Normalized Cuts and Image Segmentation

Jianbo Shi and Jitendra Malik

IEEE Transactions on Pattern Analysis and Machine Intelligence

Vol. 22, no. 8, August 2000

גיל ציוני
במסגרת קורס מתקדם בכריית נתונים, תשע"א
בהנחיית פרופ' אילן שמשוני

מבוא

<u>מטרה</u>: סגמנטציה של תמונות באופן משמעותי (בידוד קבוצות של פיקסלים).

- בהינתן ידע מוקדם על ה"עולם" בו צולמה התמונה (prior - מאפייני עצמים, תאורה, גוונים, זווית מבט, תכונות סימטריה וכו'), היה ניתן להשתמש בהסתברות בייסיאנית ליצירת החלוקה.
- אלגוריתם Normalized Cut אינו מניח את ידע מוקדם. הוא משתמש באפיוני התמונה עצמה (צבע, בהירות וכו') על מנת להחליט על החלוקה.
- יוצא מן הכלל – ההנחה כי פיקסלים מרוחקים בתמונה אינם שייכים לאותה קבוצה.
- האלגוריתם הופך את התמונה לגרף בו כל פיקסל הוא צומת.
- הסגמנטציה שמבצע האלגוריתם היא היררכית - יש לה מבנה של עץ היורד מ"התמונה הגדולה" לפרטים (dendrogram).
- אין חלוקה אחת "נכונה". השאלות אליהן צריך האלגוריתם להתייחס הן:

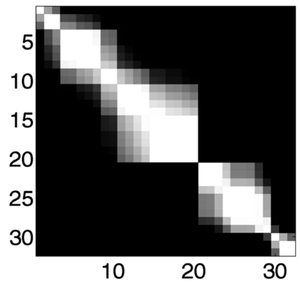
1. מהו קריטריון החלוקה האופטימאלי?

2. האם ניתן לממש אלגוריתם יעיל שיבצע חלוקה על פי קריטריון זה?

חלוקה אופטימאלית

- נציג את התמונה כגרף ממשקל בו כל פיקסל הוא צומת המחובר בקשתות לכל צומת (פיקסל) אחר בגרף.

- משקל הקשתות פרופורציונאלי לערך פונקצית הדמיון בין הצמתים (מחושב בהסתמך על תכונות הפיקסל ועל מרחק פיזי).



$W(i,j) = w_{ij}$

$$w_{ij} = e^{\frac{-\|\boldsymbol{F}_{(i)} - \boldsymbol{F}_{(j)}\|_2^2}{\sigma_I^2}} * \begin{cases} e^{\frac{-\|\boldsymbol{X}_{(i)} - \boldsymbol{X}_{(j)}\|_2^2}{\sigma_X^2}} & \text{if } \|\boldsymbol{X}(i) - \boldsymbol{X}(j)\|_2 < r \\ 0 & \text{otherwise.} \end{cases}$$

F(i) - וקטור תכונות המבוסס על עוצמה, צבע, מרקם או תנועה

X(i) – המיקום של פיקסל i

- נרצה לחלק את הגרף לשתי קבוצות זרות A, B של צמתים על ידי הסרת קשתות כך ש $A \cup B = V, A \cap B = \varnothing$ וגם:

1. הדמיון בין צמתים באותה קבוצה יהיה גדול (ע"פ מדד כלשהו)

2. הדמיון בין צמתים מקבוצות שונות יהיה קטן (ע"פ אותו מדד)

- המאמר מציע את ה Normalized Cut כקריטריון החלוקה.

