentation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 4, pp. 582–594, April 2015.

[10] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein, *Introduction to algorithms*. MIT press, 2009.

[11] F. Yi and I. Moon, “Image segmentation: A survey of graph-cut methods,” in *IEEE International Conference on Systems and Informatics*. IEEE, 2012, pp. 1936–1941.

[12] O. Veksler, “Star shape prior for graph-cut image segmentation,” in *IEEE European Conference on Computer Vision*. Springer, 2008, pp. 454–467.

[13] V. Gulshan, C. Rother, A. Criminisi, A. Blake, and A. Zisserman, “Geodesic star convexity for interactive image segmentation,” in *IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 3129–3136.

[14] S. Vicente, V. Kolmogorov, and C. Rother, “Graph cut based image segmentation with connectivity priors,” in *IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE, 2008, pp. 1–8.

[15] S. Nowozin and C. H. Lampert, “Global interactions in random field models: A potential function ensuring connectedness,” *SIAM Journal on Imaging Sciences*, vol. 3, no. 4, pp. 1048–1074, 2010.

[16] L. Gorelick, O. Veksler, Y. Boykov, and C. Nieuwenhuis, “Convexity shape prior for segmentation,” in *IEEE European Conference on Computer Vision*. Springer, 2014, pp. 675–690.

[17] X. Liu, O. Veksler, and J. Samarabandu, “Order-preserving moves for graph-cut-based optimization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 32, no. 7, pp. 1182–1196, 2010.

[18] P. F. Felzenszwalb and O. Veksler, “Tiered scene labeling with dynamic programming,” in *IEEE International Conference on Computer Vision and Pattern Recognition*. IEEE, 2010, pp. 3097–3104.

[19] Y. Zhuang, H. Dou, N. Carr, and T. Ju, “Feature-aligned segmentation using correlation clustering,” *Computational Visual Media*, vol. 3, no. 2, pp. 147–160, 2017.

[20] J.-X. Zhao, Y. Cao, D.-P. Fan, M.-M. Cheng, X.-Y. Li, and L. Zhang, “Contrast prior and fluid pyramid integration for rgbd salient object detection,” in *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2019, pp. 3927–3936.

[21] A. K. Sinop and L. Grady, “A seeded image segmentation framework unifying graph cuts and random walker which yields a new algorithm,” *IEEE Transactions on Pattern Analysis Machine Intelligence*, 2007.

[22] A. Bhusnurmath and C. J. Taylor, “Graph cuts via ℓ_1 norm minimization,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 30, no. 10, pp. 1866–1871, 2008.

[23] Y. Yu, C. Fang, and Z. Liao, “Piecewise flat embedding for image segmentation,” in *IEEE International Conference on Computer Vision*, 2015, pp. 1368–1376.

[24] H. Wang, J. Shen, J. Yin, X. Dong, H. Sun, and L. Shao, “Adaptive nonlocal random walks for image superpixel segmentation,” *IEEE Transactions on Circuits and Systems for Video Technology*, pp. 1–1, 2019.

[25] F. Meng, H. Li, S. Zhu, B. Luo, C. Huang, B. Zeng, and M. Gabbouj, “Constrained directed graph clustering and segmentation propagation for multiple foregrounds cosegmentation,” *IEEE Transactions on Circuits and Systems for Video Technology*, vol. 25, no. 11, pp. 1735–1748, Nov 2015.

[26] T. F. Chan and L. A. Vese, “Active contours without edges,” *IEEE Transactions on Image Processing*, vol. 10, no. 2, pp. 266–277, 2001.

[27] D. Tschumperle and R. Deriche, “Diffusion pdes on vector-valued images,” *IEEE Signal Processing Magazine*, vol. 19, no. 5, pp. 16–25, 2002.

[28] B. Wang, X. Gao, D. Tao, and X. Li, “A unified tensor level set for image segmentation,” *IEEE Transactions on Systems, Man, and Cybernetics, Part B*, vol. 40, no. 3, pp. 857–867, 2010.

[29] R. Tibshirani, “Regression shrinkage and selection via the lasso,” *Journal of the Royal Statistical Society: Series B (Methodological)*, vol. 58, no. 1, pp. 267–288, 1996.

[30] J. Fan, L. Xue, and H. Zou, “Strong oracle optimality of folded concave penalized estimation,” *Annals of statistics*, vol. 42, no. 3, p. 819, 2014.

[31] H. Liu, T. Yao, R. Li, and Y. Ye, “Folded concave penalized sparse linear regression: sparsity, statistical performance, and algorithmic theory for local solutions,” *Mathematical programming*, vol. 166, no. 1-2, pp. 207–240, 2017.

[32] M. Nikolova, M. K. Ng, and C.-P. Tam, “On ℓ_1 data fitting and concave regularization for image recovery,” *SIAM Journal on Scientific Computing*, vol. 35, no. 1, pp. A397–A430, 2013.

