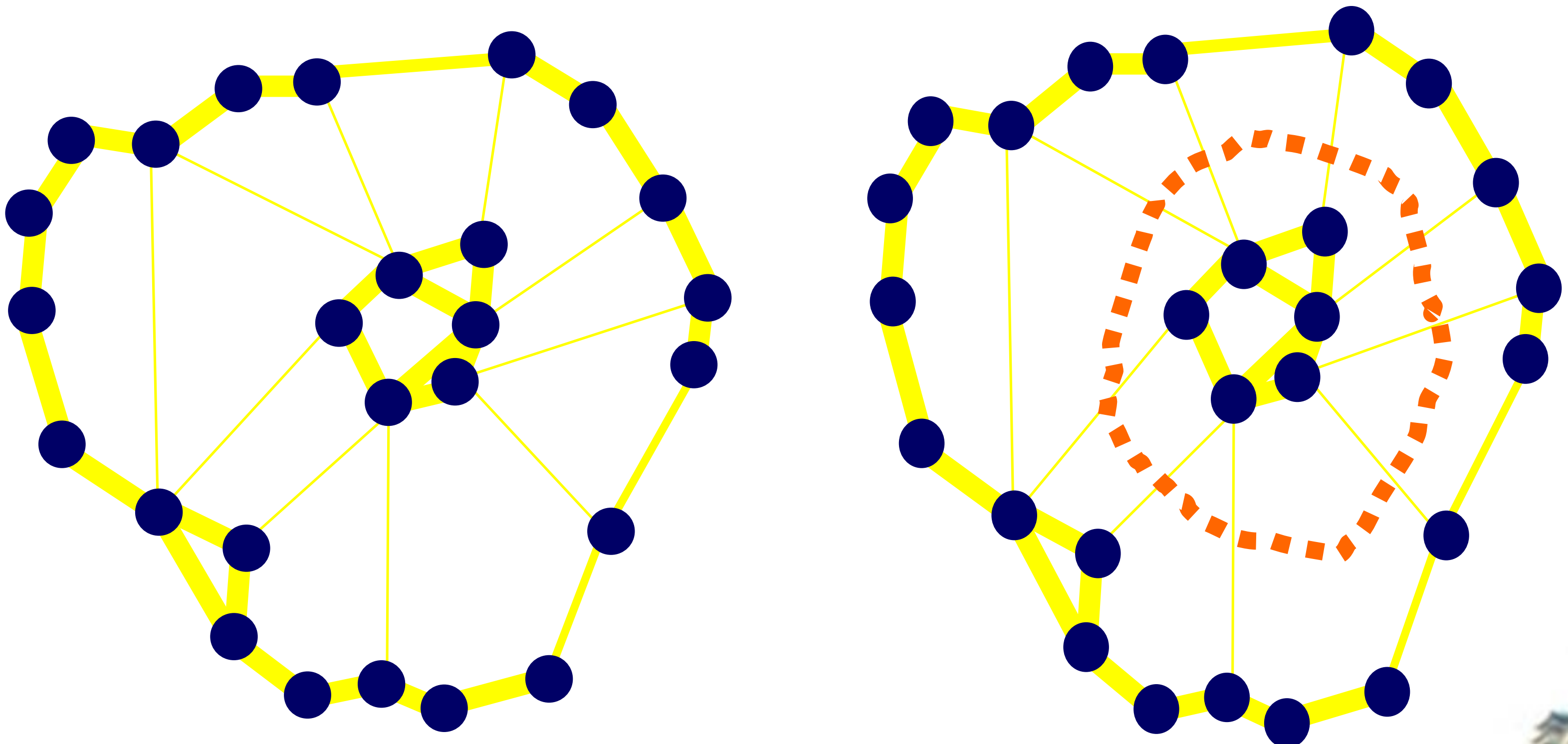
数字图像处理
Digital Image Processing

自然图像分割
Color Image Segmentation

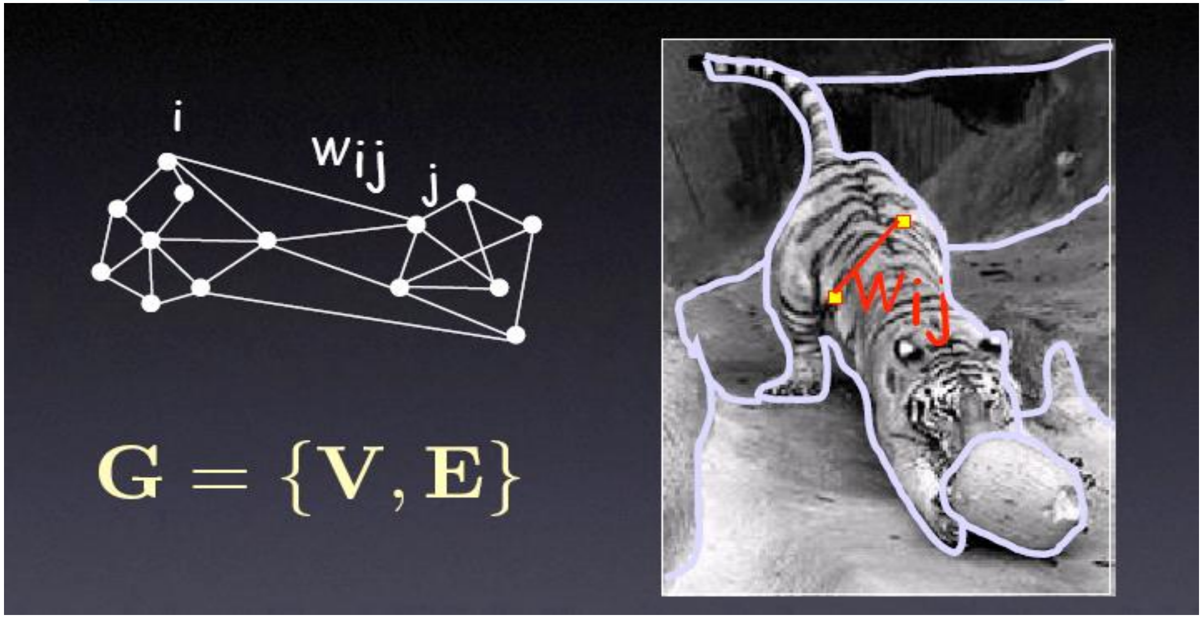
彩色图像分割及处理

1. 分水岭算法
2. Mean shift分割
3. Normalized cuts(Ncuts)分割
4. Ncuts分割改进算法
5. Graph cuts(GC) 优化
6. GC与交互式分割
7. Graph cut与变分模型

图的划分



- 将节点间的关系采用带权图来表达
- 将图划分成两个部分或多个部分



如果将图像中的每个象素看作一个节点，每对节点均用一条边连接起来，边的权值反映这两个象素之间的相似性，那么我们就可以构建一个带权的无向图G=(V,E) 。利用象素的灰度值以及它们的空间位置，可以定义图G中连接两个节点*u*和*v*的边的权值

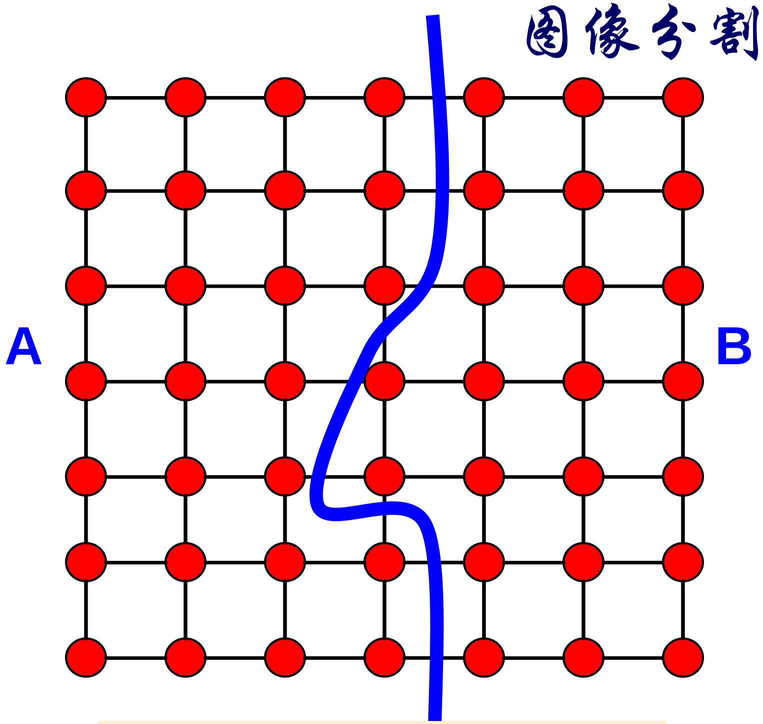
$$w(u,v)=\begin{cases} e^{-\left[\frac{\left\|F(u)-F(v)\right\|_2^2}{d_I}+\frac{\left\|X(u)-X(v)\right\|_2^2}{d_X}\right]} & \text{if } \left\|X(u)-X(v)\right\|_2 < r \\ 0 & \text{otherwise} \end{cases}$$

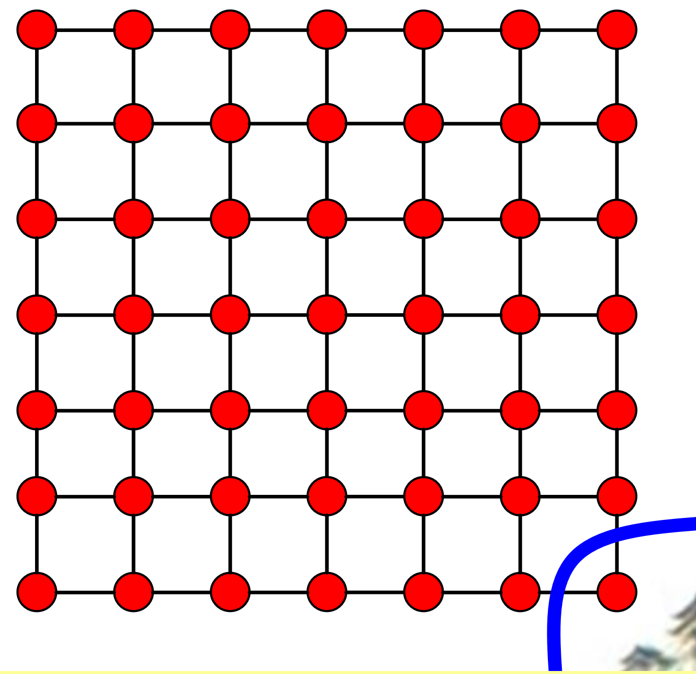
Segmentation=Graph Partition

图割(Graph Cuts)优化算法

图像 — 图

图像分割 — 图划分





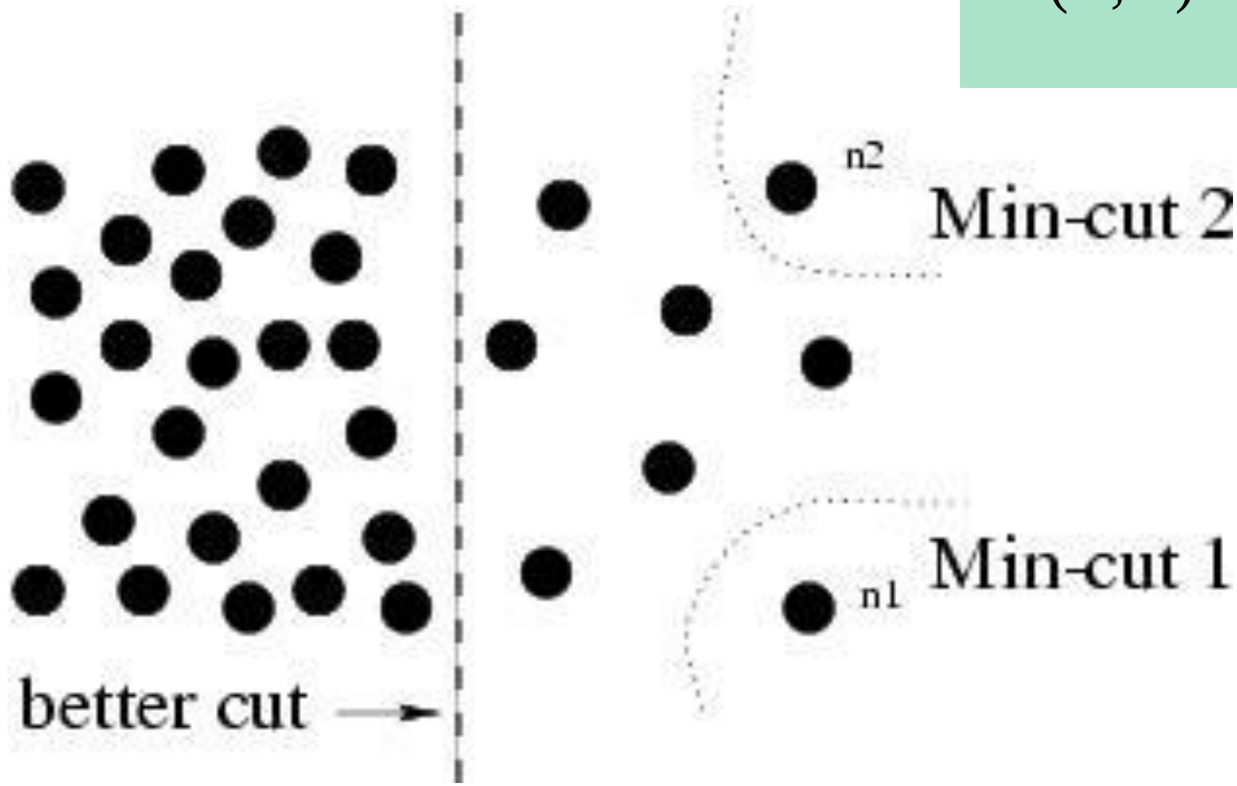
$$c(A,B)=\sum_{u\in A}\sum_{v\in B}c(u,v)$$

Minimum Cut 容易产生孤立点

- 图的割集是与切割的边的数量及权值相关的
- 一个割集 **cut(A, B)** 将图分为独立的两个部分

Disassociation Measures

$$cut(A,B) = \sum_{u \in A, v \in B} w(u,v)$$



- Minimizing the cut will give a partition with the maximum disassociation.
- However, this measure favors cutting to small sets of isolated nodes.