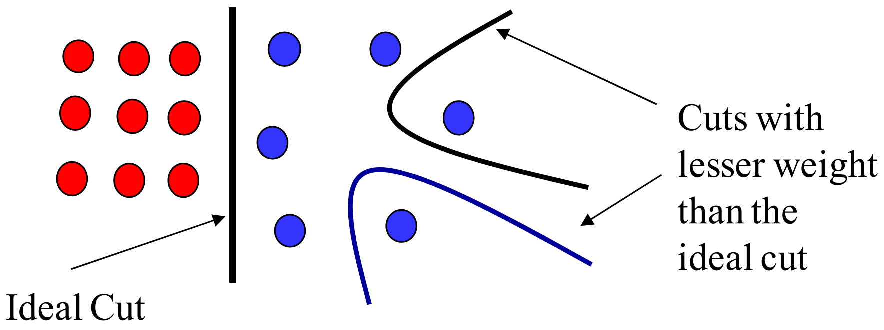
דמיון בין קבוצות: **minimum cut** (Wu and Leahy 1993)

- הדמיון בין A לבין B יוגדר כסכום משקלי הקשתות שהוסרו:

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

- האופטימיזציה של החלוקה לשתי קבוצות תושג על ידי מציאת
  $\text{Minimum}\{\text{cut}(A, B)\}$

- בעיה: נטייה ליצירת קבוצות קטנות (שאיפה להסרת כמות מינימאלית של קשתות)



Wu Z. and Leahy R. 1993. An Optimal Graph Theoretic Approach to Data Clustering: Theory and Its Application to Image Segmentation. *Pattern Analysis and Machine Intelligence* 15 (11): 1101-1113

- מבוא
- חלוקה אופטימאלית
  - בין קבוצות
  - בתוך קבוצה
  - בין ובתוך קבוצות
- יעילות חישובית
- רציפות/דיסקרטיות
- איטרטיביות
- האלגוריתם: 2-way
- ניסויים
- הרחבה: k-way
- Average cut/assoc.
- סיכום

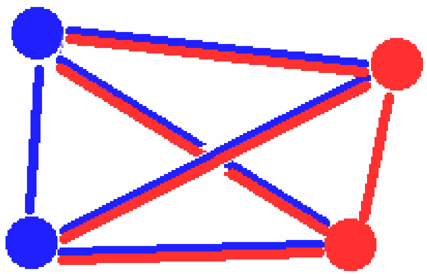
דמיון בין קבוצות: Normalized Cuts

- <u>תיקון</u>: נשקלל את cut(A, B) לפי משקל סה"כ הקשתות שיש להן צומת אחת לפחות בקבוצה A (או B), ושוב נחפש את המינימום.

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

- <u>במה שיפרנו</u>? אם נסתכל על האיור הקודם, הרי ש assoc(A,V) ≈ cut(A,B) לכן, אחד האגפים יהיה שווה בקירוב ל-1 <= *Ncut*(A, B) בדוגמא שבאיור לא יהיה מינימאלי.

- כאמור – זהו מדד לדמיון בין שני ה segments בגרף.

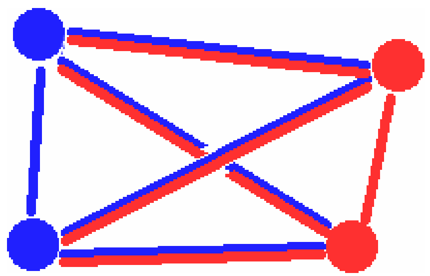


דמיון בתוך הקבוצות

באותו אופן :

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

- נשקלל את סכום משקל הקשתות המחברות בין שני צמתים מאותה קבוצה assoc(A, A) במשקל הקשתות המחברות בין צמתים מקבוצה זו לצמתים מהקבוצה האחרת assoc(A, V)
- נחפש את המקסימום.
- זהו מדד לדמיון בתוך segment יחיד בגרף.
- כנ"ל עם B



דמיון בין ובתוך הקבוצות

- יש קורלציה מלאה, ושלילית, בין שני המדדים:

$$\begin{aligned} Ncut(A,B) &= \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(A,B)}{assoc(B,V)} \\ &= \frac{assoc(A,V) - assoc(A,A)}{assoc(A,V)} \\ &\quad + \frac{assoc(B,V) - assoc(B,B)}{assoc(B,V)} \\ &= 2 - \left(\frac{assoc(A,A)}{assoc(A,V)} + \frac{assoc(B,B)}{assoc(B,V)}\right) \\ &= 2 - Nassoc(A,B). \end{aligned}$$

- מכאן – כשהאחד במינימום, השני במקסימום.

