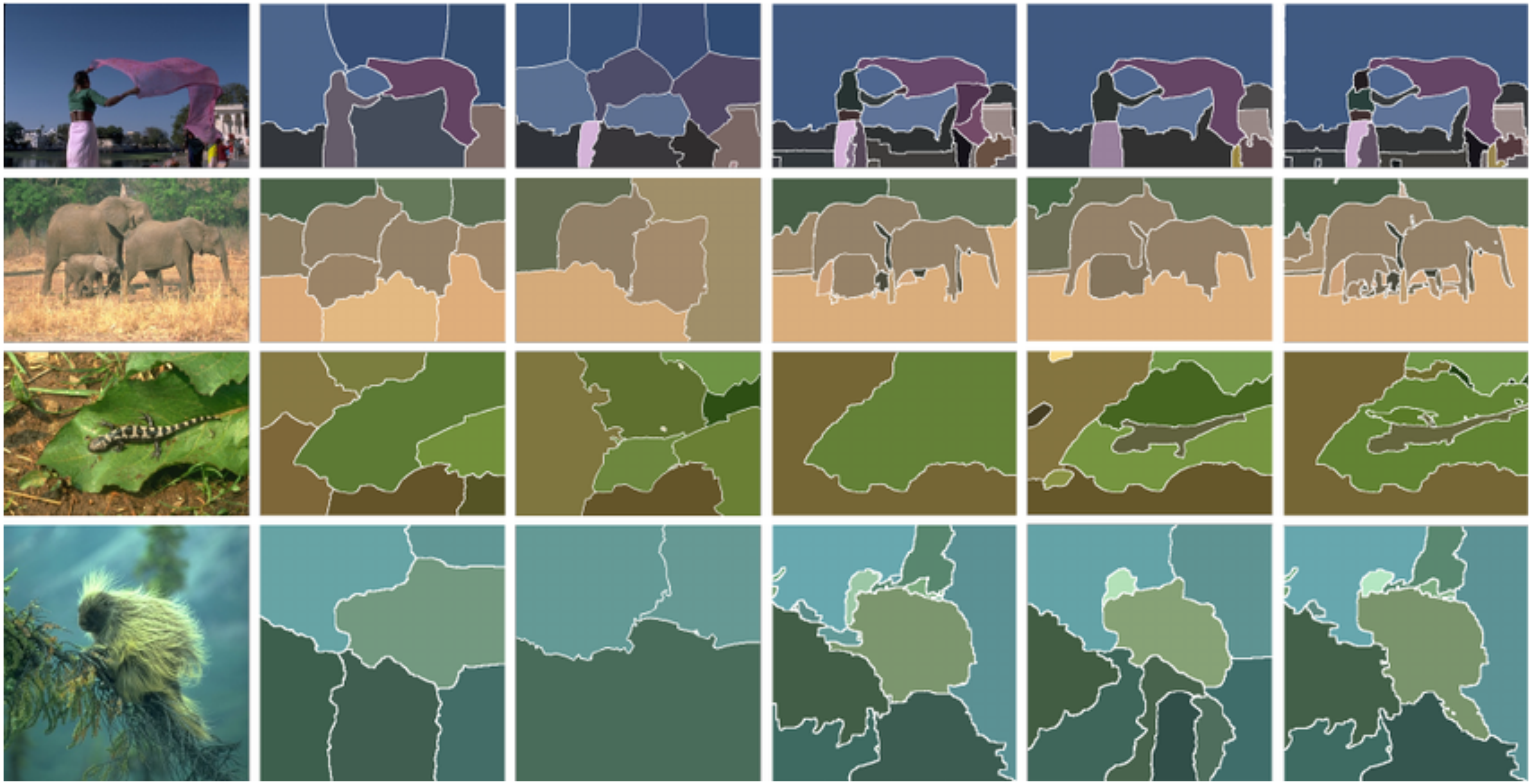
Normalized Cuts and Image Segmentation

Presented by
Yue Bi

- What is image segmentation?

- *Normalized Cuts segmentation results of Berkeley Segmentation Dataset

- *Human Segmentation, Graph Based Salient Object Detection



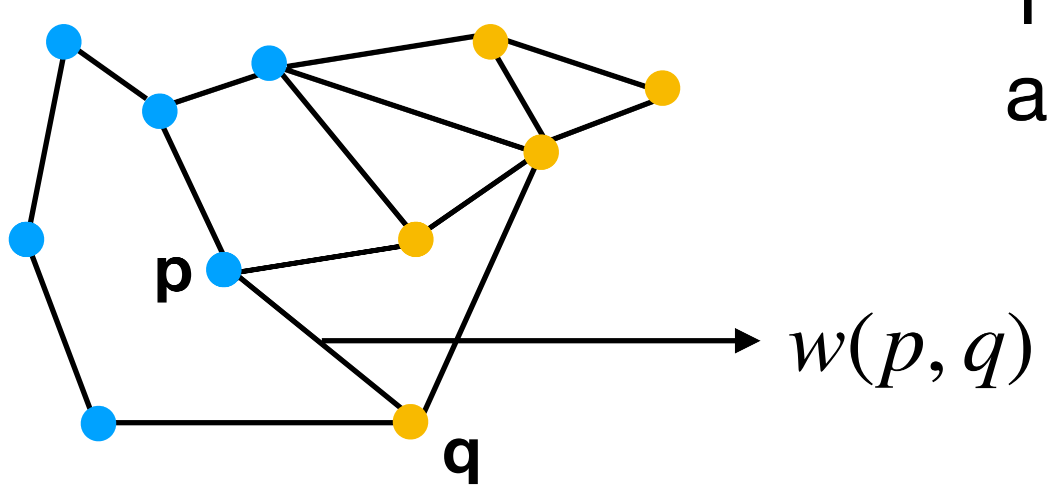
- *K-means clustering uses the Normalized Cut affinity

Outline

- Cut and Normalized Cut
- Grouping Algorithm
- Experiment Results
- Conclusion

Cut

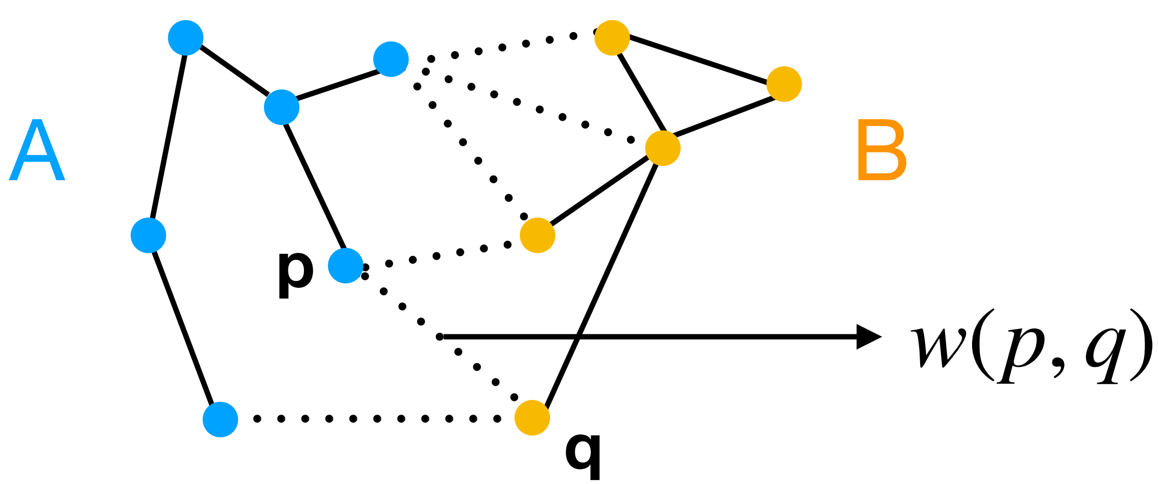
- A weighted graph $G = (V, E)$



The Edge between Node p
and Node q has weight $w(p, q)$

Cut

- Partition V into two disjoint sets A and B by removing all edges connecting two parts.