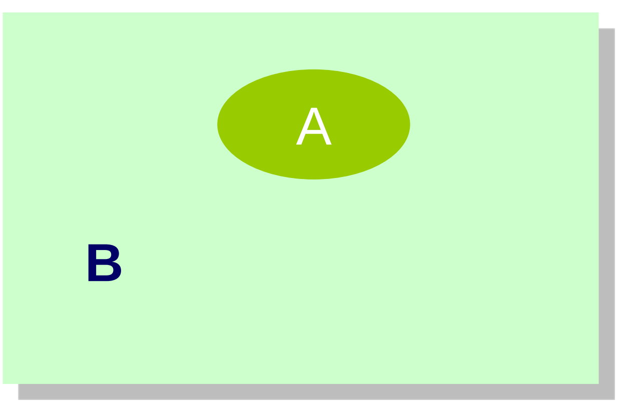
Disassociation Measures

- Normalized cut Ncut(A,B) measures similarity between two groups, normalized by the “volume” they occupy in the whole graph [Shi and Malik, 2000].

- It is more appropriate to measure the disassociation between groups A and B.

minimize

$$Ncut(A,B)=\frac{cut(A,B)}{asso(A,V)}+\frac{cut(A,B)}{asso(B,V)}$$

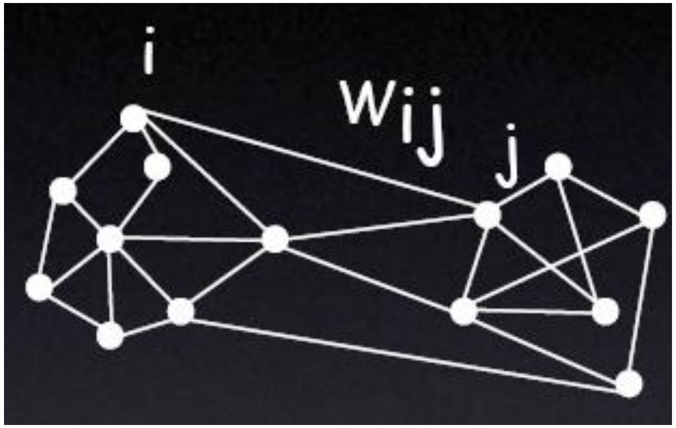


A + B = V

$$asso(A,V)=asso(A,A)+cut(A,B)$$

$$asso(B,V)=asso(B,B)+cut(A,B)$$

Graph-based Image Segmentation



G = {V,E}

V: graph nodes
E: edges connection nodes

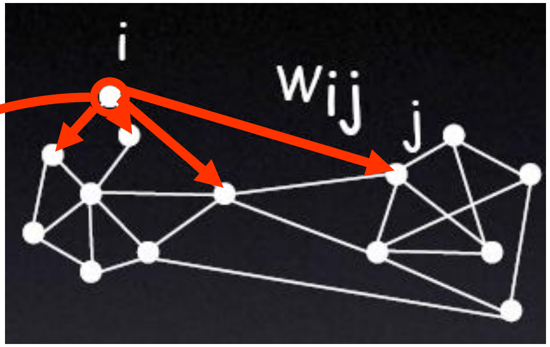
Pixels
Pixel similarity

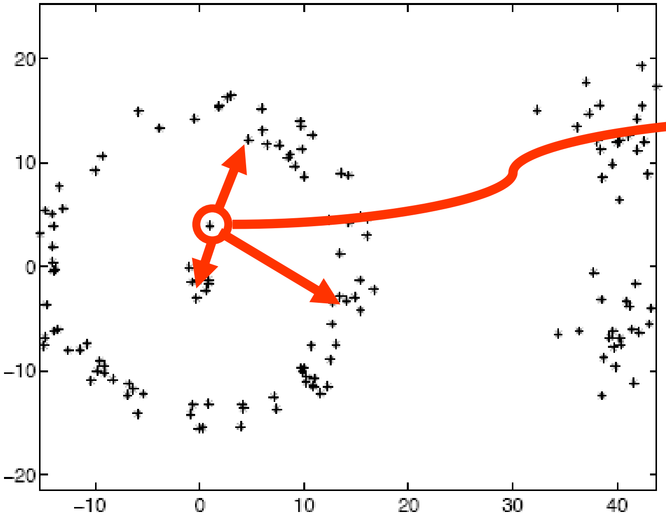
Slides from Jianbo Shi

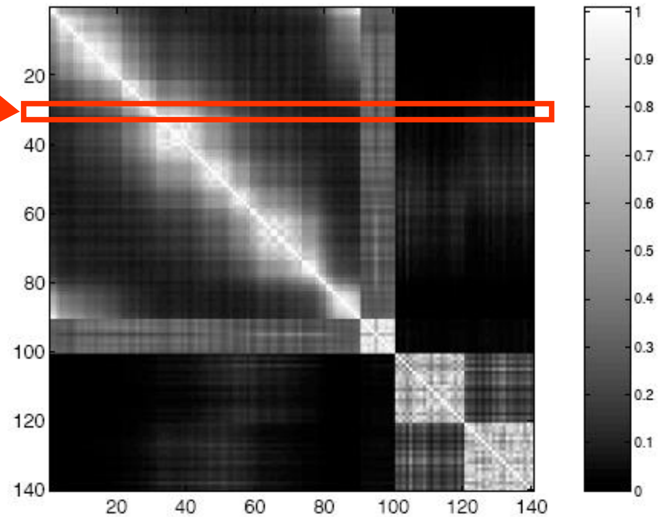
Graph terminology

- Similarity matrix: $W = \left[w_{i,j} \right]$

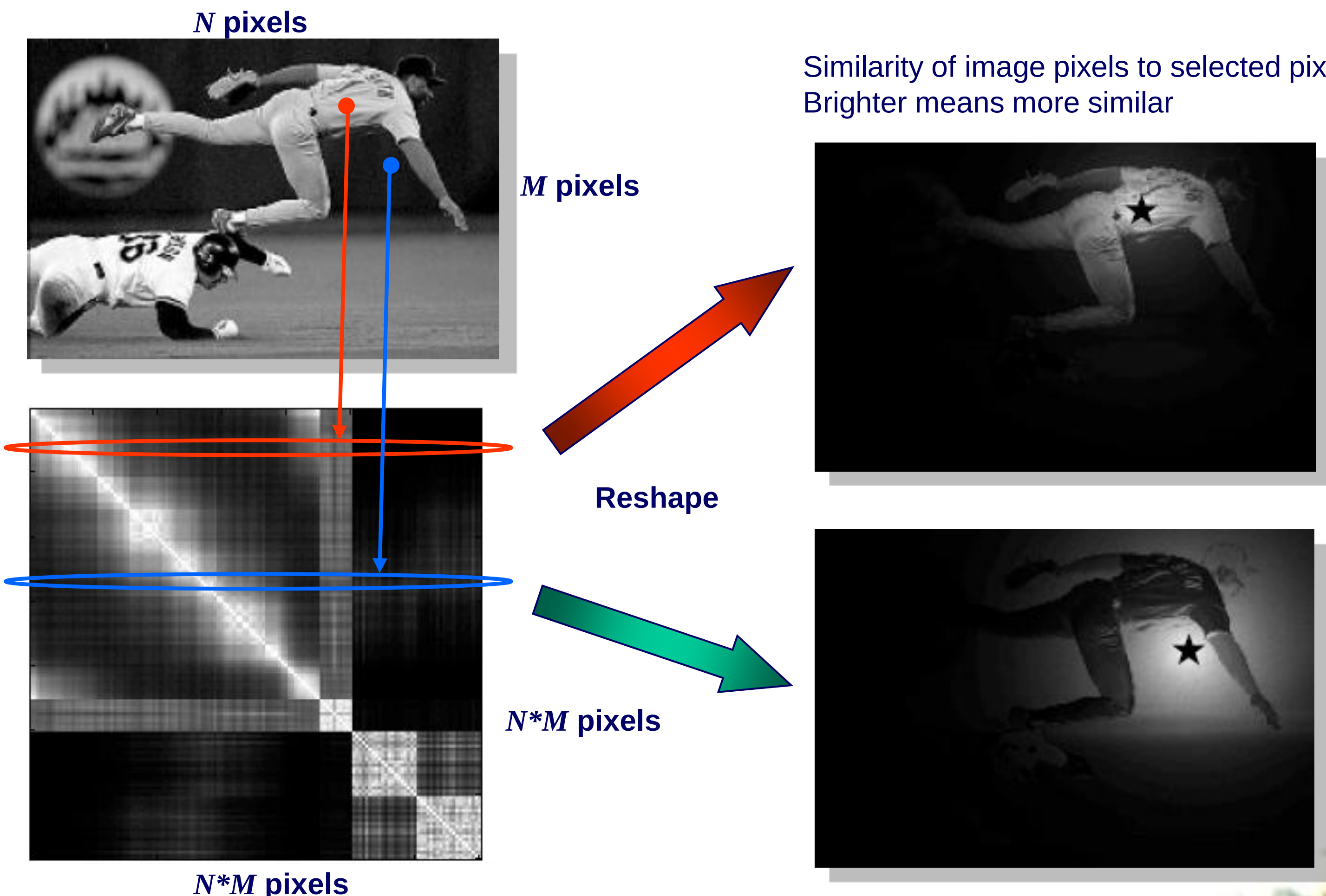
$$w_{i,j} = e^{\frac{-\left\| X_{(i)} - X_{(j)} \right\|_2^2}{\sigma_X^2}}$$







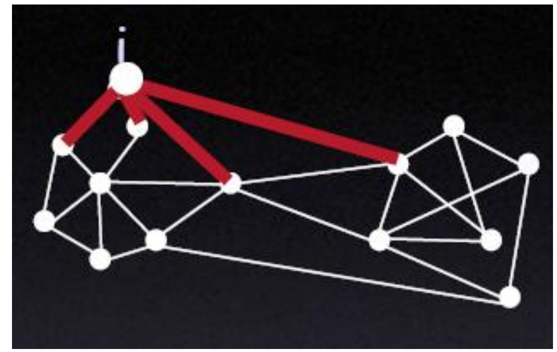
Affinity matrix

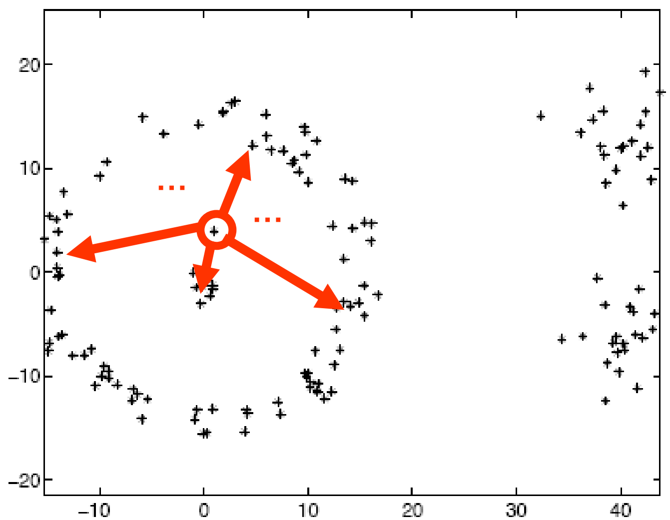


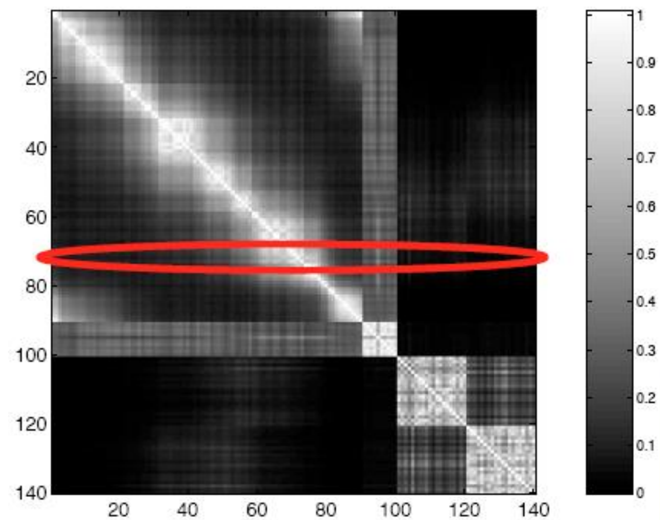
Graph terminology

- Degree of node:

$$d_i = \sum_j w_{i,j}$$



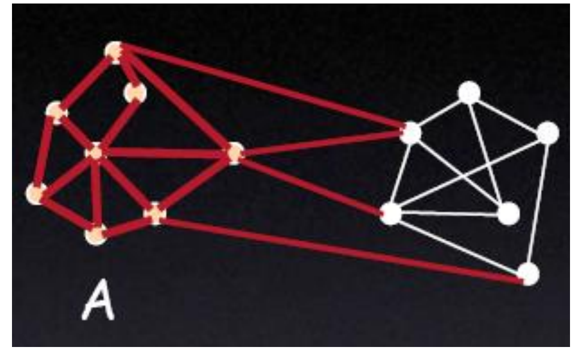


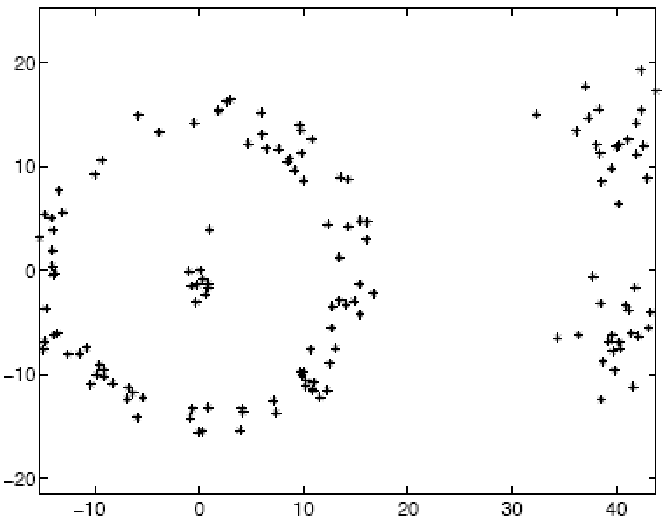


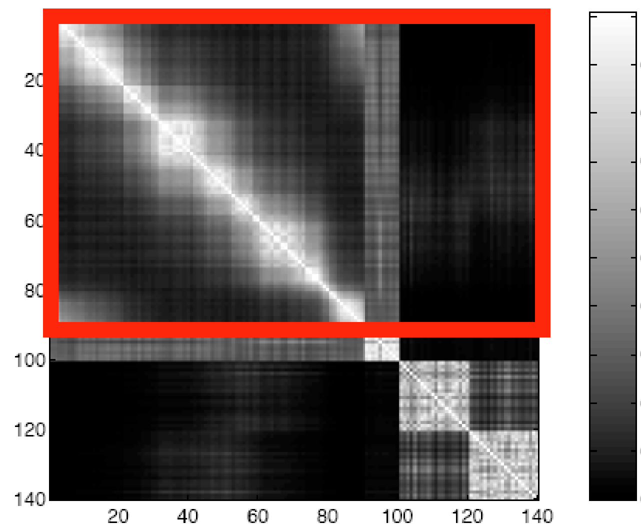
Graph terminology

- Volume of set:

$$vol(A) = \sum_{i \in A} d_i, A \subseteq V$$





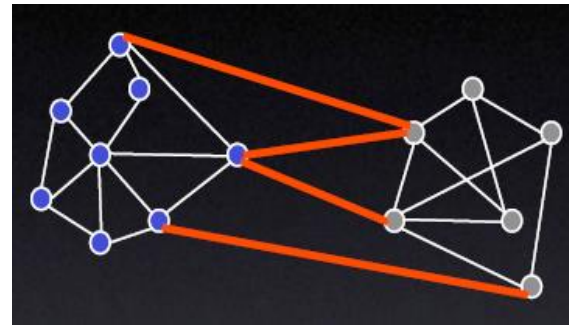


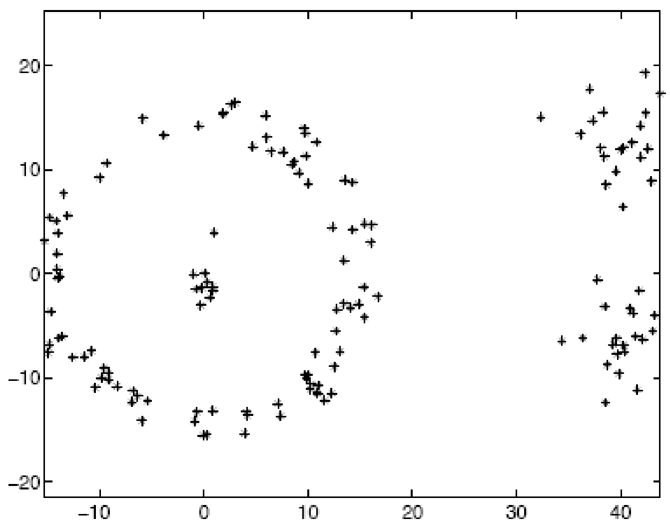
总计140个点，假定1-95为集合A，96-140为集合B

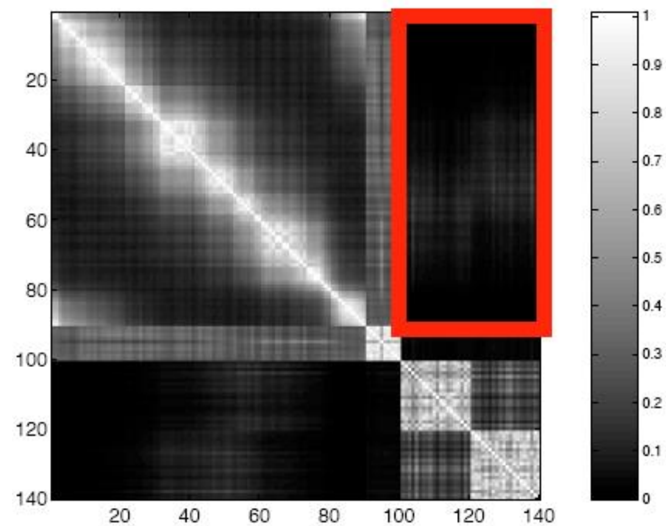
Graph terminology

- Cuts in a graph:

$$cut(A,\bar{A}) = \sum_{i\in A,\, j\in \bar{A}} w_{i,j}$$

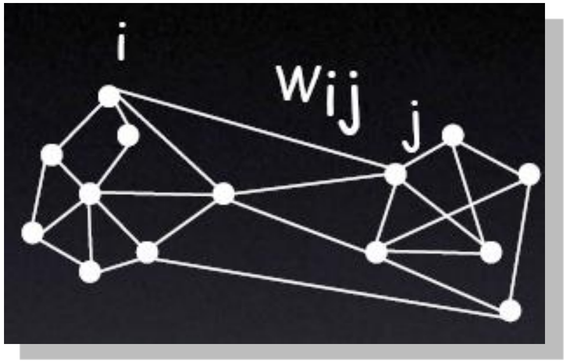




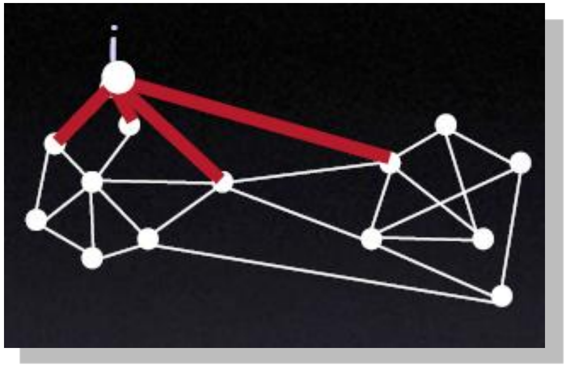


Graph terminology

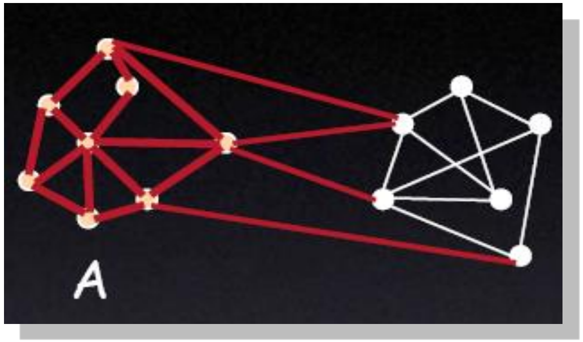
Similarity matrix: $W = \left[w_{i,j} \right]$



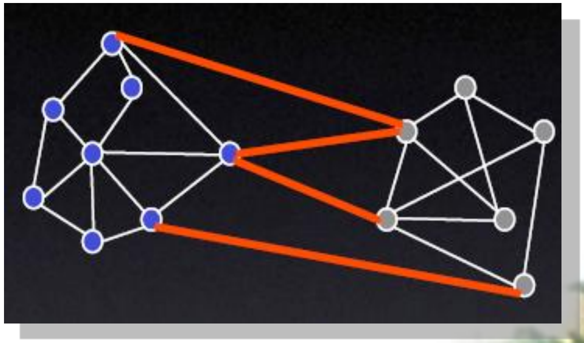
Degree of node: $d_i = \sum w_{i,j}$



Volume of set:



Graph cuts:

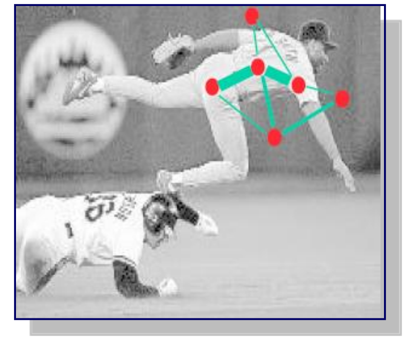


Representation

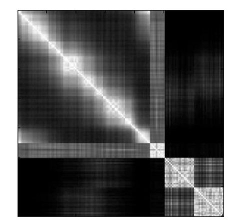
Partition matrix *X*:

$$X = [X_1, \ldots, X_K]$$

$$X = \begin{array}{cc} & \text{segments} \\ & \begin{bmatrix} 1 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix} \text{pixels} \end{array}$$



Pair-wise similarity matrix *W*: $W(i, j) = aff(i, j)$



Degree matrix *D*: $D(i,i) = \sum_j w_{i,j}$

Laplacian matrix *L*: $L = D - W$

Pixel similarity functions

Intensity $$W(i,j)=e^{\frac{-\left\|I_{(i)}-I_{(j)}\right\|_2^2}{\sigma_I^2}}$$

Distance $$W(i,j)=e^{\frac{-\left\|X_{(i)}-X_{(j)}\right\|_2^2}{\sigma_X^2}}$$

Texture $$W(i,j)=e^{\frac{-\left\|c_{(i)}-c_{(j)}\right\|_2^2}{\sigma_c^2}}$$

$$w(u,v)=\begin{cases} e^{-\left[\frac{\left\|F(u)-F(v)\right\|_2^2}{d_I}+\frac{\left\|X(u)-X(v)\right\|_2^2}{d_X}\right]} & \text{if } \left\|X(u)-X(v)\right\|_2<r \\ 0 & \text{otherwise} \end{cases}$$

Disassociation Measures

- The solution to the problem

minimize
$$Ncut(A,B)=\frac{cut(A,B)}{asso(A,V)}+\frac{cut(A,B)}{asso(B,V)}$$

is given by the following eigen-system

$$\mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-\frac{1}{2}}z=\lambda z$$

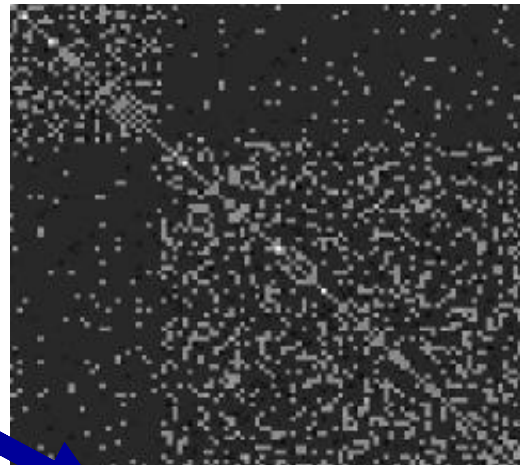
$$W_{ij}=W(x_i,x_j)$$

$$D_{ii}=\sum_j W_{ij}$$

- For an image with N pixels, the matrix size is NxN.
- Computational cost increases dramatically as the image size increases!

Graph-based Image Segmentation

Image (I)

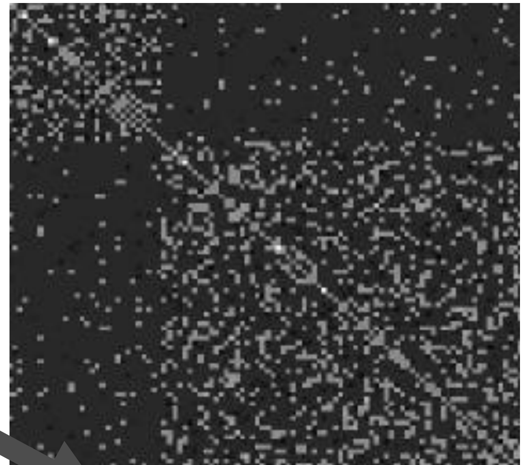


Graph Affinities
(W)

Intensity
Color
Edges
Texture

Graph-based Image Segmentation

Image (I)



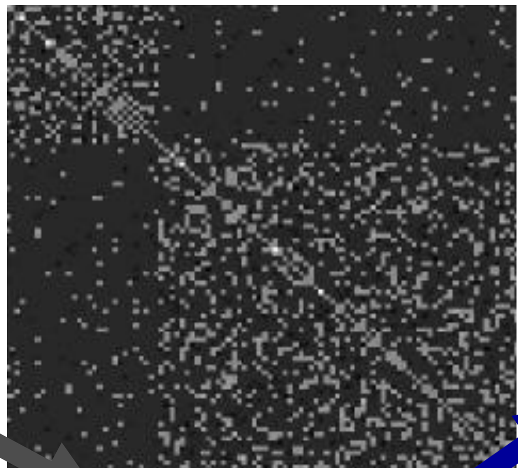
Graph Affinities
(W)

Intensity
Color
Edges
Texture

$$Ncut(A,B) = cut(A,B)\left(\frac{1}{vol(A)} + \frac{1}{vol(B)}\right)$$

Graph-based Image Segmentation

Image (I)



Graph Affinities (W)

Eigenvector X(W)

Intensity
Color
Edges
Texture

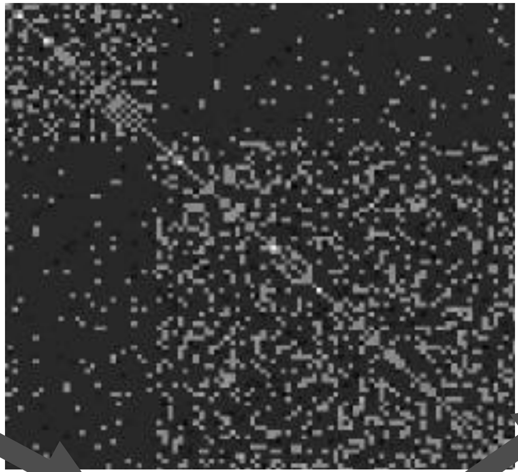
$$Ncut(A,B) = cut(A,B)\left(\frac{1}{vol(A)} + \frac{1}{vol(B)}\right)$$

$$(D-W)X = \lambda DX$$

$$X_A(i) = \begin{cases} 1 & if \quad i \in A \\ 0 & if \quad i \notin A \end{cases}$$

Graph-based Image Segmentation

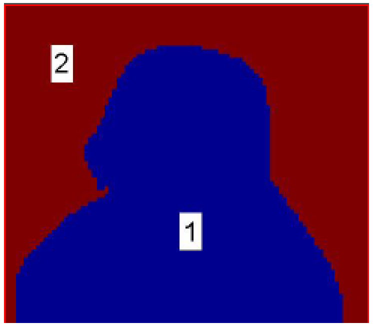
Image (I)



Graph Affinities
(W)

Intensity
Color
Edges
Texture

Eigenvector
X(W)



Discretization

$$Ncut(A,B) = cut(A,B)\left(\frac{1}{vol(A)} + \frac{1}{vol(B)}\right)$$

$$(D-W)X = \lambda DX$$

$$X_A(i) = \begin{cases} 1 & if \quad i \in A \\ 0 & if \quad i \notin A \end{cases}$$

彩色图像分割及处理

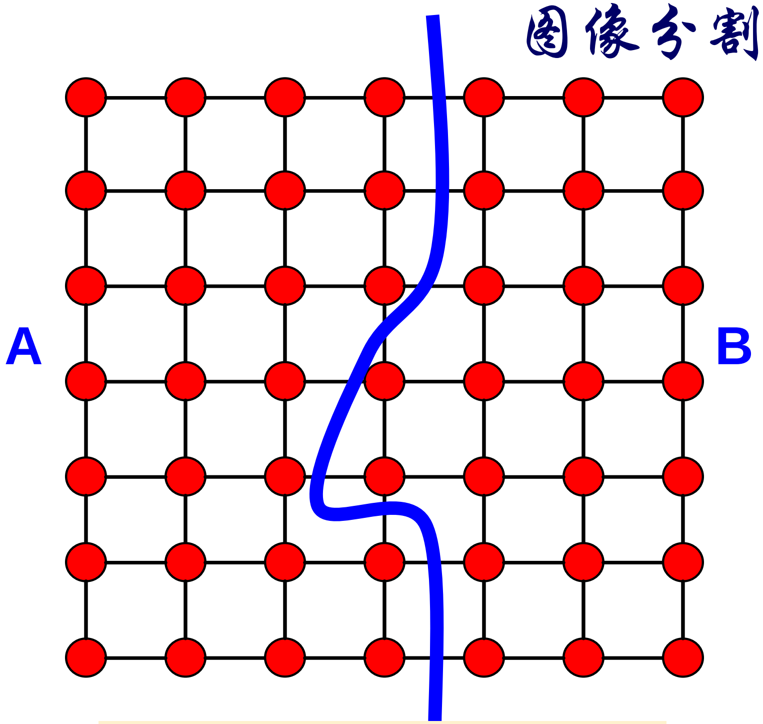
1. 分水岭算法
2. Mean shift (MS) 分割
3. Normalized cuts(Ncuts)分割
4. Ncuts分割改进算法
5. Graph cuts(GC) 优化
6. GC与交互式分割
7. Graph cut与变分模型