יעילות חישובית

Given a partition of nodes of a graph, V, into two sets A and B:

- Let **x** be an N = |V| dimensional indicator vector where:

$$\text{Xi} = \begin{cases} 1 & \text{if node } i \text{ is in A} \\ -1 & \text{otherwise.} \end{cases} \qquad X = [X_1, \ldots, X_N]$$

- Let $d_{(i)} = \sum_j w(i,j)$ be the total connection from node *i* to all other nodes.

$$D = \begin{pmatrix} d_1 & 0 & \cdots & 0 \\ 0 & d_2 & \ddots & \vdots \\ \vdots & \ddots & \ddots & 0 \\ 0 & \ldots & 0 & d_N \end{pmatrix}$$

- With the definitions **x** and **d**, we can rewrite *Ncut*(A,B) as:

$$\begin{aligned} Ncut(A,B) &= \frac{cut(A,B)}{assoc(A,V)} + \frac{cut(B,A)}{assoc(B,V)} \\ &= \frac{\sum_{(\boldsymbol{x}_i>0,\boldsymbol{x}_j<0)} -w_{ij}\boldsymbol{x}_i\boldsymbol{x}_j}{\sum_{\boldsymbol{x}_i>0} \boldsymbol{d}_i} \\ &\quad + \frac{\sum_{(\boldsymbol{x}_i<0,\boldsymbol{x}_j>0)} -w_{ij}\boldsymbol{x}_i\boldsymbol{x}_j}{\sum_{\boldsymbol{x}_i<0} \boldsymbol{d}_i}. \end{aligned}$$

- נגדיר: $y = (1 + x) - b(1 - x)$ $\qquad b = \dfrac{\sum_{x_i>0} d_i}{\sum_{x_i<0} d_i}$

- ניתן להראות כי:

$$min_{\boldsymbol{x}} Ncut(\boldsymbol{x}) = min_{\boldsymbol{y}} \frac{\boldsymbol{y}^T(\mathbf{D}-\mathbf{W})\boldsymbol{y}}{\boldsymbol{y}^T\mathbf{D}\boldsymbol{y}},$$

with the condition $\boldsymbol{y}(i) \in \{1, -b\}$ and $\boldsymbol{y}^T\mathbf{D}\mathbf{1} = 0$.

- אם "*y is relaxed*" כך שיקבל ערכים רציפים אזי ניתן למצוא את המינימום שלו על ידי פתרון ה generalized eigenvalue system (מסתמך על Rayleigh quotien)

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

- ותחת האילוצים על y ניתן להמיר ל standard eigenvalue system

$$D^{-1/2}(D-W)D^{-1/2}g = \lambda g$$

{λ – ערך עצמי $\qquad$ g – וקטור עצמי $\qquad$ D – מטריצה אלכסונית של d_i $\qquad$ W – מטריצת דמיון}