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

$$A \cap B = \varnothing \quad A \cup B = V$$

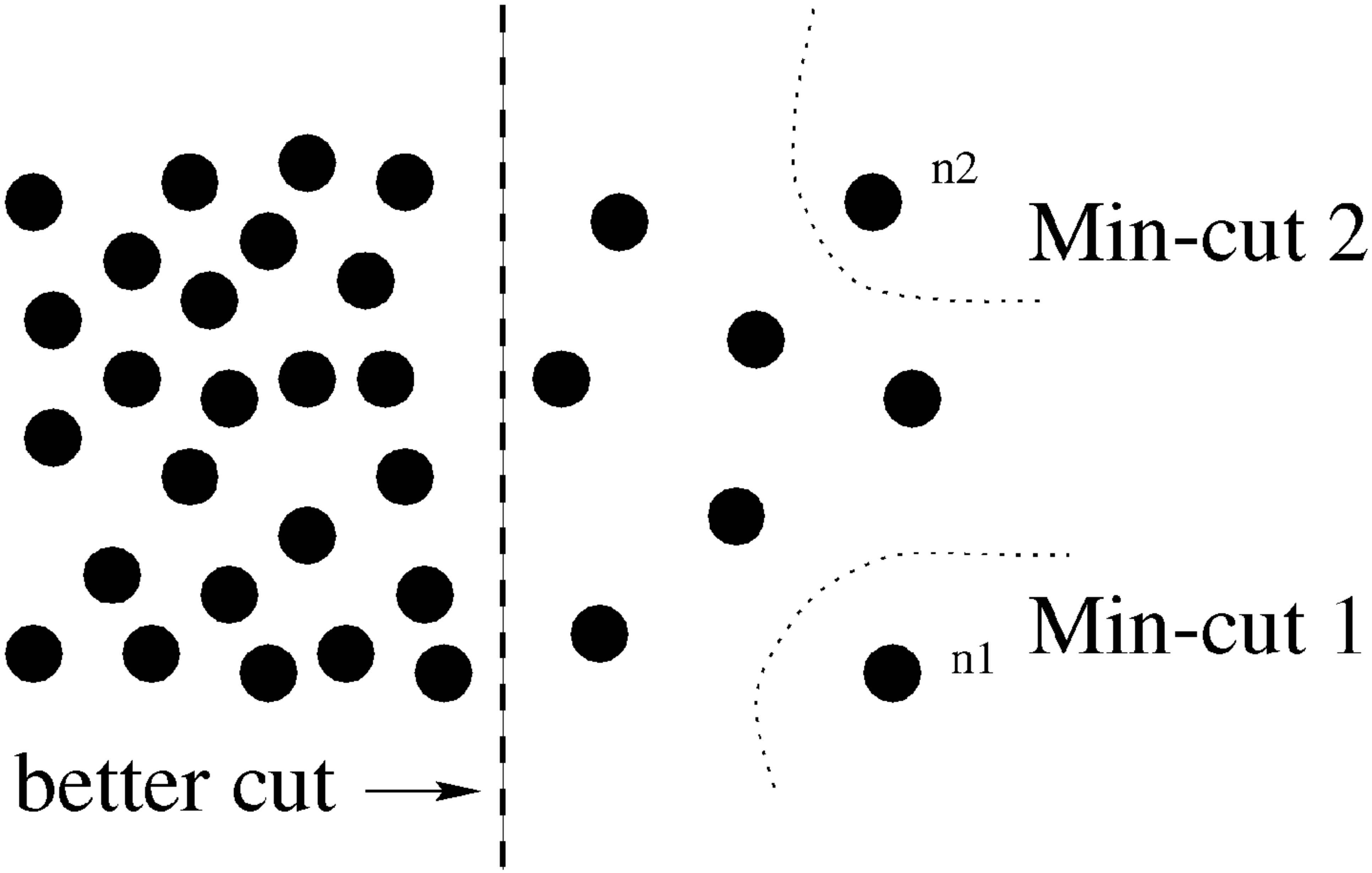
- The optimal bipartitioning of a graph is the one that minimizes the cut.

Drawback of Using Cut

- Cut increases as the number of edges going across the two partitioned parts increases.
- So the minimum cut criterion favors cutting small sets of isolated nodes in the graph.

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

Example of Bad Partition



New Criterion 1: Normalized Cut

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

- $assoc(A, V) = \sum_{u \in A, t \in V} w(u, t)$
- $assoc(A, V) = assos(A, A) + cut(A, B)$

- So the cut that partitions out small isolated points will no longer have small Ncut value.

New Criterion 2: Normalized Association

$$Nassoc(A,B) = \frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)}$$

- So we know:

$$Ncut(A,B) = \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(B,A)}{assoc(B,V)} = 2 - Nassoc(A,B)$$

- Hence, the two partition criteria that we seek, minimizing Ncut and maximizing Nassoc, are in fact identical.

Computing the Optimal Partition

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

- This is a NP-complete problem
- Can we simplify it?

Computing the Optimal Partition

$$Ncut(A,B) = \frac{\sum_{(x_i>0,x_j<0)} - w_{ij}x_ix_j}{\sum_{x_i>0} d_i} + \frac{\sum_{(x_i<0,x_j>0)} - w_{ij}x_ix_j}{\sum_{x_i<0} d_i}$$

- x is an $N = |V|$ dimensional vector, $x_i = 1$ if node $i \in A$ and $x_i = -1$ if $i \in B$
- $d_i = \sum_j w(i,j)$

Computing the Optimal Partition

- D: $N \times N$ diagonal matrix with d on its diagonal
- W: $N \times N$ symmetric weight matrix with $W(i,j) = w_{ij}$
- 1 : $N \times 1$ vector with all ones
- $\frac{1+x}{2}$: vector for A $\qquad$ $\frac{1-x}{2}$: vector for B
- $k = \frac{\sum_{x_i>0} d_i}{\sum_i d_i}$.

So we get:

$$4 \times Ncut(A,B) = \frac{(1+x)^T(D-W)(1+x)}{k1^TD1} + \frac{(1-x)^T(D-W)(1-x)}{(1-k)1^TD1}$$

$$4 \times Ncut(A,B) = \frac{(1+x)^T(D-W)(1+x)}{k1^TD1} + \frac{(1-x)^T(D-W)(1-x)}{(1-k)1^TD1}$$

- Let:

$$b = \frac{k}{1-k} = \frac{\sum_{x_i>0} d_i}{\sum_{x_i<0} d_i}$$

$$y = \frac{(1+x) - b(1-x)}{2}$$

- Since:

$$y^TD1 = \sum_{i=1}^{n} y_i d_i = \sum_{x_i>0} d_i - b \sum_{x_i<0} d_i = 0$$

$$y^TDy = \sum_{i=1}^{n} y_i^2 d_i = \sum_{x_i>0} d_i + b^2 \sum_{x_i<0} d_i = b1^TD1$$

- Then:

$$Ncut(A,B) = \frac{y^T(D-W)y}{b1^TD1} = \frac{y^T(D-W)y}{y^TDy}$$

Constrained Optimization Problem

$$min_x Ncut(x) = min_y \frac{y^T(D-W)y}{y^T D y}$$

- Subject to: $y(i) \in \{1, -b\} \quad y^T D 1 = 0$
- The above expression is Rayleigh quotient
- Minimize Ncut by solving eigenvalue system of

$$(D-W)y = \lambda D y \qquad (y^T D 1 = 0)$$

Normalized Laplacians

- $D^{-\frac{1}{2}}(D-W)D^{-\frac{1}{2}}z = \lambda z \quad (z = D^{\frac{1}{2}}y)$
- Smallest eigenvalue is $\lambda_0 = 0$ with eigenvector $z_0 = D^{\frac{1}{2}}1$
- Eigenvector with second smallest eigenvalue is the real solution to the normalized cut problem since it is perpendicular to z_0
- Eigenvector with third smallest eigenvalue is that optimally sub partitions the first two parts
- …

Drawback of using higher eigenvectors

- In practice, solutions based on higher eigenvectors become unreliable.