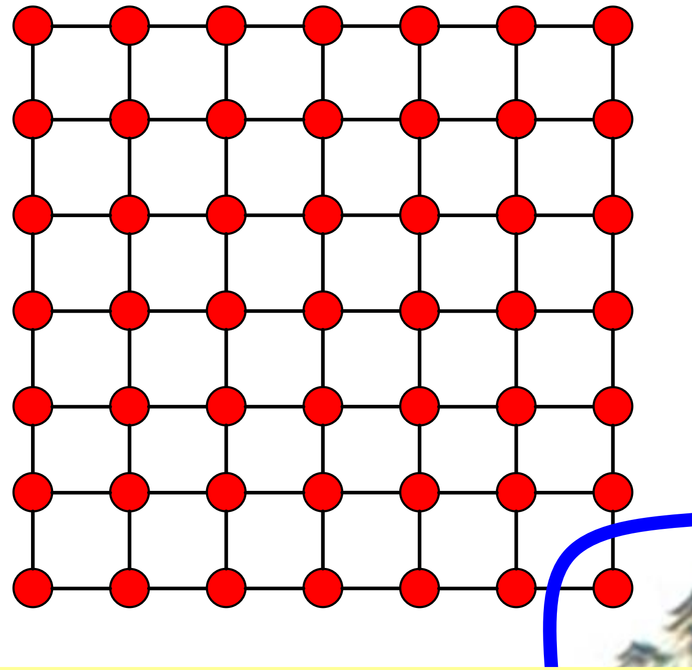
改进1——基于Ncuts的阈值分割

Wenbing Tao, et.al, "Image Thresholding Using Graph Cuts", *IEEE Transactions on Systems Man and Cybernetics Part A-Systems and Humans*. 2008.

图像 — 图

图像分割 — 图划分





$$c(A,B)=\sum_{u\in A}\sum_{v\in B}c(u,v)$$

Minimum Cut 容易产生孤立点

- 图的割集是与切割的边的数量及权值相关的
- 一个割集 **cut(A, B)** 将图分为独立的两个部分

Disassociation Measures

- The solution to the problem

minimize $$Ncut(A,B) = \frac{cut(A,B)}{asso(A,V)} + \frac{cut(A,B)}{asso(B,V)}$$

is given by the following eigen-system

$$\mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-\frac{1}{2}}z = \lambda z$$

$$W_{ij} = W(x_i, x_j)$$

$$D_{ii} = \sum_j W_{ij}$$

- For an image with N pixels, the matrix size is NxN.
- Computational cost increases dramatically as the image size increases!

■正确的划分函数

■有效的优化算法

$$\boxed{Ncut(A,B)} = \frac{cut(A,B)}{asso(A,V)} + \frac{cut(A,B)}{asso(B,V)}$$

Jianbo Shi

最小化

NP难问题

$$\mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-\frac{1}{2}}z = \lambda z$$

计算 Laplacian 矩阵 D-W的特征矢量

矩阵D、W和D-W的维数为图像中象素的个数

问题：维数太高，效率低下？

基于图划分的阈值法基本原理

$$Ncut(A,B)=\frac{cut(A,B)}{asso(A,V)}+\frac{cut(A,B)}{asso(B,V)}$$

对每一个设定阈值T(0≦T≦255)计算*Ncut*(*A*, *B*)
最小的*Ncut*对应的阈值T为最佳阈值

计算高维权值矩阵耗时
权值矩阵太大无法存储

影响阈值方法的效率和实现

Proposed Approach

- Consider V_k, k = 0, …, 255 corresponds to the gray scale levels.

$$V_k = \{(x,y): f(x,y)=k, (x,y)\in V\}, k \in L$$

$$cut(V_i,V_j) = \sum_{u\in V_i, v\in V_j} w(u,v)$$

$$A = \bigcup_{k=0}^{t} V_k \qquad B = \bigcup_{k=t+1}^{255} V_k$$

$$cut(A,B) = \sum_{u\in A, v\in B} w(u,v) = \sum_{u\in A}\Big[\sum_{v\in B} w(u,v)\Big]$$
$$= \sum_{i=0}^{t}\sum_{u\in V_i}\Big[\sum_{j=t+1}^{255}\sum_{v\in V_j} w(u,v)\Big] = \sum_{i=0}^{t}\sum_{j=t+1}^{255}\Big[\sum_{u\in V_i, v\in V_j} w(u,v)\Big] = \sum_{i=0}^{t}\sum_{j=t+1}^{255}[cut(V_i,V_j)]$$

$$asso(A,A) = \sum_{u\in A, v\in A} w(u,v) = \sum_{i=0}^{t}\sum_{j=i}^{t}\Big[\sum_{u\in V_i, v\in V_j} w(u,v)\Big] = \sum_{i=0}^{t}\sum_{j=i}^{t}[cut(V_i,V_j)]$$

$$asso(B,B) = \sum_{u\in B, v\in B} w(u,v) = \sum_{i=t+1}^{255}\sum_{j=i}^{255}\Big[\sum_{u\in V_i, v\in V_j} w(u,v)\Big] = \sum_{i=t+1}^{255}\sum_{j=i}^{255}[cut(V_i,V_j)]$$

Proposed Approach

$$\begin{array}{cccccc|ccccc}
m_{0,0} & m_{0,1} & \cdots & m_{0,j} & \cdots & m_{0,t} & m_{0,t+1} & \cdots & & m_{0,254} & m_{0,255} \\
 & m_{1,1} & & & & \vdots & \vdots & & & & m_{1,255} \\
 & & \ddots & asso(A,A) & & & & cut(A,B) & & & \vdots \\
 & & & m_{i,j} & & & & & & & m_{i,255} \\
 & & & & \ddots & & & & & & \vdots \\
 & & & & & m_{t,t} & m_{t,t+1} & \cdots & \cdots & & m_{t,255} \\
\hline
 & & & & & & m_{t+1,t+1} & \cdots & \cdots & & m_{t+1,255} \\
 & & & & & & & \ddots & asso(B,B) & & \vdots \\
 & & & & & & & & & m_{254,254} & m_{254,255} \\
 & & & & & & & & & & m_{255,255}
\end{array}$$

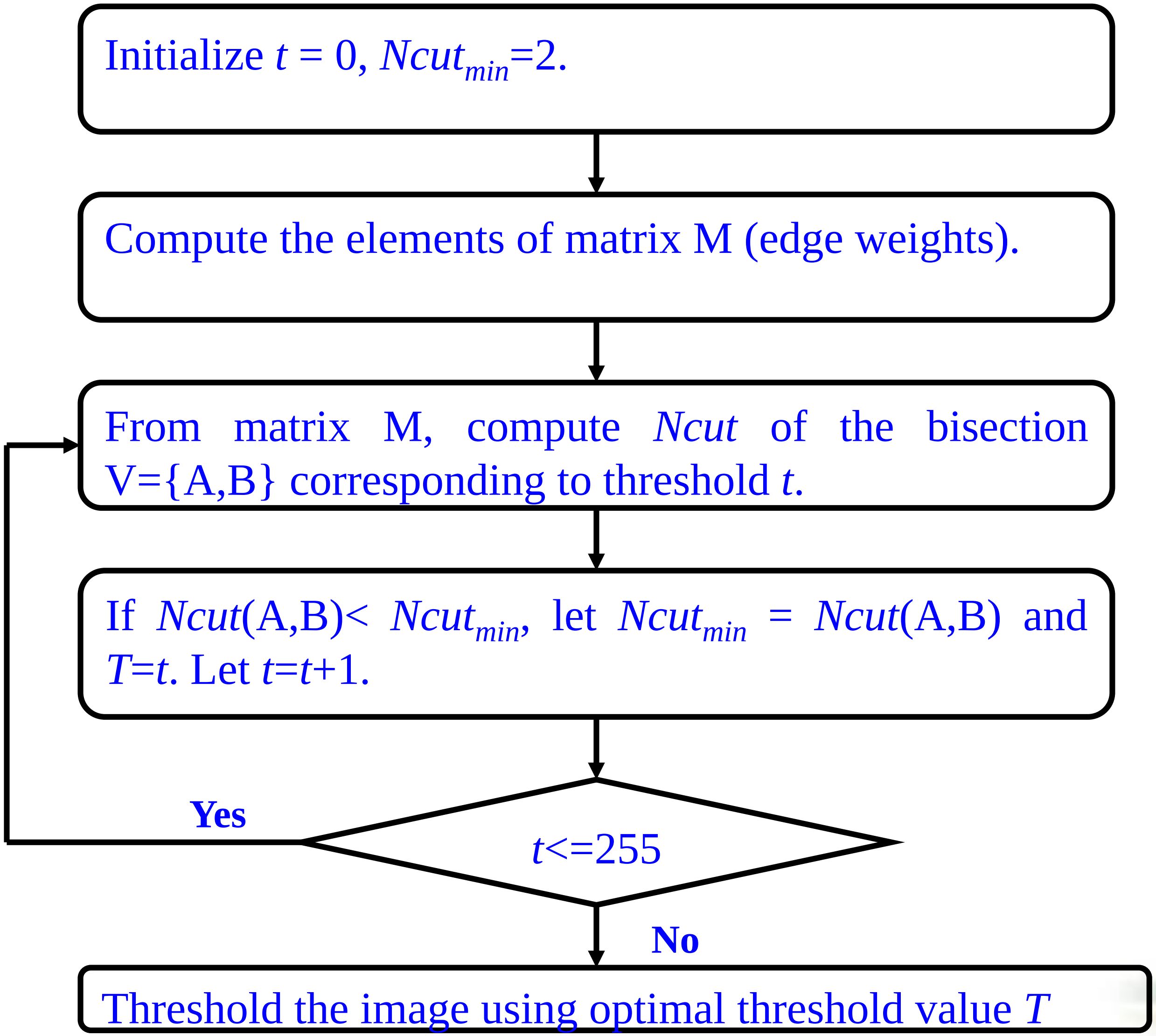
$$cut(V_i,V_j)=\sum_{u\in V_i,v\in V_j} w(u,v)$$

$$asso(A,V)=asso(A,A)+cut(A,B)$$

$$asso(B,V)=asso(B,B)+cut(A,B)$$

$$Ncut(A,B)=\frac{cut(A,B)}{asso(A,A)+cut(A,B)}+\frac{cut(A,B)}{asso(B,B)+cut(A,B)}$$

Proposed Approach



Advantages of the Proposed Approach

- Low computational cost
- Suited for real-time vision processing
- Provide superior and robust image thresholding performance