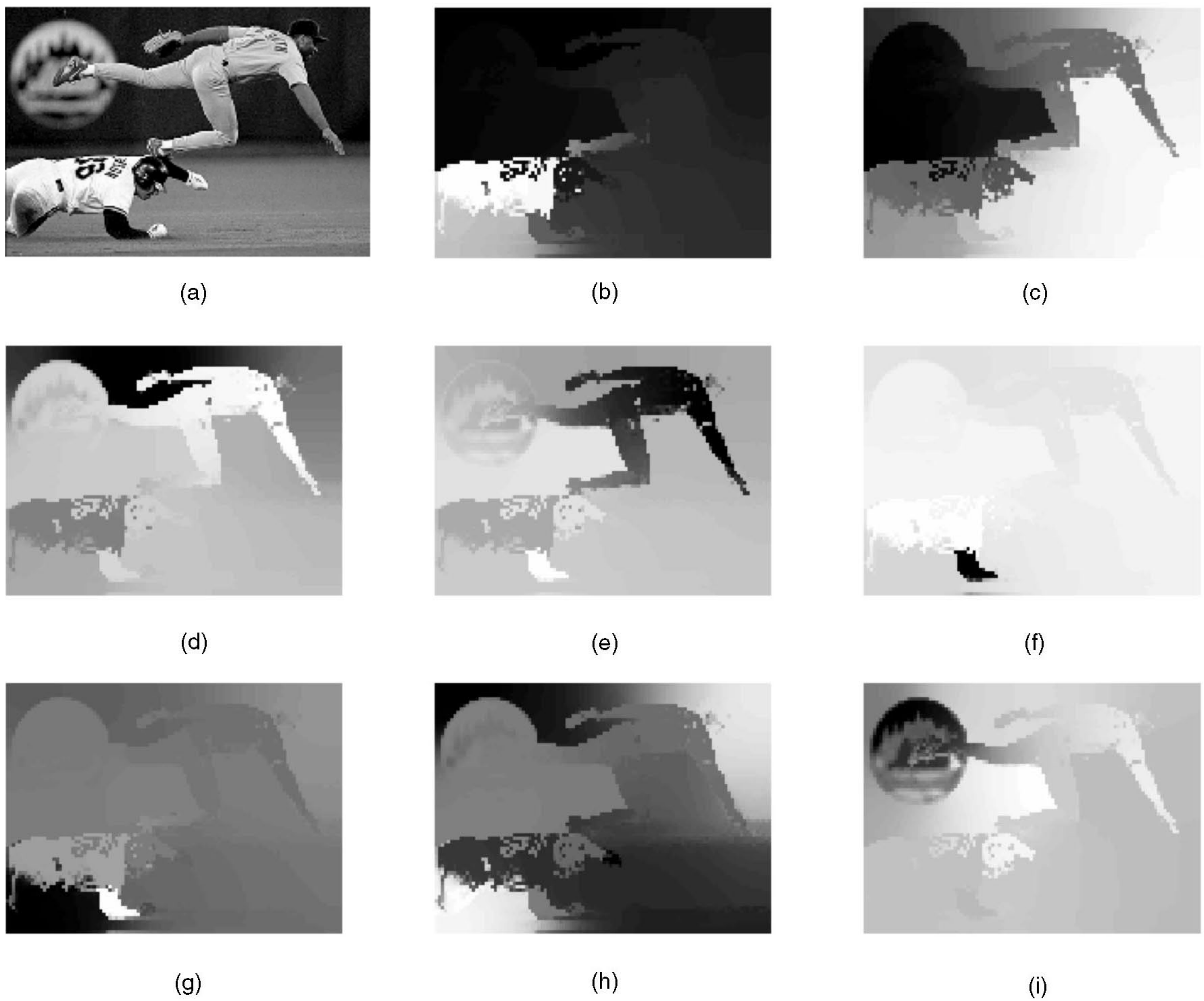
- ניתן לפתור {ב $O(n^3)$ } ולהראות כי :
- הווקטור העצמי עם הערך העצמי הקטן ביותר הוא $x_0=1$ עם ערך עצמי 0 (פתרון טריביאלי, לא שמיש עבור האלגוריתם).
- הווקטור העצמי השני מאונך לראשון ומחושב על ידי : $0 = z_1^T z_0 = x_1^T D1$

מסקנות:

- הווקטור העצמי עם הערך העצמי השני מלמטה מהווה את החלוקה האופטימאלית של התמונה לשתי קבוצות זרות A, B (בהסתמך על Rayleigh quotient).
- באופן דומה, הווקטור העצמי עם הערך העצמי השלישי מלמטה מהווה את החלוקה האופטימאלית הבאה בתור בהיררכיה, וכן הלאה על פי סדר הווקטורים העצמיים.

פרקטיקה:

- היעילות החישובית נמוכה בהרבה מ $O(n^3)$ בגלל :
  1. מטריצת הדמיון הדלילה.
  2. אנו מחפשים רק את הווקטורים העצמיים הראשונים.
  3. לא דרוש דיוק רב, מספיק לדעת אם הערך בווקטור חיובי או שלילי.
- אם נשתמש ב Lanczos eigensolver נקבל $\sim O(n^{1.5})$



Subplots (b)-(i) show the eigenvectors corresponding the second smallest to the ninth smallest eigenvalues of the system. The eigenvectors are reshaped to be the size of the image.