- It is best to restart the algorithm and use only second smallest eigenvector.

Outline

- Cut and Normalized Cut
- Grouping Algorithm
- Experiment Results
- Conclusion

Recursive 2-way Grouping Algorithm

- Set the weight on the edge in graph $G = (V, E)$ as:

$$w_{ij} = e^{\frac{-\|F_{(i)} - F_{(j)}\|_2^2}{\sigma_I^2} + \frac{-\|X_{(i)} - X_{(j)}\|_2^2}{\sigma_X^2}} \quad if \parallel X_{(i)} - X_{(j)} \parallel_2 < r$$

- where $X_{(i)}$ is the spatial location of node i.

- $F_{(i)}$ is feature vector based on intensity, color, or texture information.

Recursive 2-way Grouping Algorithm

- Set the weight on the edge in graph $G = (V, E)$ as:

$$w_{ij} = e^{\frac{-\|F_{(i)} - F_{(j)}\|_2^2}{\sigma_I^2} + \frac{-\|X_{(i)} - X_{(j)}\|_2^2}{\sigma_X^2}} \quad if \ \| X_{(i)} - X_{(j)} \|_2 < r$$

- Solve $(D - W)y = \lambda Dy$

- Use the eigenvector with the second smallest eigenvalue to bipartition the graph

- Recursion if needed.

Complexity

- $O(n^3)$ in general.
- $O(n^{\frac{3}{2}})$ in actual experiment observation since:
  - Sparse resulting eigensystems
  - Few eigenvectors needed
  - Low precision requirement

Simultaneous K-way Cut with Multiple Eigenvectors

- Step 1: Use clustering algorithm to partition image into k* (k*>k) groups.

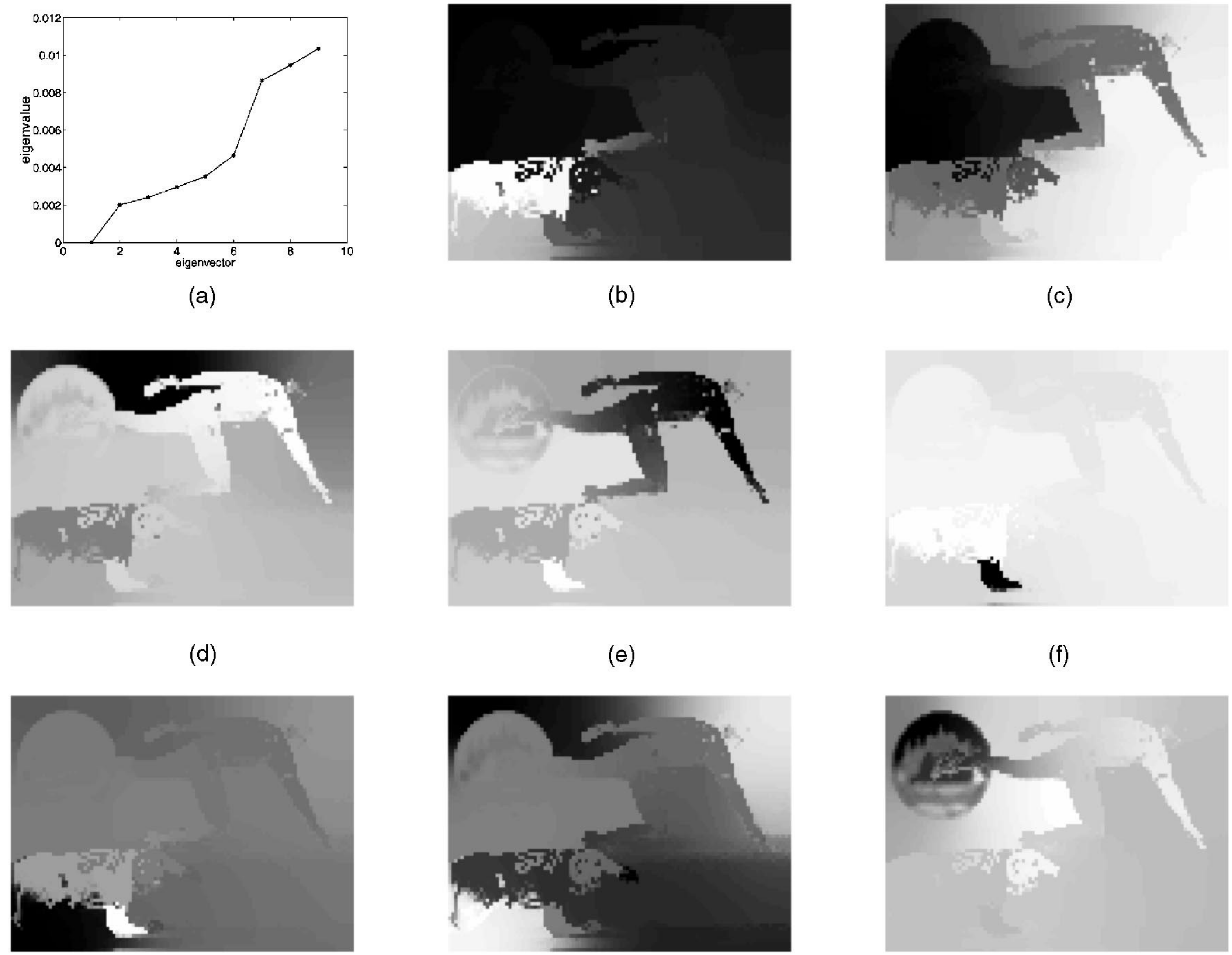
- Step 2: Iteratively merge two segments if that minimizes the k-way Ncut criterion:

$$Ncut_k = \frac{cut(A_1, V - A_1)}{assoc(A_1, V)} + \frac{cut(A_2, V - A_2)}{assoc(A_2, V)} + \ldots + \frac{cut(A_k, V - A_k)}{assoc(A_k, V)}$$