Experimental Results

Compared methods

- Pikaz
- Kittler
- Kapur
- Yanowitz
- Ramesh
- Pal

Test images

- Infrared object images
- Standard test images

Experimental Results

- **Intruder – infrared image: 185 x 141**

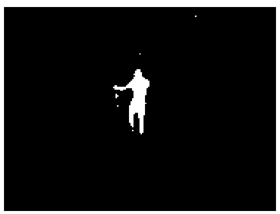
original

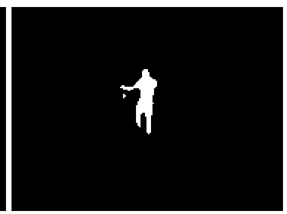
propose
d

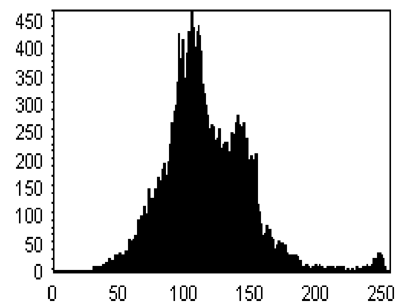
manua
l

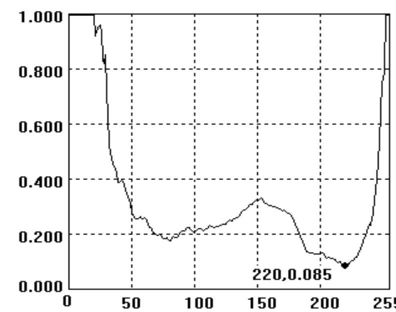
histogram

Ncut

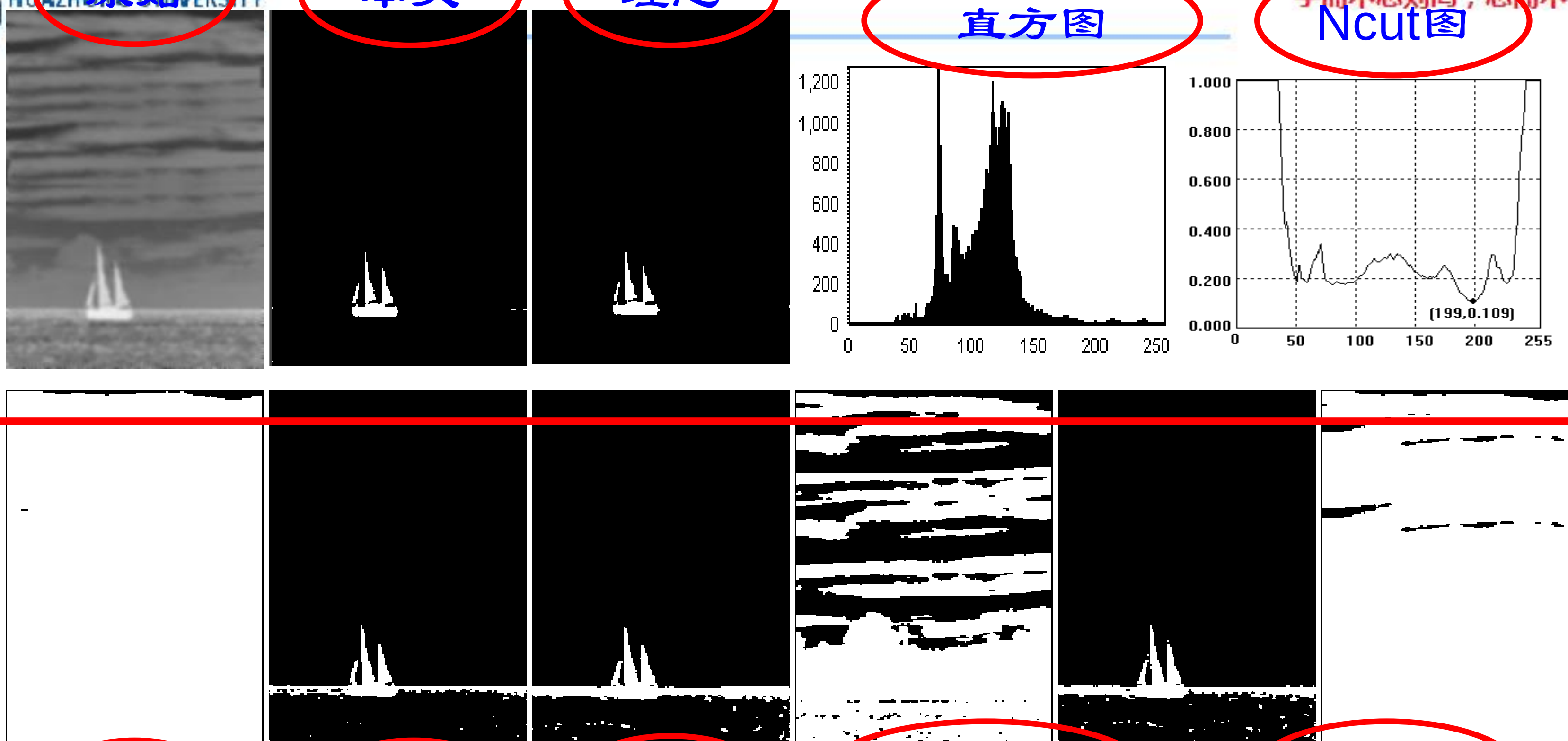


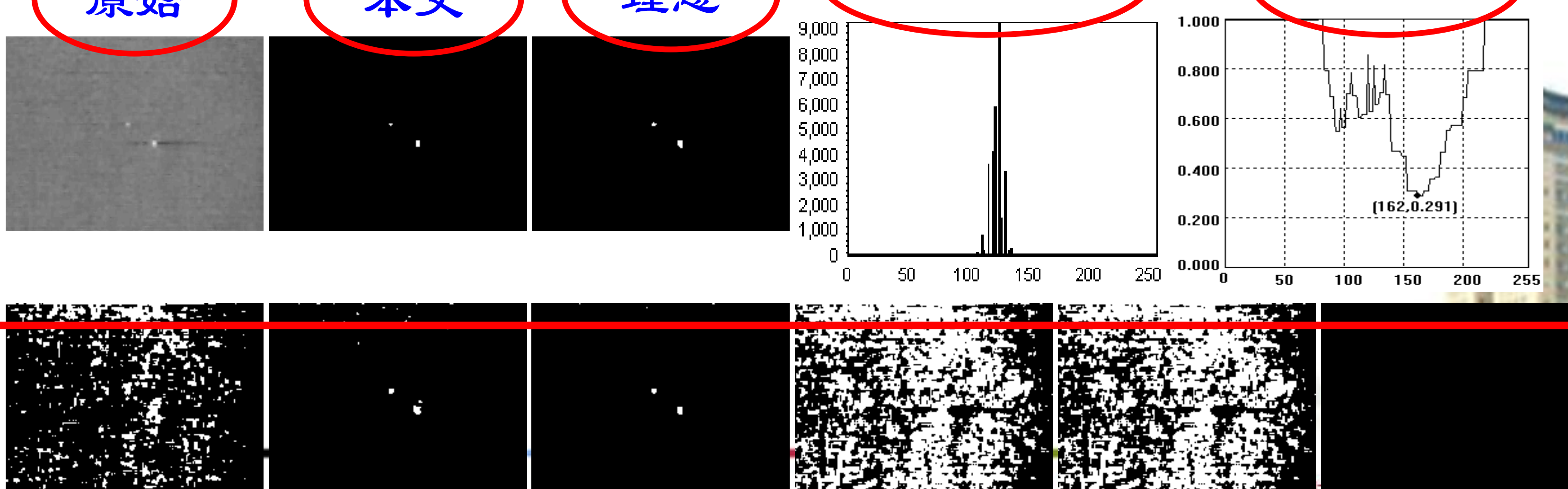


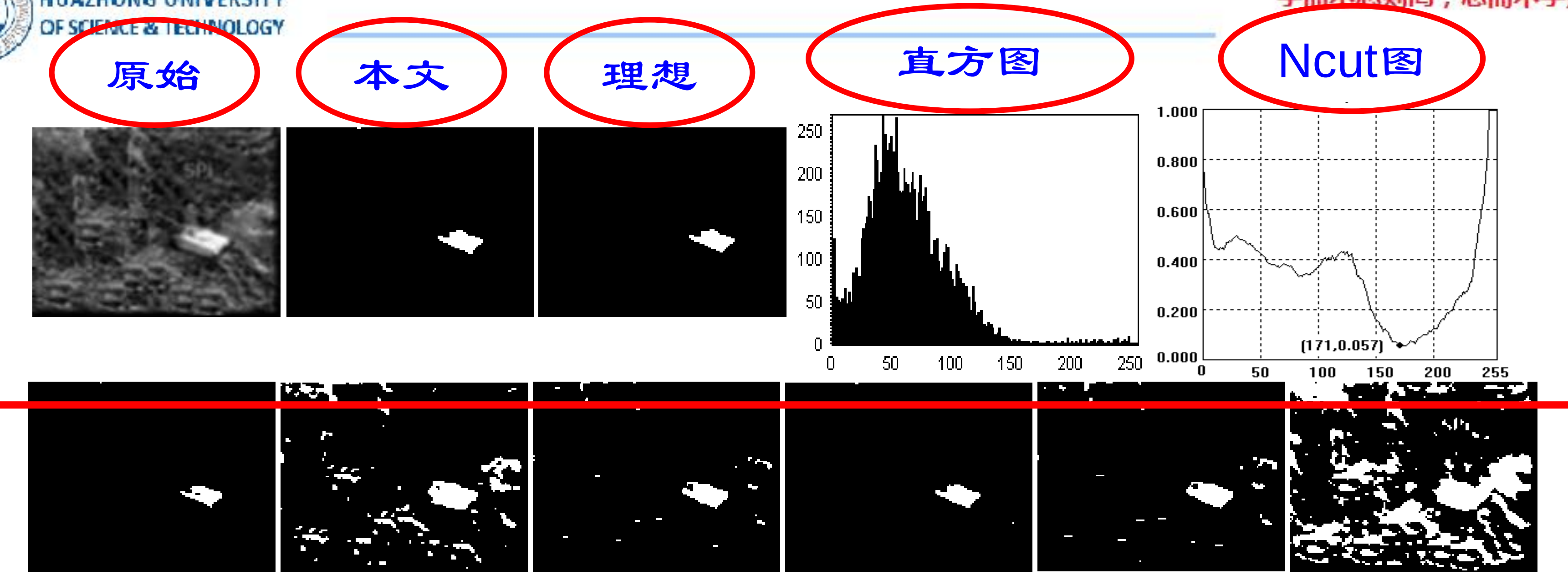


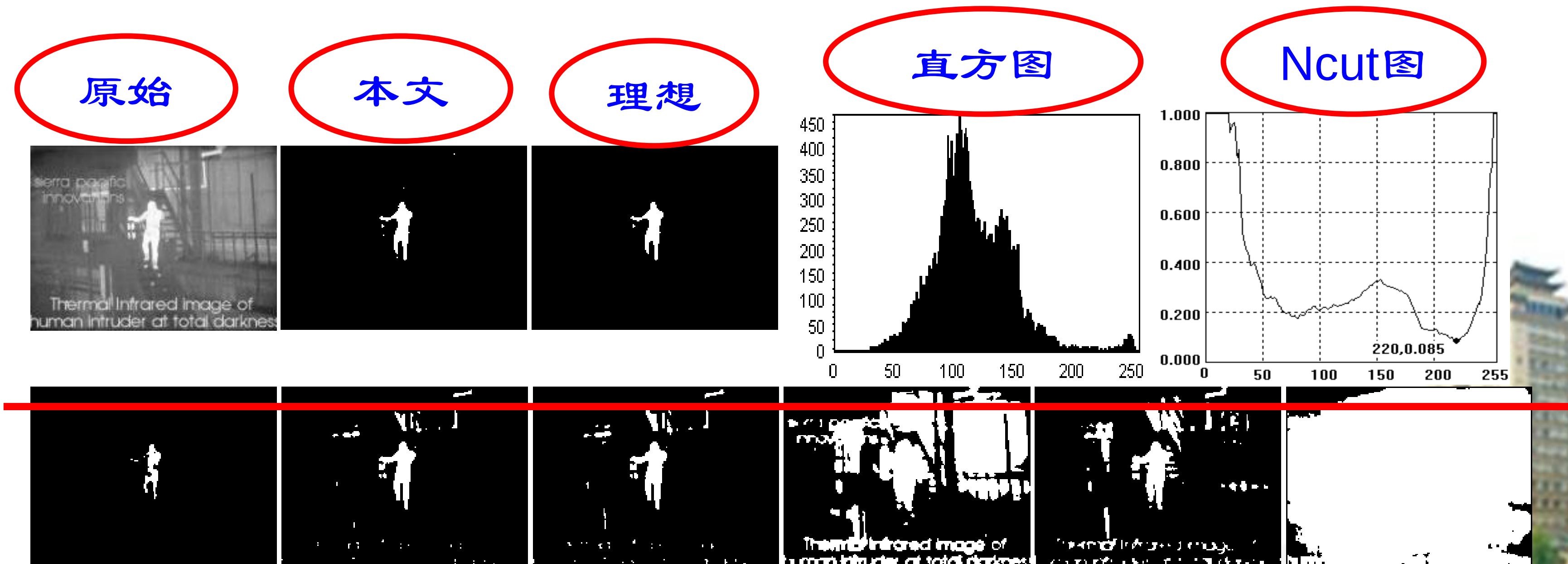


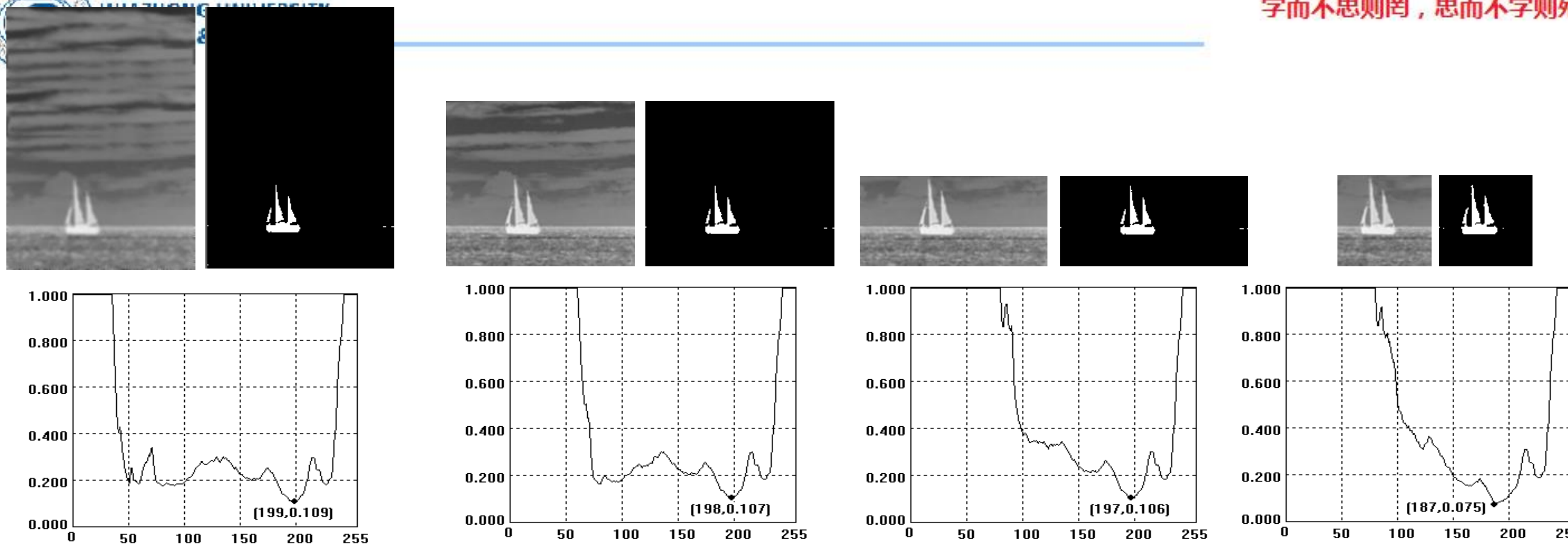
Other methods

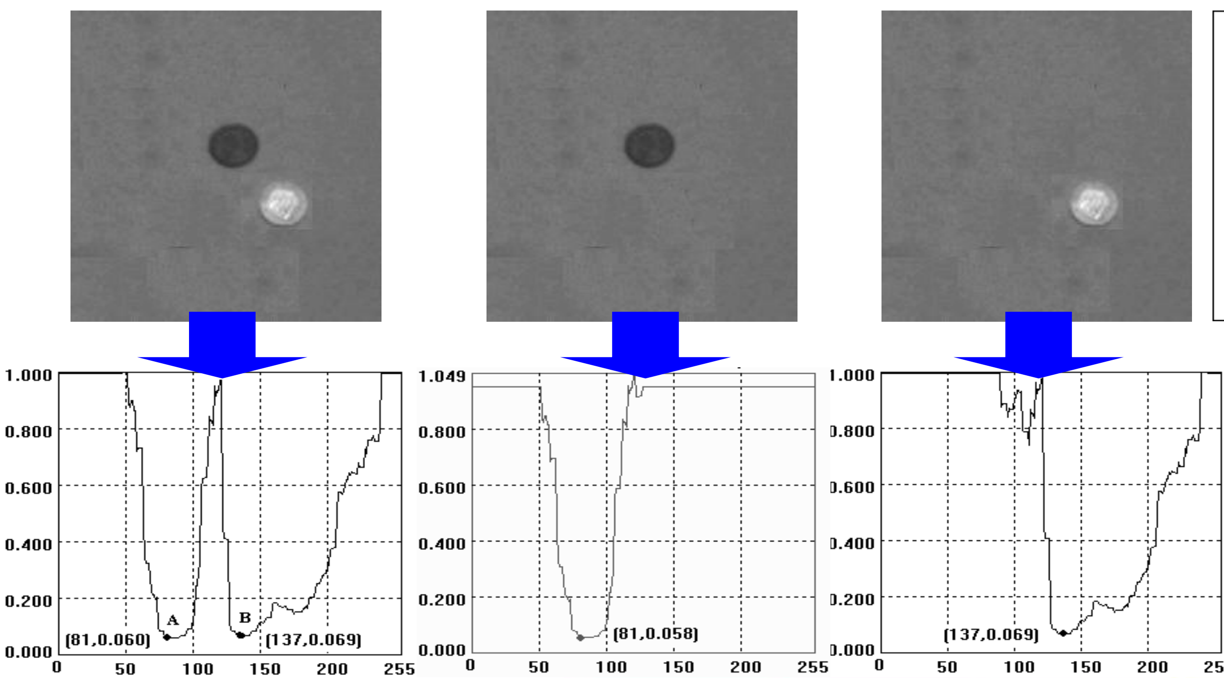

原始
本文
理想
直方图
Ncut图
1,200
1,000
800
600
400
200
0
0 50 100 150 200 250
1.000
0.800
0.600
0.400
0.200
0.000
0 50 100 150 200 255
(199,0.109)