[33] X. Zhang, M. Bai, and M. K. Ng, “Nonconvex-TV based image restoration with impulse noise removal,” *SIAM Journal on Imaging Sciences*, vol. 10, no. 3, pp. 1627–1667, 2017.

[34] X. Zhang, “A nonconvex relaxation approach to low-rank tensor completion,” *IEEE Transactions on Neural Networks and Learning Systems*, 2018.

[35] C. Lu, J. Tang, S. Yan, and Z. Lin, “Nonconvex nonsmooth low rank minimization via iteratively reweighted nuclear norm,” *IEEE Transactions on Image Processing*, vol. 25, no. 2, pp. 829–839, 2016.

[36] K. Mohan and M. Fazel, “Iterative reweighted algorithms for matrix rank minimization,” *Journal of Machine Learning Research*, vol. 13, no. Nov, pp. 3441–3473, 2012.

[37] C. Rother, V. Kolmogorov, and A. Blake, “Grabcut: Interactive foreground extraction using iterated graph cuts,” *ACM Transactions on Graphics*, vol. 23, no. 3, pp. 309–314, 2004.

[38] M. Venkatesh and C. S. Seelamantula, “Directional bilateral filters,” in *IEEE International Conference on Acoustics, Speech and Signal Processing*. IEEE, 2015, pp. 1578–1582.

[39] P. Purkait, T.-J. Chin, A. Sadri, and D. Suter, “Clustering with hypergraphs: the case for large hyperedges,” *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 39, no. 9, pp. 1697–1711, 2017.

[40] D. Zhou, J. Huang, and B. Schölkopf, “Learning with hypergraphs: Clustering, classification, and embedding,” in *Advances in Neural Information Processing Systems*, 2007, pp. 1601–1608.

[41] M. Fazel, T. K. Pong, D. Sun, and P. Tseng, “Hankel matrix rank minimization with applications to system identification and realization,” *SIAM Journal on Matrix Analysis and Applications*, vol. 34, no. 3, pp. 946–977, 2013.

[42] J. Liu, J. Sun, and H.-Y. Shum, “Paint selection,” *ACM Transactions on Graphics*, vol. 28, no. 3, p. 69, 2009.

[43] F. Wang, C. Zhao, J. Liu, and H. Huang, “Normalized cut with adaptive similarity and spatial regularization,” *arXiv preprint arXiv:1806.01977*, 2018.

[44] X. Chen, F. Nie, J. Z. Huang, and M. Yang, “Scalable normalized cut with improved spectral rotation.” in *IJCAI*, 2017, pp. 1518–1524.

[45] M. Meil, “Comparing clusterings: an axiomatic view,” in *Proceedings of the 22nd International Conference on Machine Learning*. ACM, 2005, pp. 577–584.

[46] W. M. Rand, “Objective criteria for the evaluation of clustering methods,” *Journal of the American Statistical association*, vol. 66, no. 336, pp. 846–850, 1971.

Hu Zhu (M’17) received the B.S. degree in mathematics and applied mathematics from Huaibei Coal Industry Teachers College, Huaibei, China, in 2007, and the M.S. and Ph.D. degrees in computational mathematics and pattern recognition and intelligent systems from Huazhong University of Science and Technology, Wuhan, China, in 2009 and 2013, respectively. In 2013, he joined the Nanjing University of Posts and Telecommunications, Nanjing, China. His research interests include pattern recognition, image processing, and computer vision.

Jieke Zhang received the B.S. degree in communication engineering from Qufu Normal University in 2017. Currently, he is pursuing his master degree in electronics and communications engineering at Nanjing University of Posts and Telecommunications. His research interest is image processing.

Guoxia Xu (M’19) received the B.S. degree in information and computer science from Yancheng Teachers University, Jiangsu Yancheng, China in 2015, and the M.S. degree in computer science and technology from Hohai University, Nanjing, China in 2018. He was a research assistant in City university of Hong Kong and Chinese University of Hong Kong. Now, he is pursing his Ph.D. degree in Department of Computer Science, Norwegian University of Science and Technology, Gjovik Norway. His research interest includes pattern recognition, image processing, and computer vision.

Lizhen Deng (M’17) received the B.S. degree in electronic information science and technology from Huaibei Coal Industry Teachers College, Huaibei, China, in 2007, and the M.S. degree in communication and information systems from Nanjing University of Aeronautics and Astronautics, Nanjing, China, in 2010. She received her Ph.D. degree in electrical engineering from Huazhong University of Science and Technology, China, in 2014. In 2014, she joined the Nanjing University of Posts and Telecommunications, Nanjing, China. Her current research interests include image processing, computer vision, pattern recognition, and spectral data processing.