Outline

- Cut and Normalized Cut
- Grouping Algorithm
- Experiment Results
- Conclusion

Eigenvectors

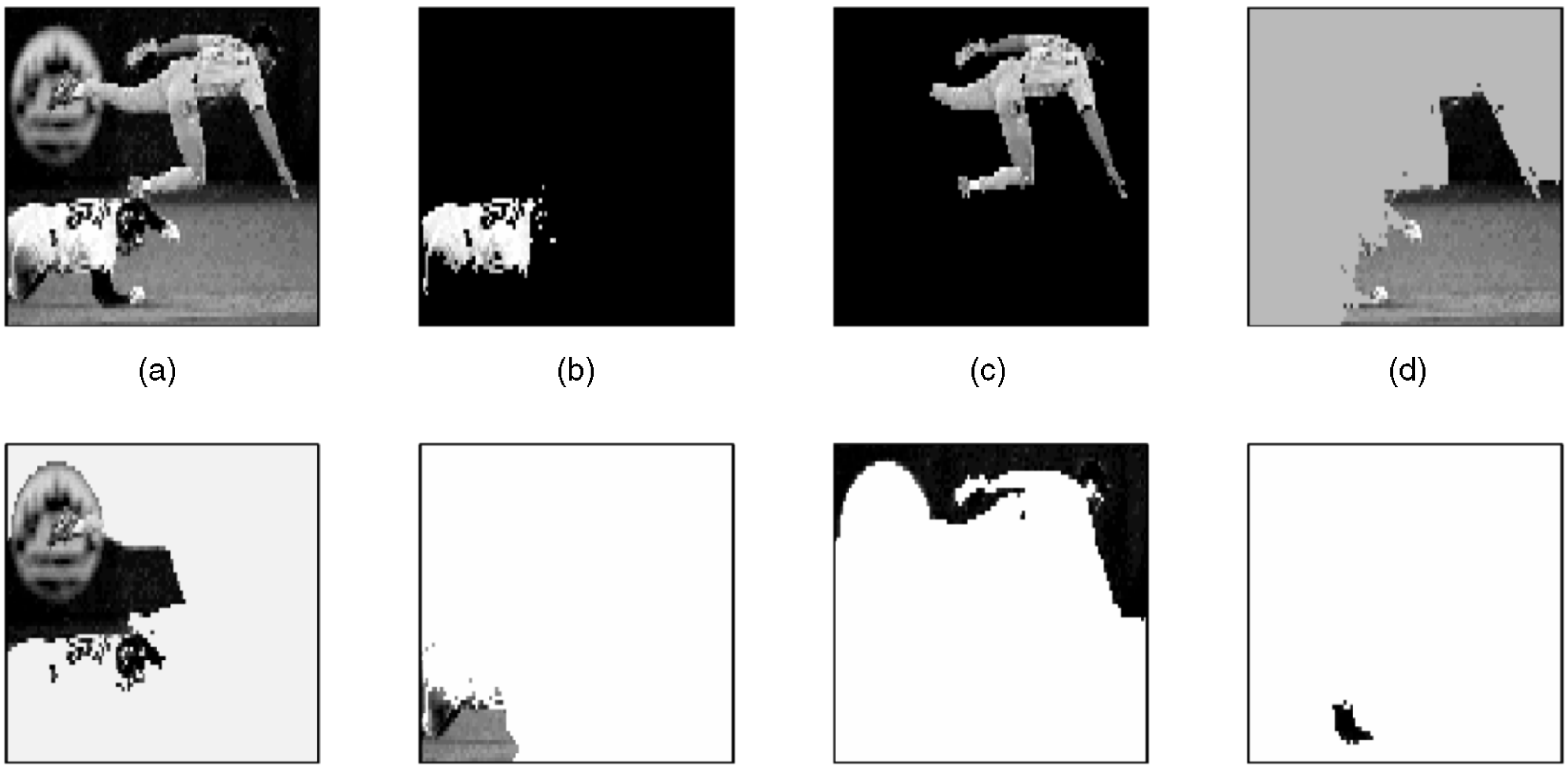


(a) (b) (c)

(d) (e) (f)

(g) (h) (i)

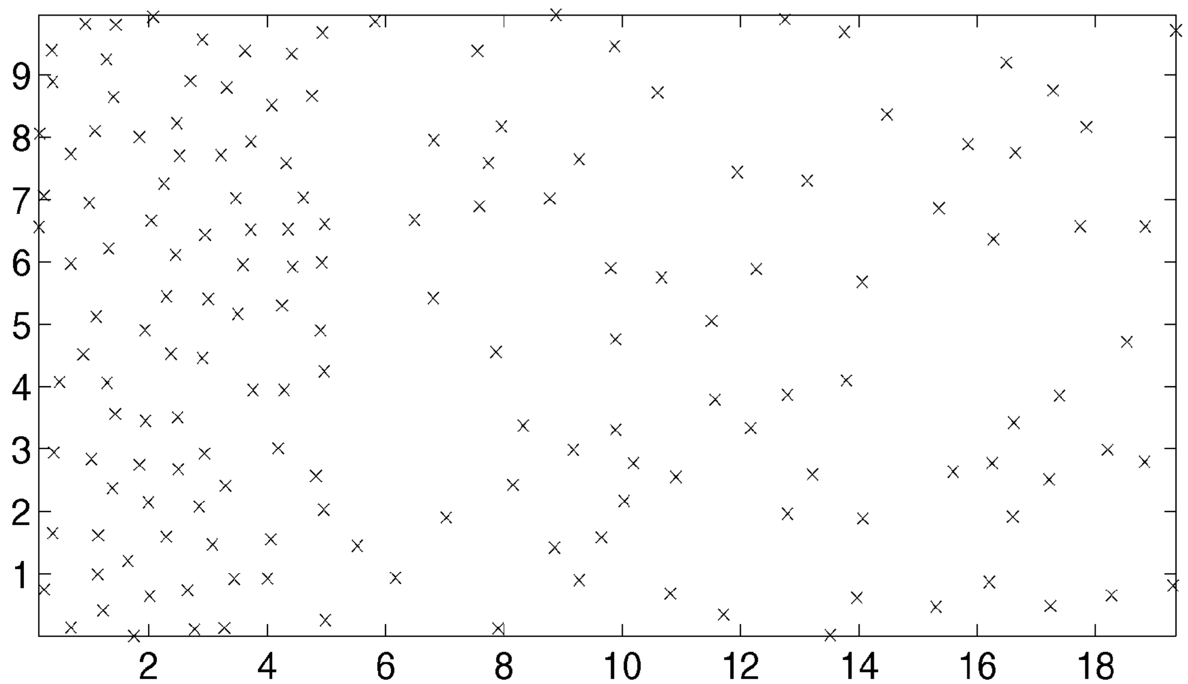
Partitions



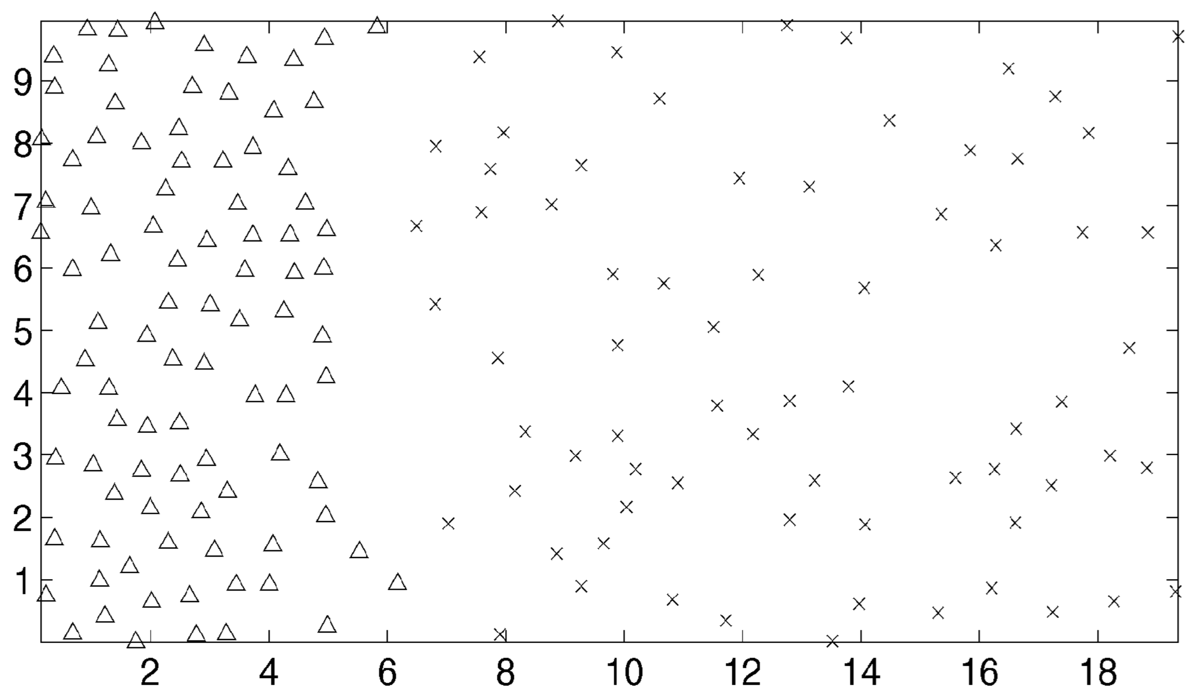
(a) (b) (c) (d)

(e) (f) (g) (h)