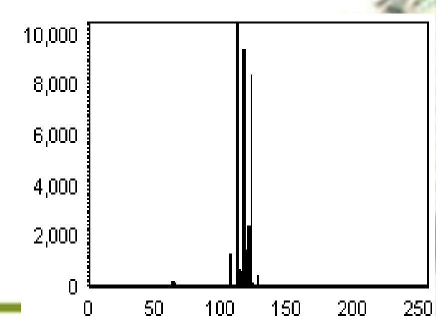

原始
本文
理想
直方图
Ncut图
9,000
8,000
7,000
6,000
5,000
4,000
3,000
2,000
1,000
0
0 50 100 150 200 250
1.000
0.800
0.600
0.400
0.200
0.000
0 50 100 150 200 255
(162,0.291)


原始
本文
理想
直方图
Ncut图
(171,0.057)


原始
本文
理想
直方图
Ncut图
Thermal Infrared image of
human intruder at total darkness
220,0.085


(199,0.109)
(198,0.107)
(197,0.106)
(187,0.075)
1.000
0.800
0.600
0.400
0.200
0.000
0
50
100
150
200
255


A
B
(81,0.060)
(137,0.069)
(81,0.058)
(137,0.069)
1.049
1.000
0.800
0.600
0.400
0.200
0.000
0
50
100
150
200
255


10,000
8,000
6,000
4,000
2,000
0
0
50
100
150
200
250

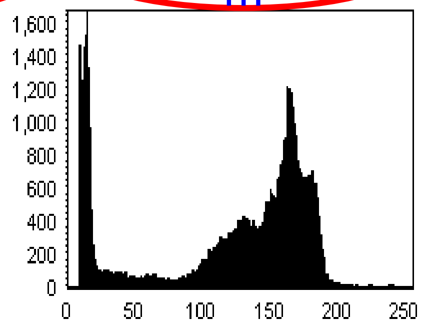
Experimental Results

- **Good results for standard test images**

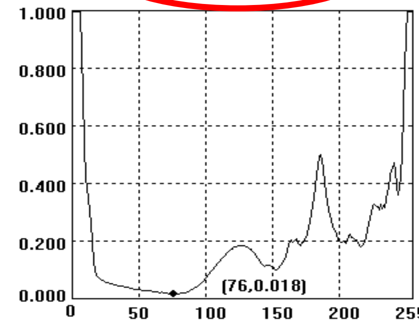
original

proposed

histogram



Ncut



Other methods

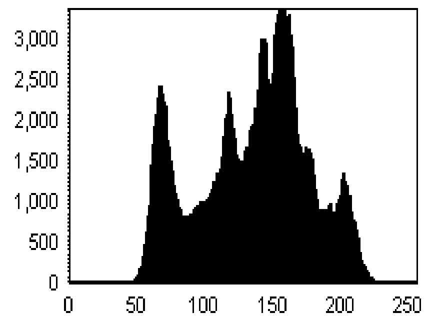
Experimental Results

- **Good results for standard test images**

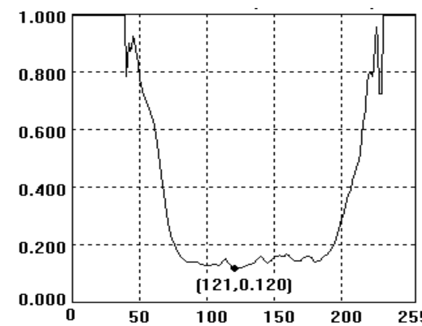
original

proposed

histogram



Ncut



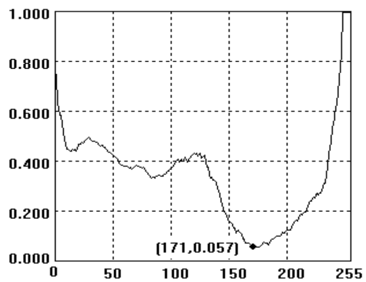
Other methods

Experimental Results

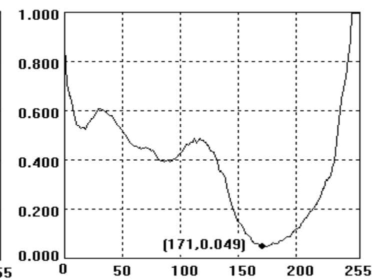
$$w(u,v)=\begin{cases} e^{-\left[\frac{\|F(u)-F(v)\|_2^2}{d_I}+\frac{\|X(u)-X(v)\|_2^2}{d_X}\right]} & \text{if } \|X(u)-X(v)\|_2<r \\ 0 & \text{otherwise} \end{cases}$$

Tank image

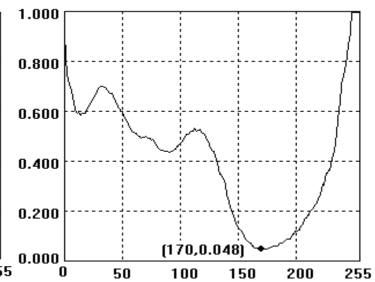
- As r increases, the Ncut curve becomes smoother.



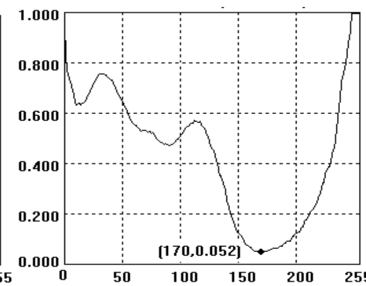
(a) $r=2$, $d_I=625$, $d_X=4$



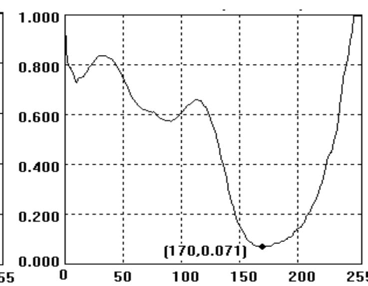
(b) $r=4$, $d_I=625$, $d_X=8$



(c) $r=8$, $d_I=625$, $d_X=16$

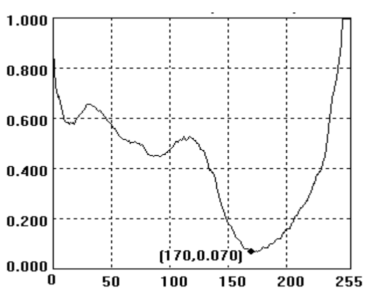


(d) $r=16$, $d_I=625$, $d_X=32$

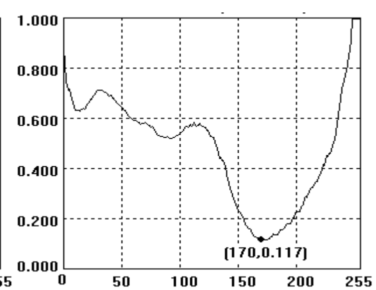


(e) $r=32$, $d_I=625$, $d_X=250$

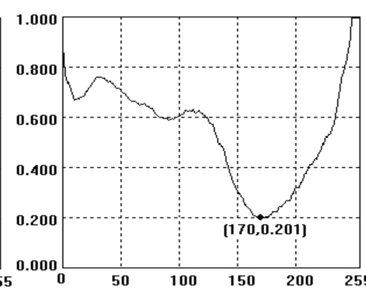
- As d_I increases, Ncut values increases.



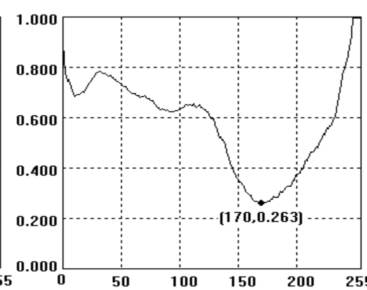
(f) $r=4$, $d_I=1000$, $d_X=8$



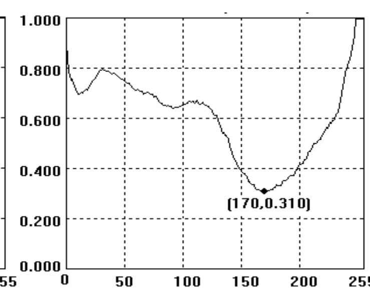
(g) $r=4$, $d_I=2000$, $d_X=8$



(h) $r=4$, $d_I=5000$, $d_X=8$



(i) $r=4$, $d_I=10000$, $d_X=8$

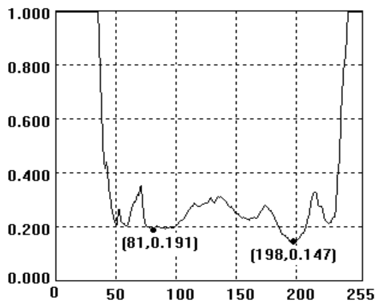


(j) $r=4$, $d_I=20000$, $d_X=8$

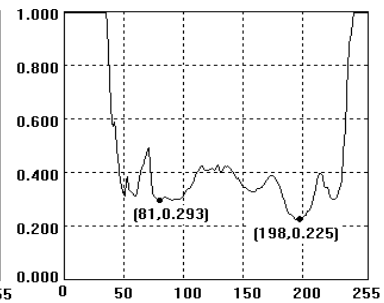
- In general, Ncut is insensitive to these parameters.

Experimental Results

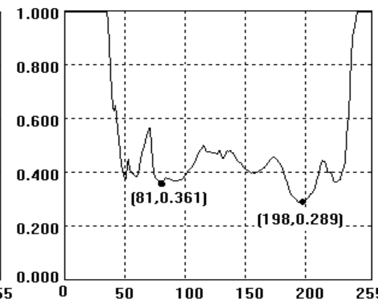
Ship image



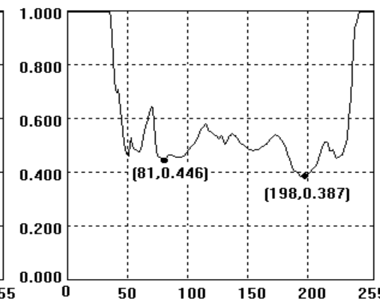
r=2, d_I=1000, d_X=4



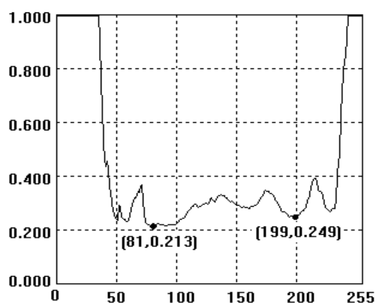
r=4, d_I=2000, d_X=8



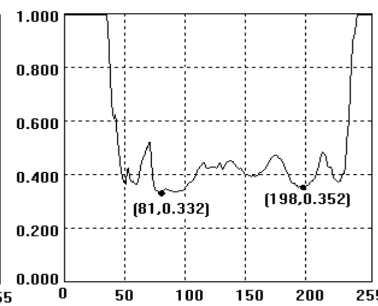
r=6, d_I=3000, d_X=10



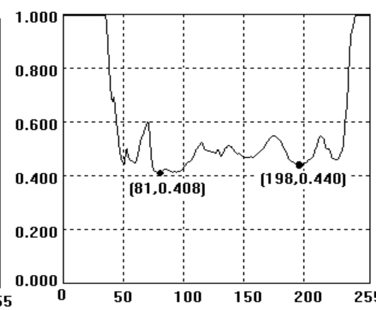
r=8, d_I=5000, d_X=16



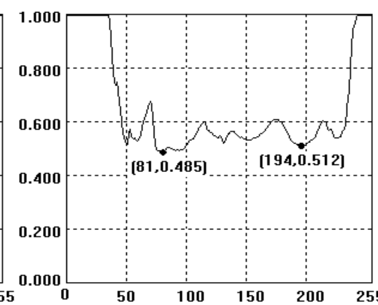
r=2, d_I=4000, d_X=4



r=4, d_I=7000, d_X=8



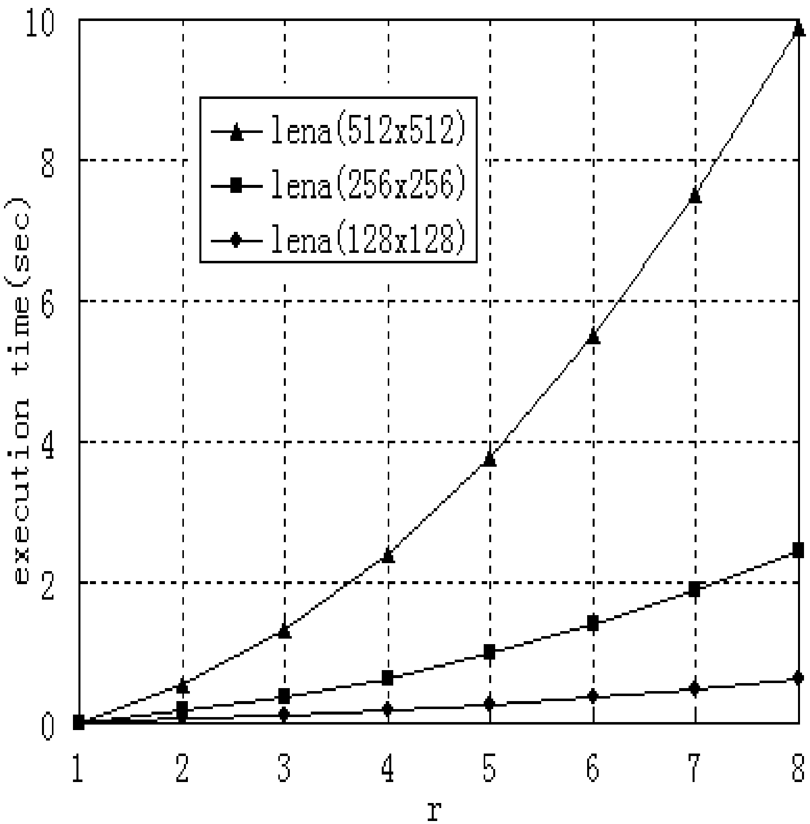
r=6, d_I=13000, d_X=10

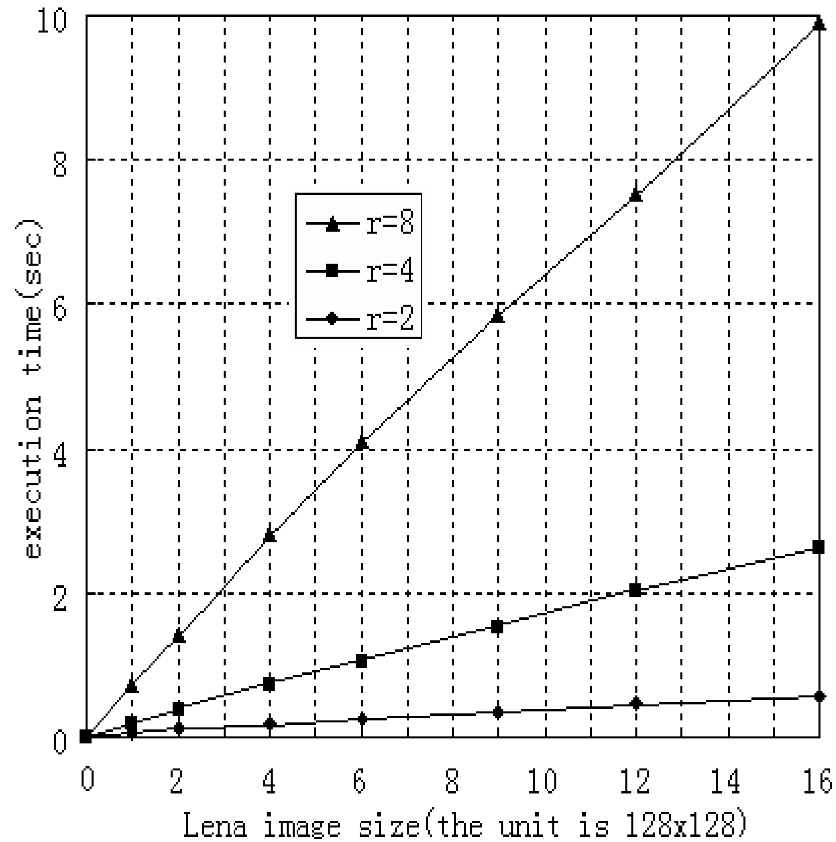


r=8, d_I=18000, d_X=16

- **As d_I increases to an extremely large value, the optimum value of T shifts from right to left.**
- **This does not usually happen for typical values of d_I (400 – 1000).**

Computational Complexity





Execution time vs. *r*.

Execution time vs. image size in the multiple of 128x128.

改进2——基于Mean Shift and Ncuts的彩色图像分割

Wenbing Tao, et.al, "Color Image Segmentation Based on Mean Shift and Normailized Cuts", *IEEE Transactions on Systems Man and Cybernetics Part B-Cybernetics*. 2007, Vol 37, No.5, October, pp 1382-1389.

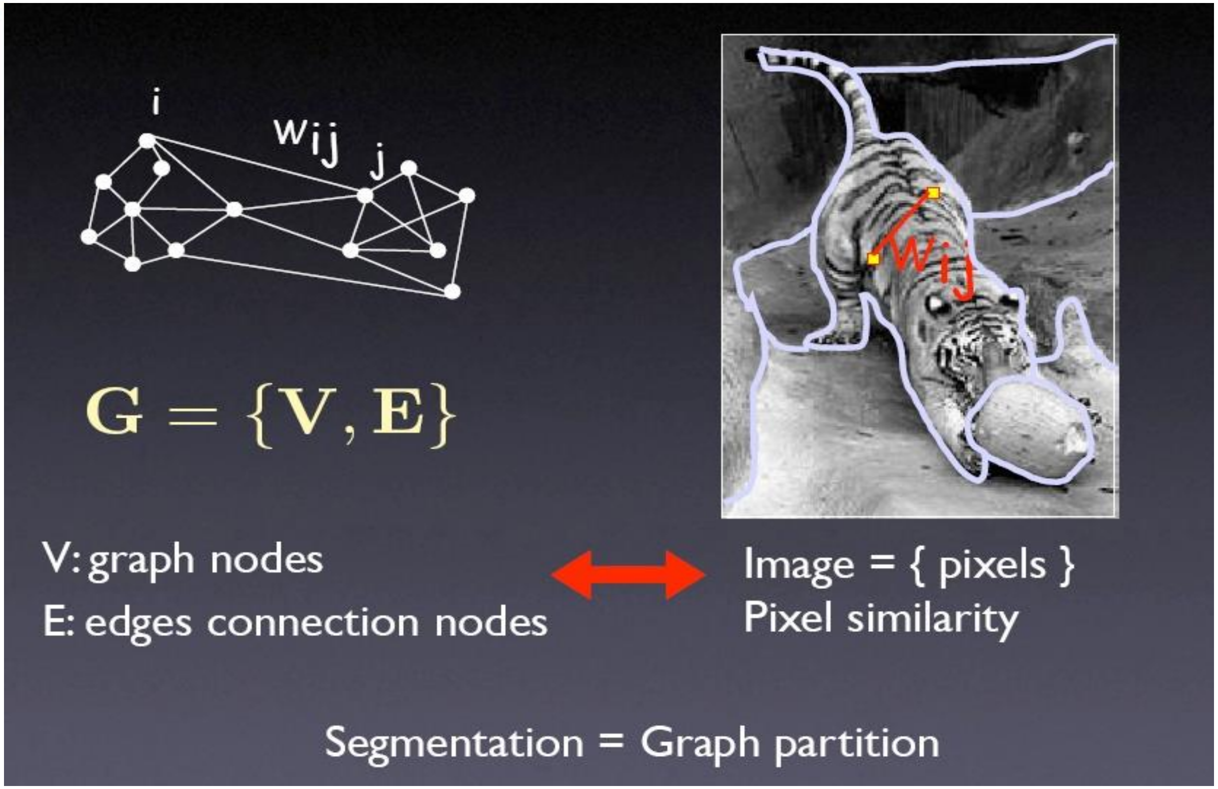
图像分割方法分类

- 基于特征的聚类方法：利用图像的颜色、纹理等视觉信息
- 基于空间的分割方法：在空间域内处理分割，基于区域
- 基于图论的分割方法：融合图像特征信息和空间信息的知觉分组

$$Ncut(A,B)=\frac{cut(A,B)}{asso(A,V)}+\frac{cut(A,B)}{asso(B,V)}$$

- ✓ Segmentation=Graph Partition
- ✓ 将图像中每个象素看作图的一个节点，构建节点关系权值矩阵
- ✓ 求解Laplacian Matrix的特征系统，矩阵维数为节点的个数，维数较高
- ✓ 有效，但是复杂性高,处理一幅160×160的灰度图像需30秒到2分钟

如何克服其存储及计算复杂性问题？



Problem?

- **Image size**
- **Storage space**
- **Computational cost**

$$\boxed{Ncut(A,B)} = \frac{cut(A,B)}{asso(A,V)} + \frac{cut(A,B)}{asso(B,V)}$$

minimized

NP hard

$$\mathbf{D}^{-\frac{1}{2}}(\mathbf{D}-\mathbf{W})\mathbf{D}^{-\frac{1}{2}}z = \lambda z$$

Compute the eigen vector of matrix D-W

Average cut
Average associate

Be superior to

Robustly generate balanced clusters

video summarization

scene detection

cluster-based image retrieval

Image segmentation

Be used to

Normalized Cut (Ncut)

Problems

High dimension

Low computation efficiency

Not suitable for real-time

Down-sample to 160×160 gray image
The clustering number: 5
Computation time: about 30s

Image size is 240×160

Normalized Cuts(Ncut)算法

1. 能够有效集成图像象素颜色、纹理等特征信息及空间位置信息
2. 具有良好的平衡划分性能，分割性能较好
3. 分割类别个数事先指定，利于场景划分
4. 权值矩阵维数为图像中象素的个数，特征系统维数太高，无法处理大尺度的图像
5. 计算复杂性高，不利于实时应用

Discontinuity preserving smoothing
Image segmentation performanc

Excellent

Mean Shift Algorithm (MS)

advantages

- Reduce the image basic entities
- The salient features are retained
- Regions can represent image

Problem

Difficult to partition image scenes, depending only on the MS

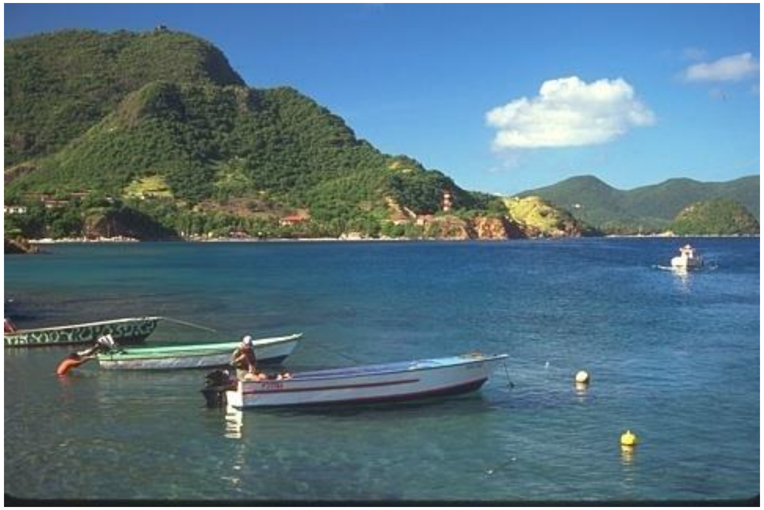
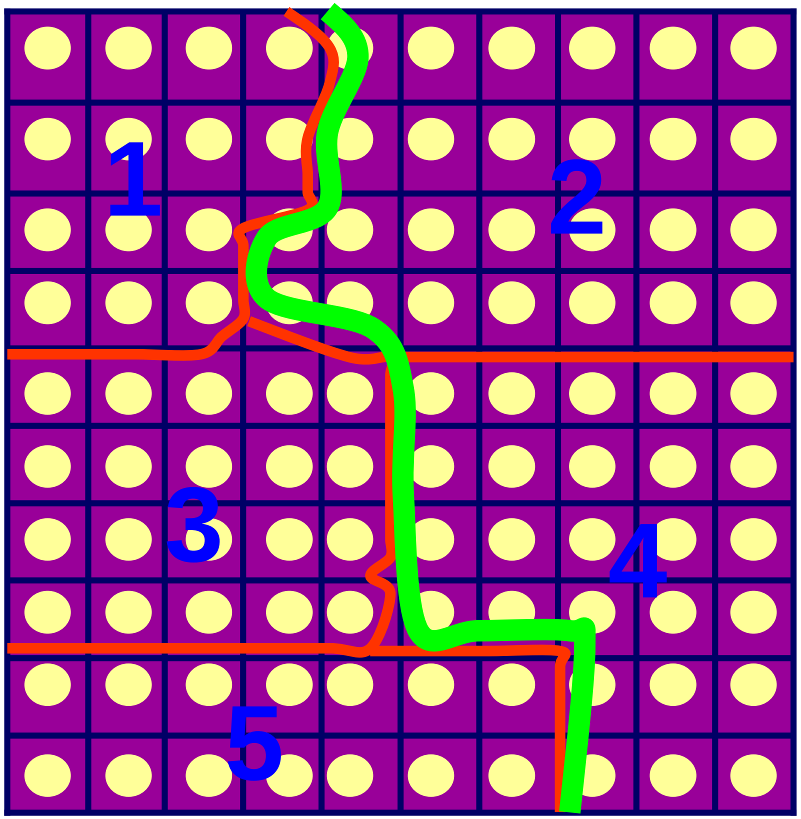


Image sized is 240×160

MS result: 128 regions
Computation time: about 2s

Graph partition based on regions



Computational complexity significantly decreases

100 pixel nodes
Weighted matrix:100 × 100
Solve 100 × 100 eigen system

Pre-segment into 5 regions
Weighted matrix: 5 × 5
Sove 5 × 5 eigen system

Apply the graph representation strategy on the regions that are derived from the original image by **MS** method. Then, the **Ncut** method can be applied to form the final segmentation results.