Partition of Generated Points

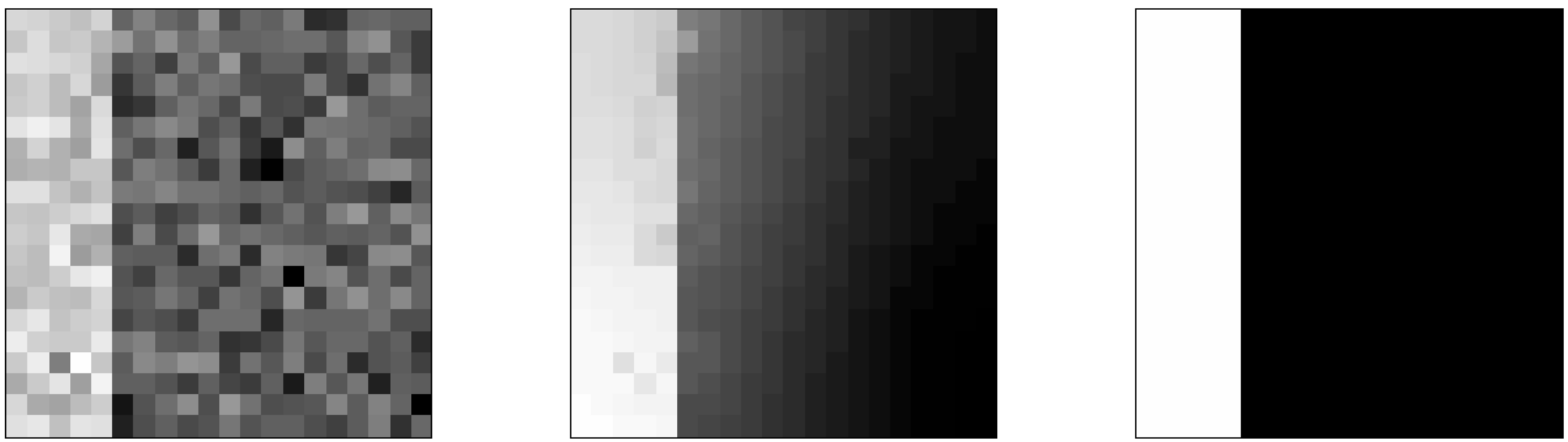


(a)



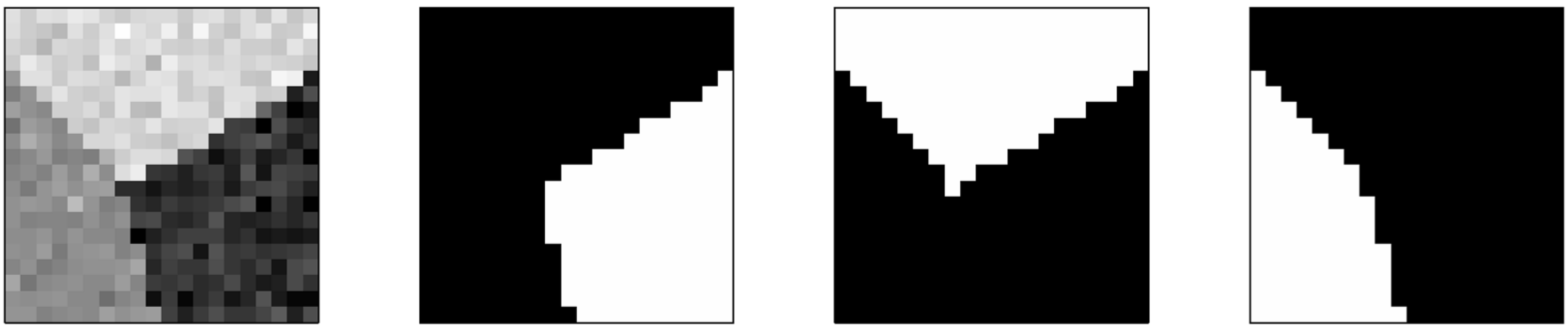
(b)

Partition of Synthetic Noisy Image



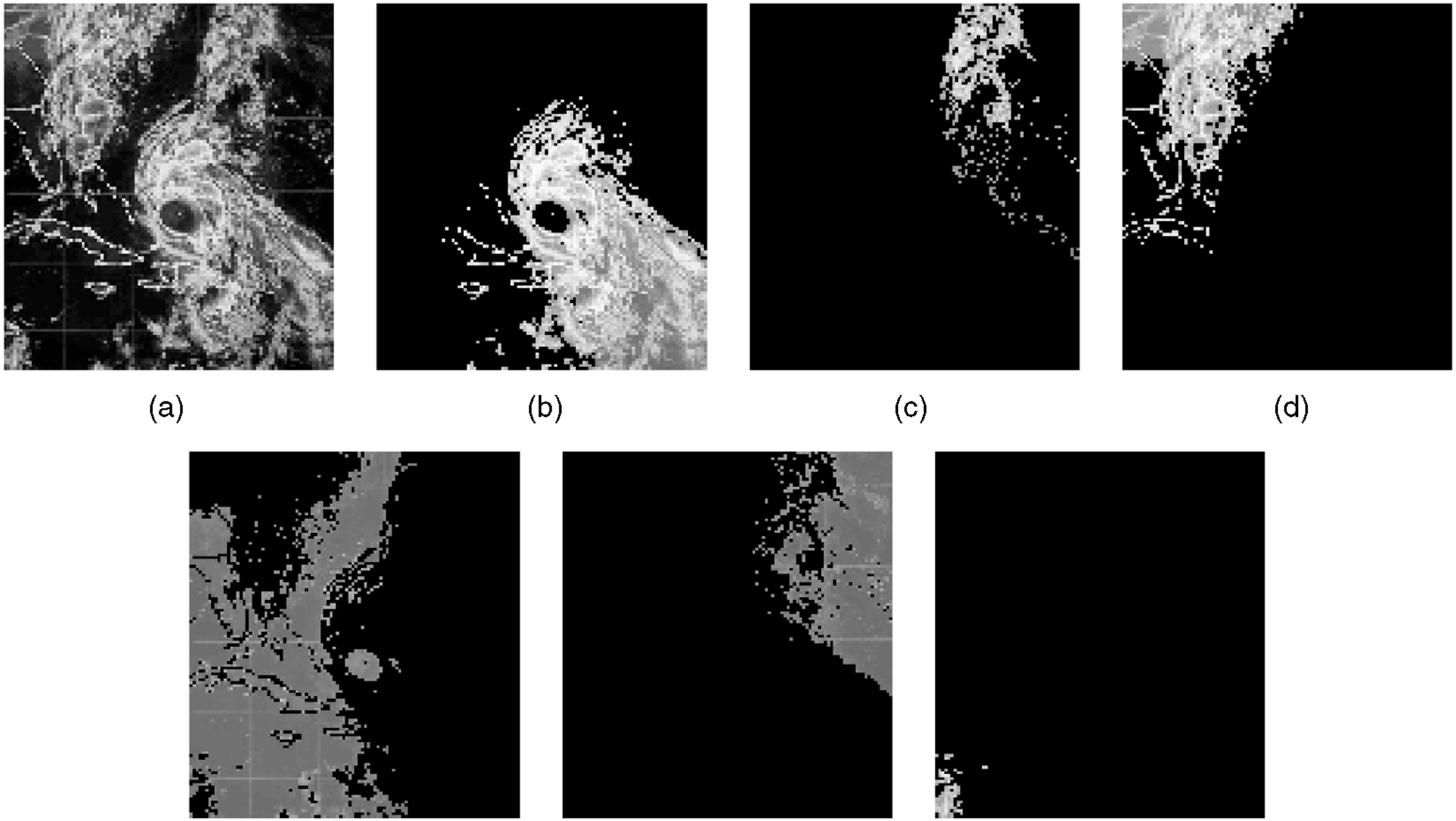
(a) (b) (c)

Partition of Synthetic Joint 3-patches Image



(a) (b) (c) (d)