Principle

Integrate MS and Ncut

Implementaion procedure

Advantages

- The dimension largely decreases
- Low computational complexity
- Noise is reduced
- The salient features is retained
- Improved segmentation performance

MS segmentation

Graph strategy represents regions → Construct weighted matrix → Solve eigen system → Partition image

MS和Ncut结合

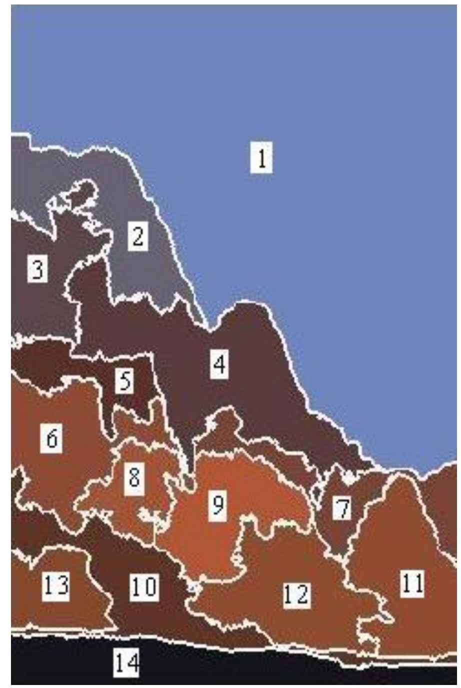
集成Mean Shift算法和Normalized Cuts算法的优点，弥补各自的不足

先采用Mean Shift进行初始区域分割，再采用Ncut对区域节点进行分组，完成场景的划分，提高算法的实时性能。

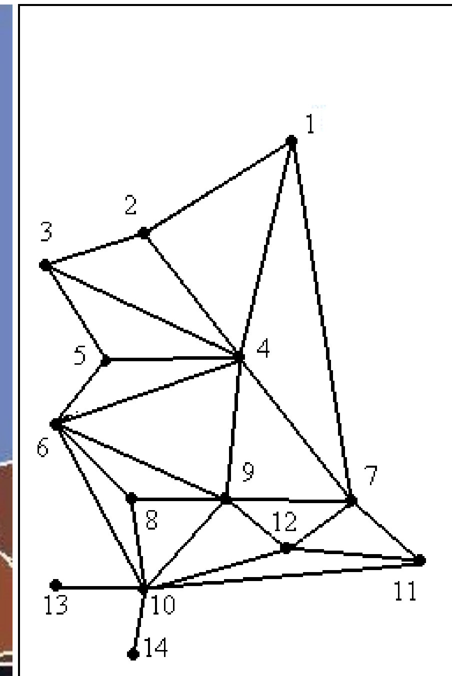
Illustration of the Implementation procedure

Original image

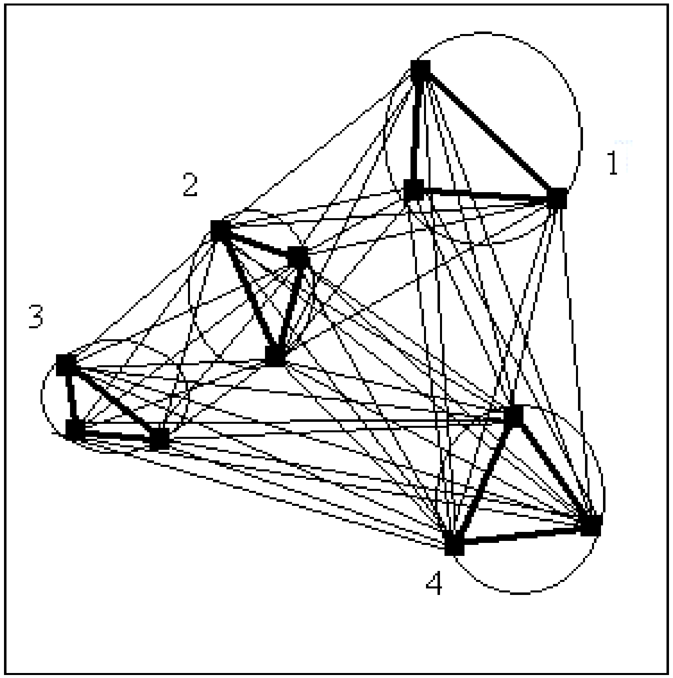
Mean shift: 14 regions



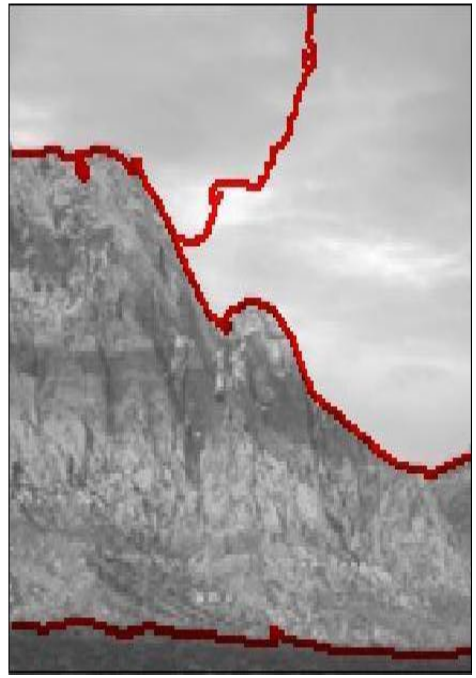
Region label



Region adjacency graphs (RAG)



RAG with a region corresponding to three nodes

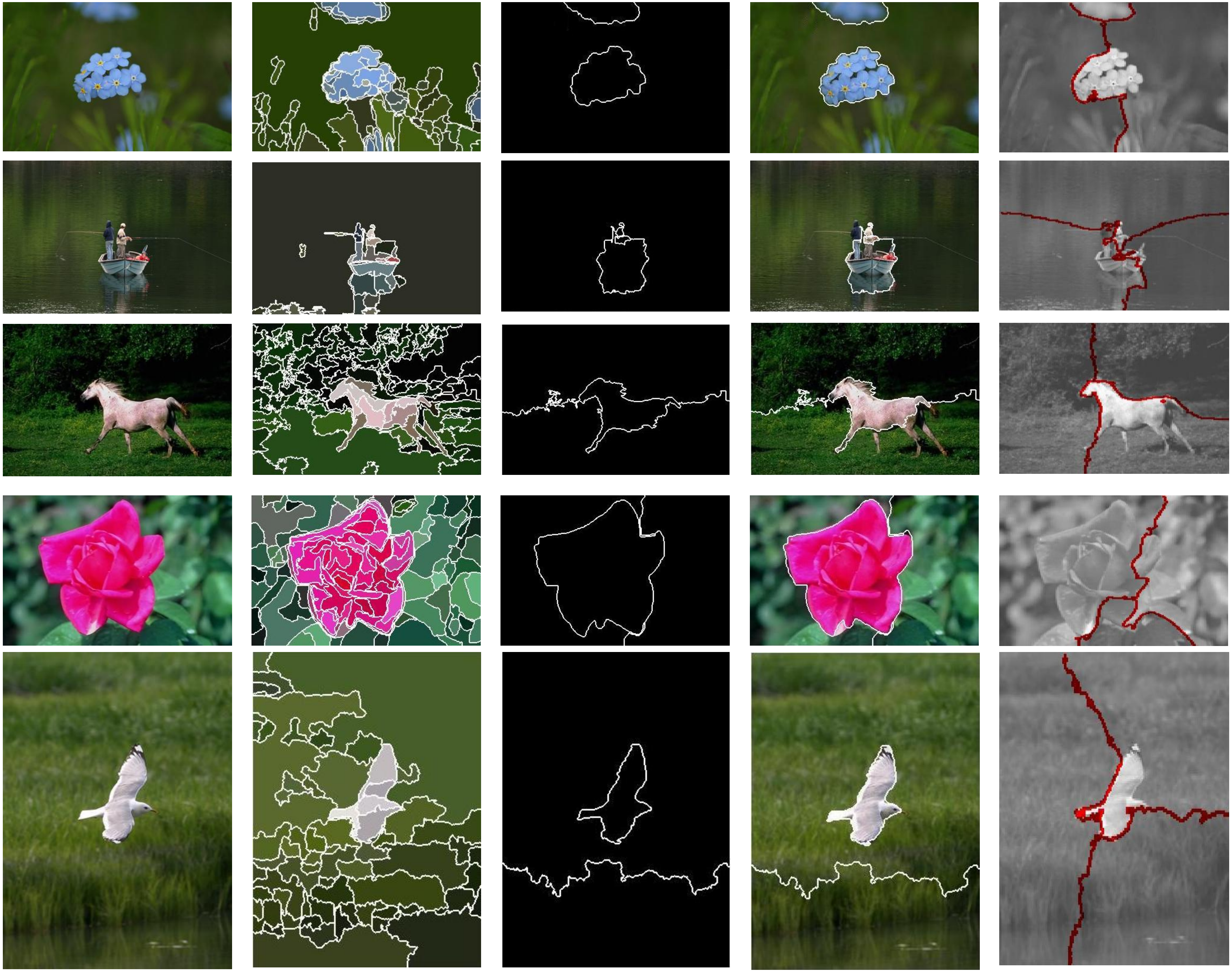


The result by Ncut

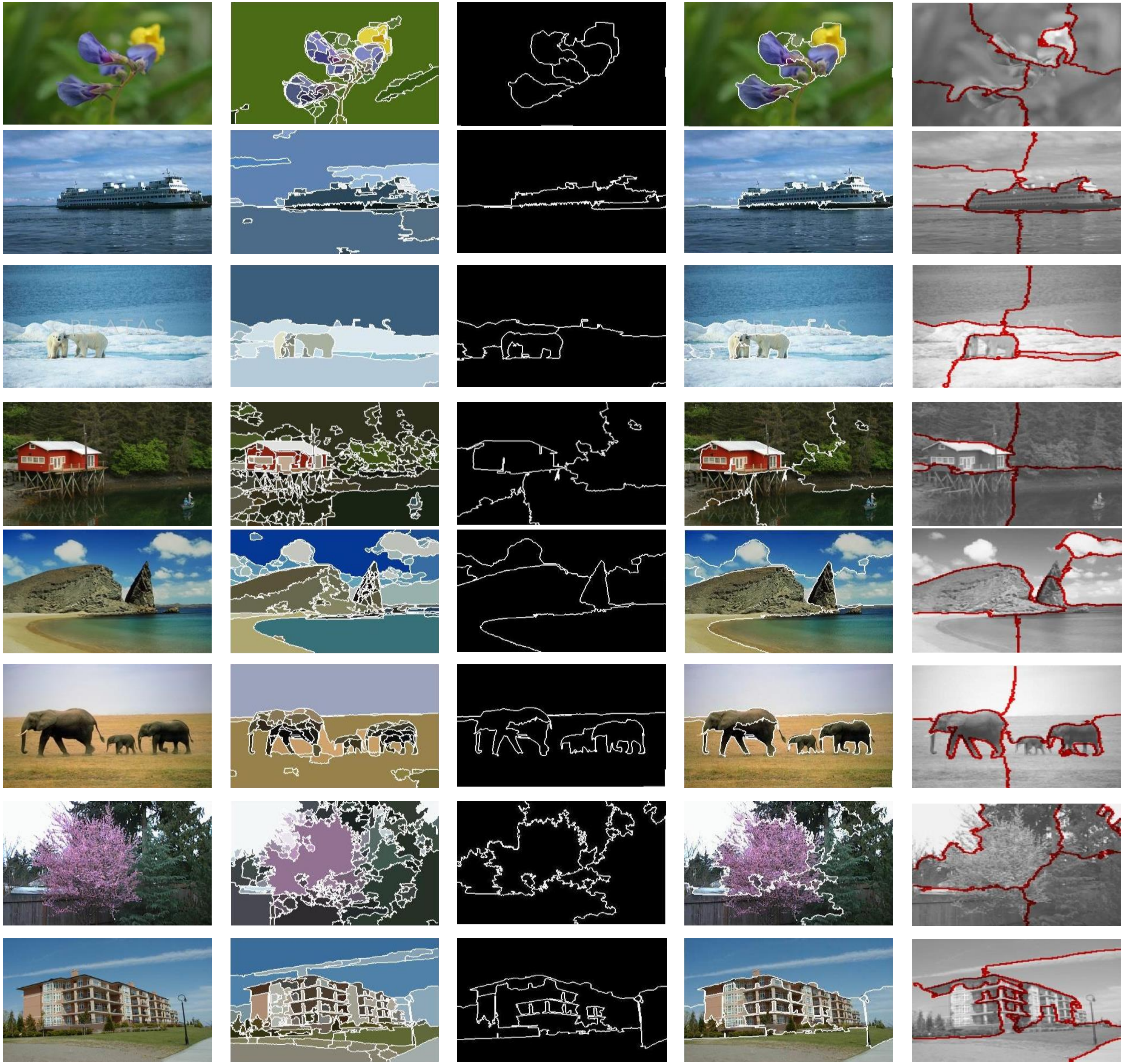
The result by proposed

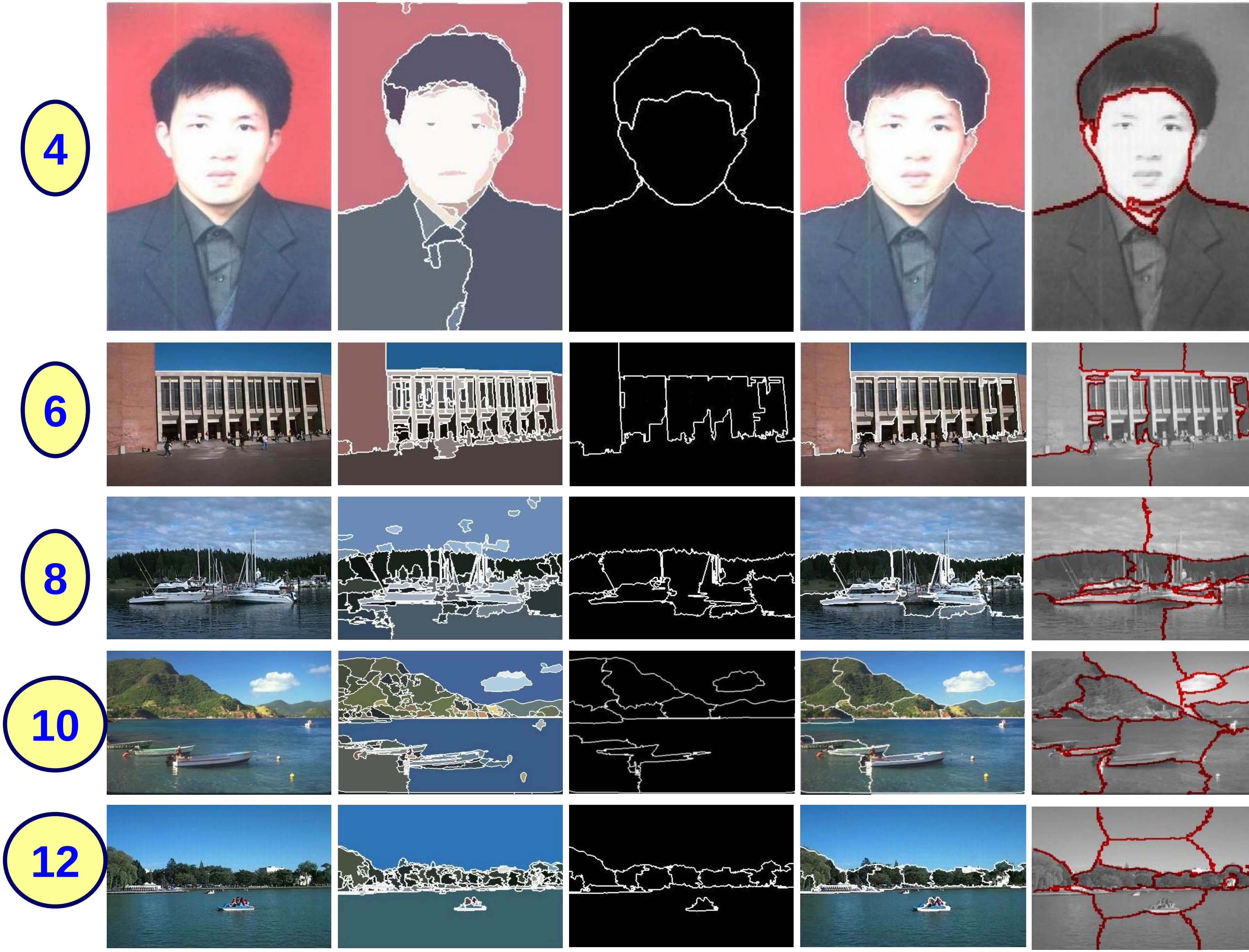
3

3



6




4
6
8
10
12