Partition of Weather Radar Image



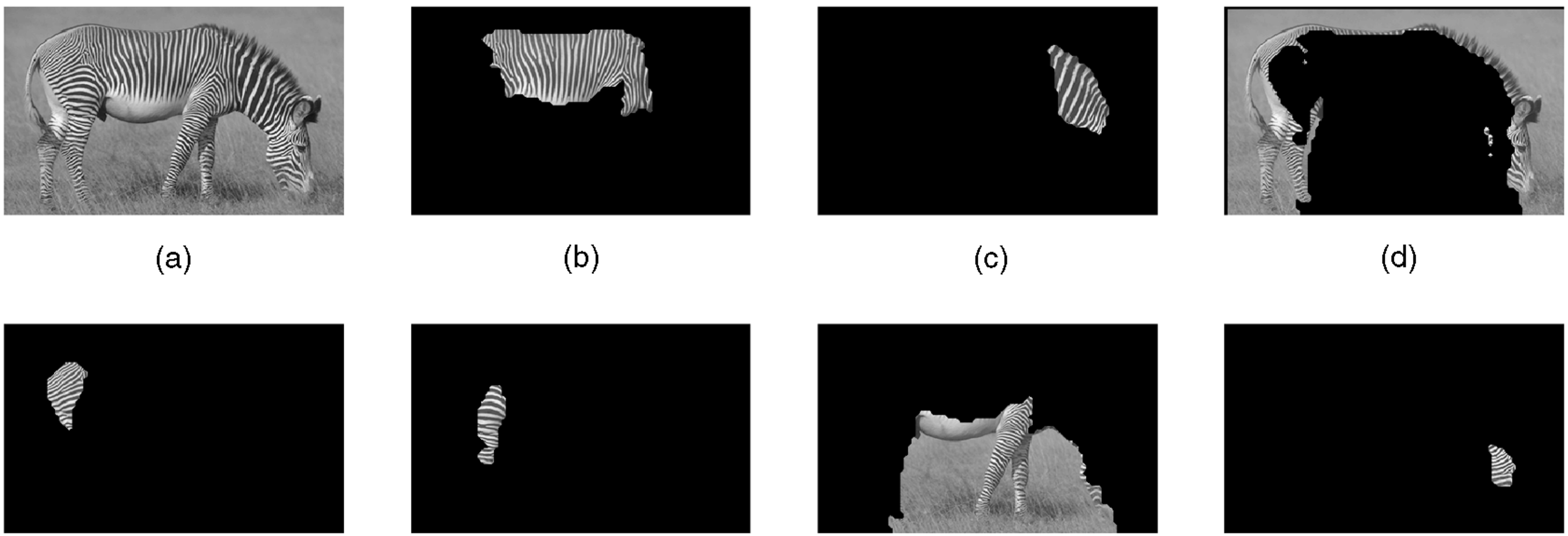
(a) (b) (c) (d)

(e) (f) (g)

Partition of Color Image (Reproduced)

(a) (b) (c) (d) (e)

Texture Segmentation



(a) (b) (c) (d)

(e) (f) (g) (h)

Partition of Image Sequence

(a) (b)

(c) (d) (e) (f) (g)

Outline

- Cut and Normalized Cut
- Grouping Algorithm
- Experiment Results
- Conclusion

Comparison with Other Eigenvector-Based Methods

Finding clumps ⟷ Finding splits

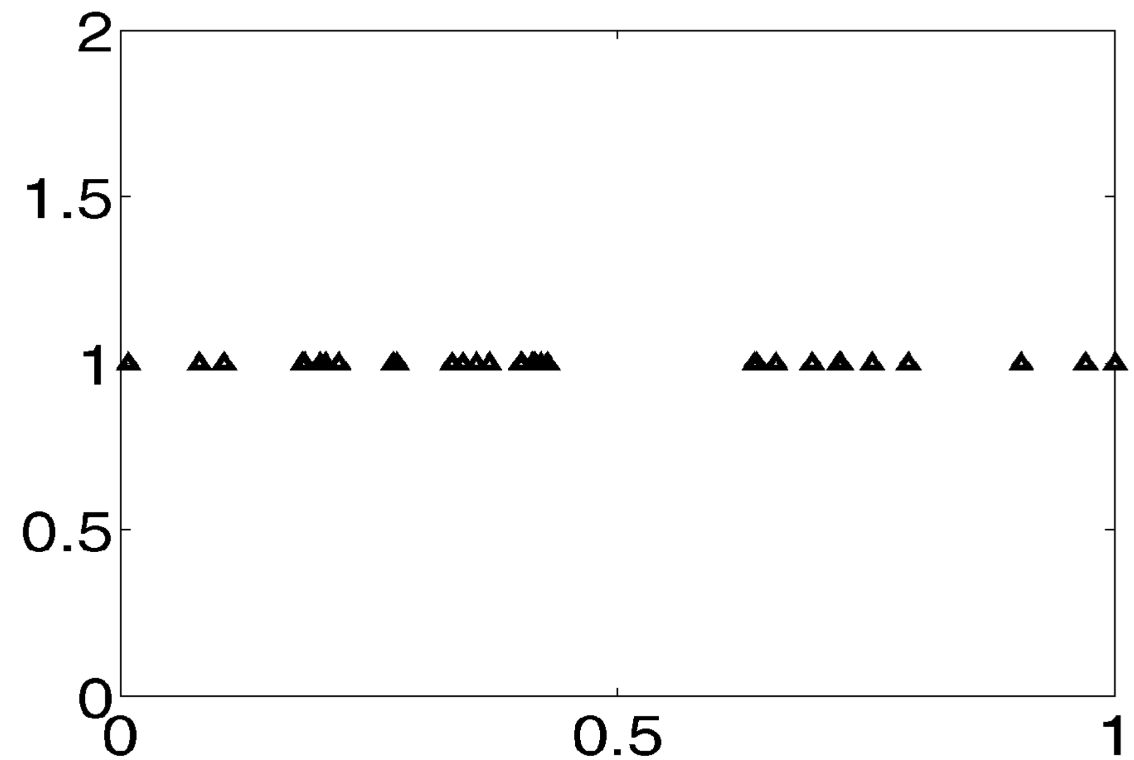
	Average association	Normalized Cut	Average cut
Discrete formulation	$\frac{asso(A,A)}{\lvert A\rvert} + \frac{asso(B,B)}{\lvert B\rvert}$	$\frac{cut(A,B)}{asso(A,V)} + \frac{cut(A,B)}{asso(B,V)}$ or $2 - \left(\frac{asso(A,A)}{asso(A,V)} + \frac{asso(B,B)}{asso(B,V)}\right)$	$\frac{cut(A,B)}{\lvert A\rvert} + \frac{cut(A,B)}{\lvert B\rvert}$
Continuous solution	$Wx = \bar{\lambda} x$	$(D-W)x = \bar{\lambda} D x$ or $W x = (1-\bar{\lambda}) D x$	$(D-W)x = \bar{\lambda} x$

Comparison Example

- A set of randomly distributed points in one dimension.
- The weight is defined to be inversely proportional to the distance as:

$$w(x) = f(d(x))$$

- The paper shows results of segmentation using three different monotonically decreasing function



$$w(x) = e^{-\left(\frac{d(x)}{0.1}\right)^2}$$

Normalized Cut:

$$(D - W)x = \lambda D x$$

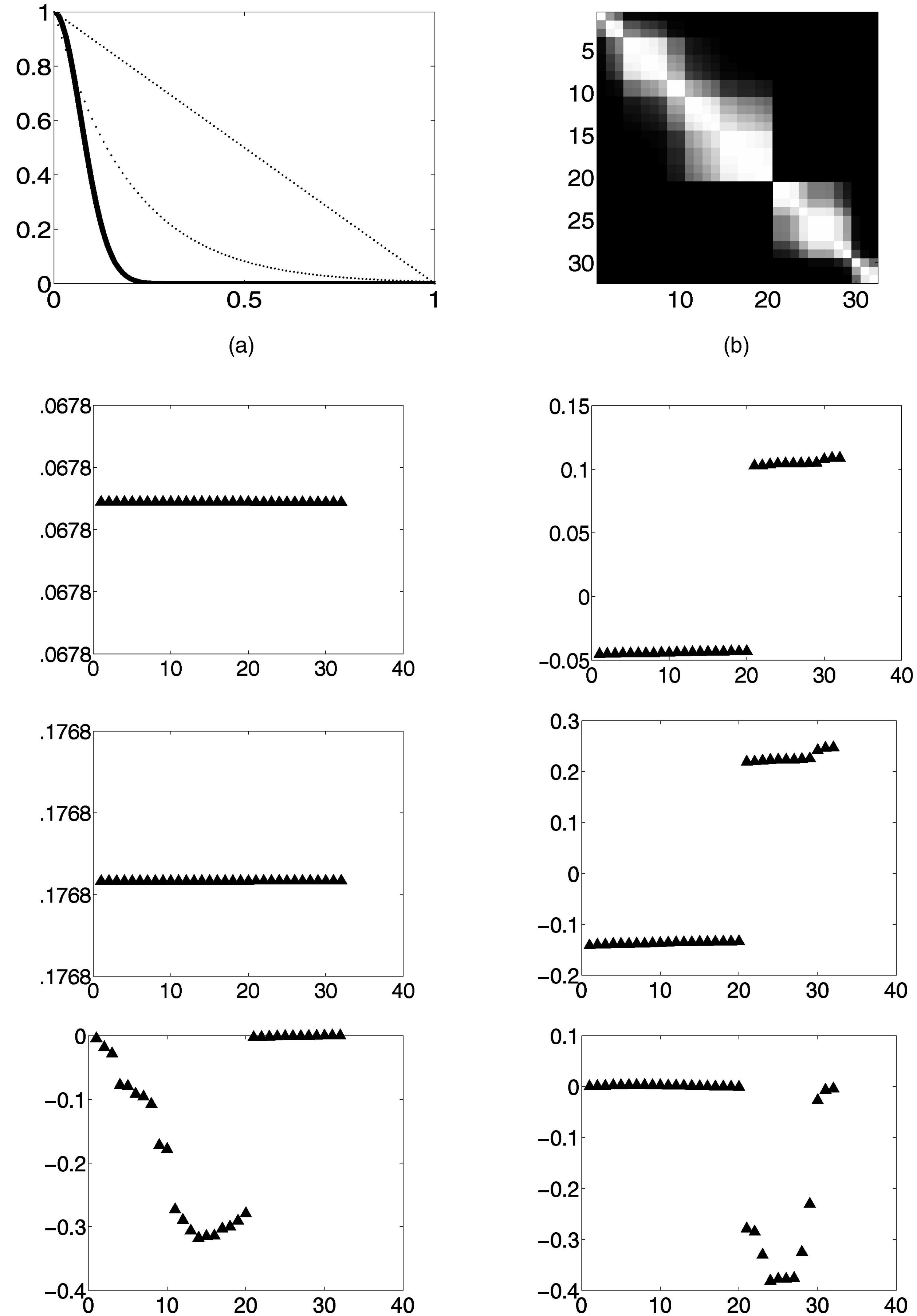
$$Wx = (1 - \lambda) D x$$

Average Cut:

$$(D - W)x = \lambda x$$

Average Association:

$$Wx = \lambda x$$



(a) (b) (c) (d)

$$w(x) = 1 - d(x)$$

Normalized Cut:

$$(D - W)x = \lambda D x$$

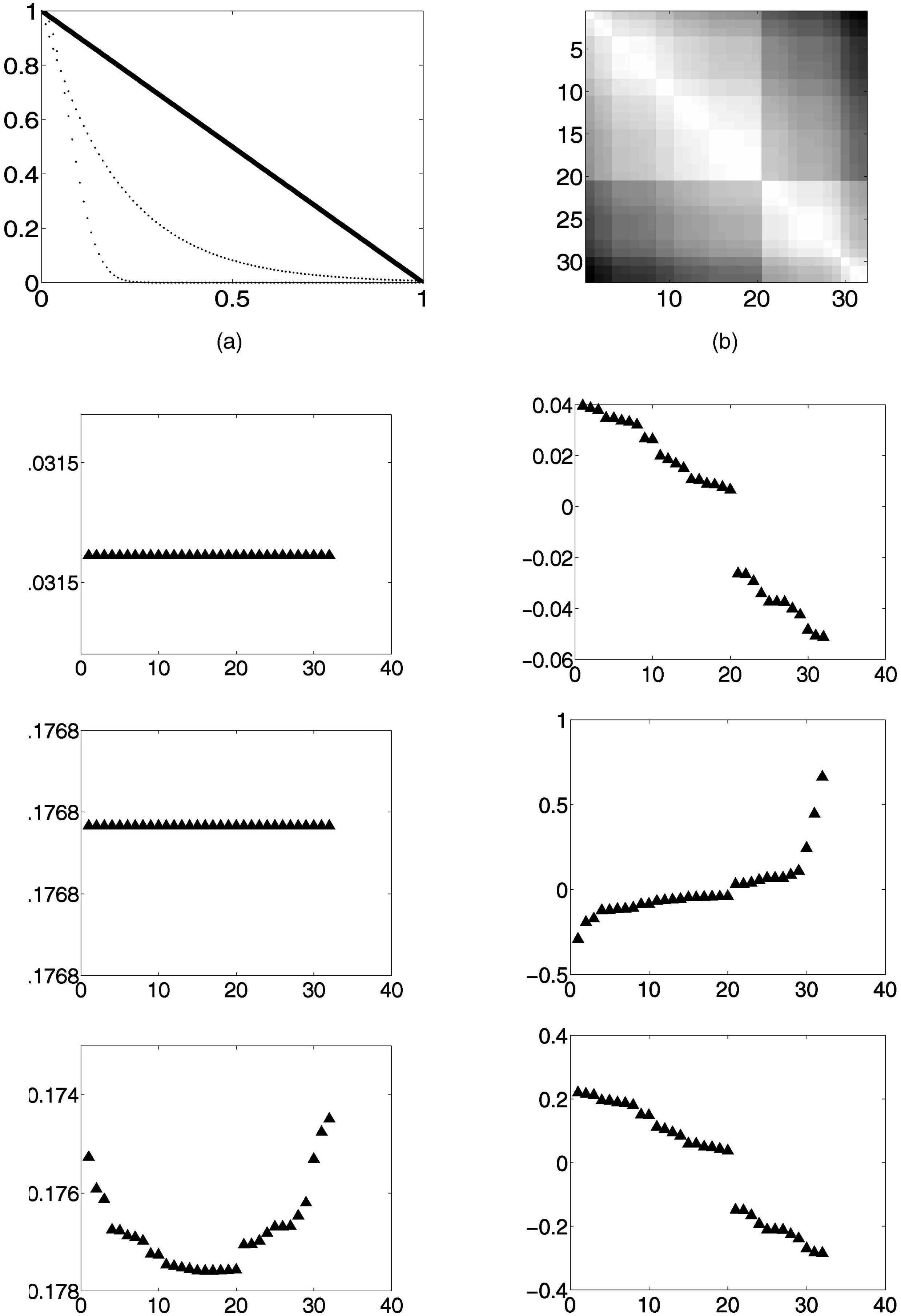
$$Wx = (1 - \lambda) D x$$

Average Cut:

$$(D - W)x = \lambda x$$

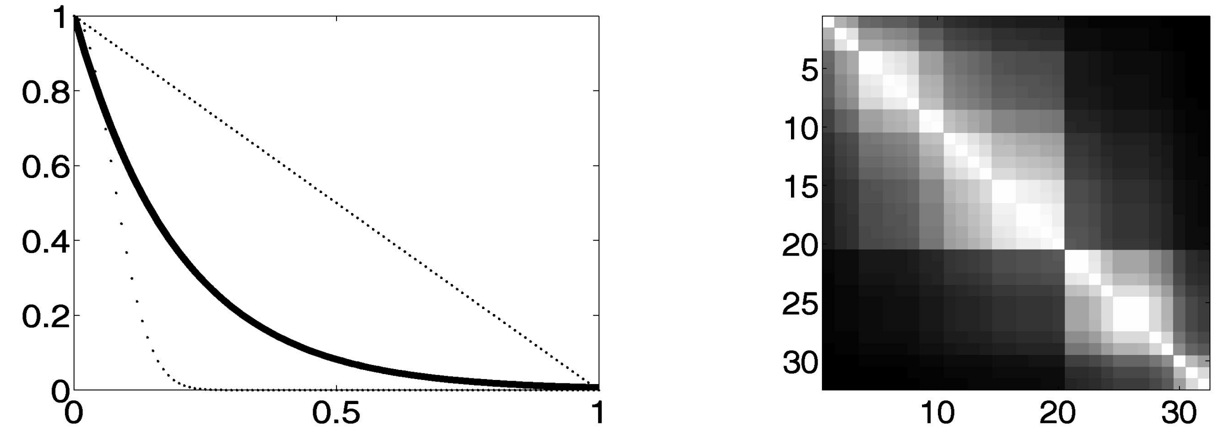
Average Association:

$$Wx = \lambda x$$



(c) (d)

$$w(x) = e^{-\left(\frac{d(x)}{0.2}\right)^2}$$



(a) (b)

Normalized Cut:

$$(D - W)x = \lambda D x$$

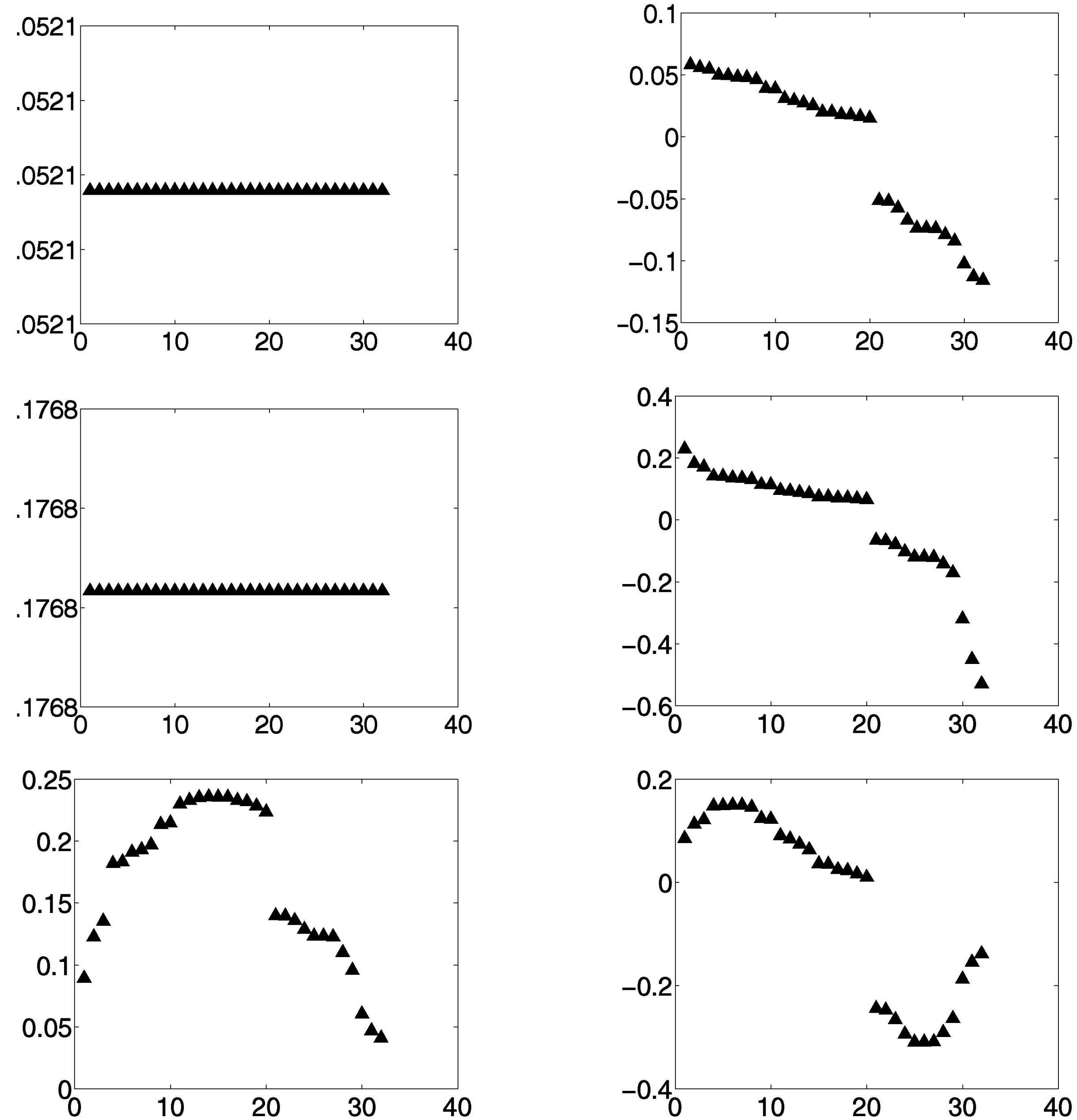
$$W x = (1 - \lambda) D x$$

Average Cut:

$$(D - W)x = \lambda x$$

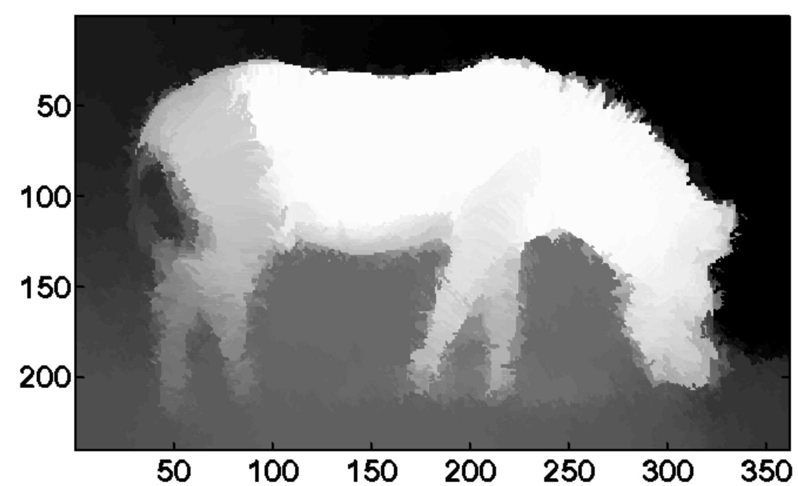
Average Association:

$$W x = \lambda x$$

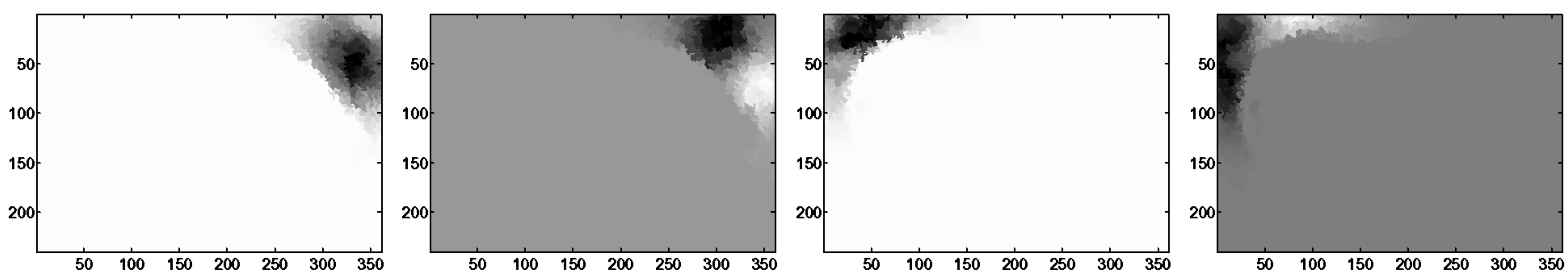


(c) (d)

Comparison Example



(a)



(b) (c) (d) (e)