רציפות/דיסקרטיות

- הווקטור העצמי שמתקבל הוא רציף, ולא 1/1- כפי שהיינו רוצים.
- צריך לקבוע כלל החלטה שיבצע התמרה לווקטור של ערכים בדידים (0, חציון...).
- ההצעה במאמר:
  - לחלק את התחום [Min(X_i), Max(X_i)] ל-L מקטעים שווים
  - לבדוק לכל אחד ציון $Ncut$
  - לבחור את L שיביא ל Minimum($Ncut$) ככלל ההחלטה.
- קריטריון היציבות:
  - נקבע עד כמה הווקטור העצמי "חלק" (יחס max/min בהיסטוגרמה).
  - ווקטור "חלק" מעיד על כך שהמשך החלוקה כרוך באי-ודאות רבה
  - נקבע סף מינימום לקריטריון היציבות. לא נשתמש בווקטור עם ערך יציבות נמוך מהסף.
  - <u>חיסרון</u>: אובדן אינפורמציה שאולי קיימת בווקטורים הבאים בתור.

אלגוריתם איטרטיבי

- אמינות החלוקה של הווקטורים העצמיים הולכת ויורדת ככל שהווקטור מדרג גבוה יותר כי הטעויות הולכות ומצטברות (בעיקר בשל המעבר למודל רציף).
- לכן רצוי בכל שלב "לאתחל" את האלגוריתם ולהתייחס ל A בנפרד מ B, כל אחד מהם כקלט חדש (כל קבוצה היא גרף נפרד עליו נפעיל איטרטיבית את האלגוריתם).
- כלל ההחלטה לעצירה : קריטריון היציבות של הווקטור העצמי השני מתחת לסף שנקבע.

האלגוריתם: Recursive Two-Way Ncut

1. Given a set of features, set up a weighted graph G = (V, E), compute the weight on each edge, and summarize the information into W and D.

2. Solve $(D-W)y=\lambda Dy$ for eigenvectors with the smallest eigenvalues.

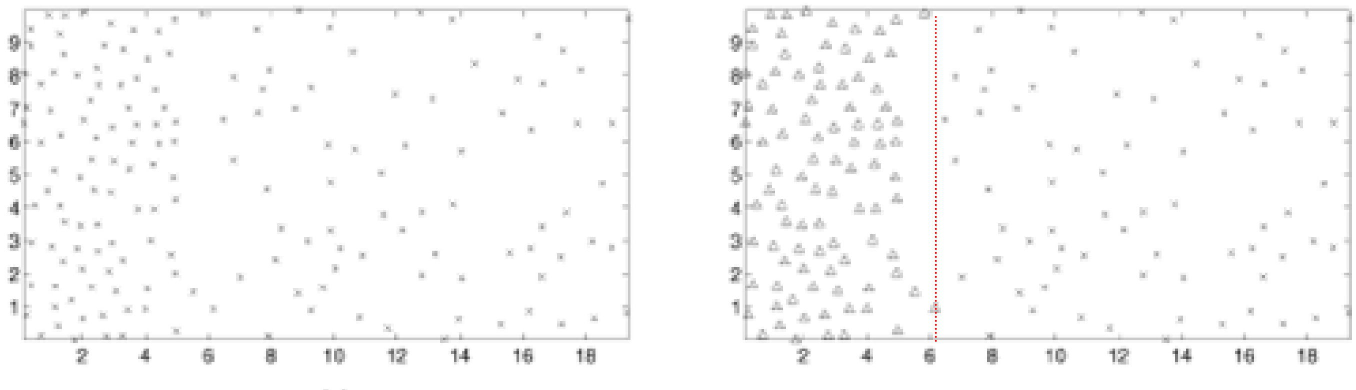
3. Use the eigenvector with the 2^{nd} smallest eigenvalue to bipartition the graph
   - ➢ Find the splitting point that minimizes *Ncut*

4. Check stability and *Ncut* threshold to decide whether to divide the current partition.

5. Recursively repartition the segmented parts if necessary.

Experiments (I)

Point set segmentation



(a) Point set generated by Poisson process (2.5, 1.0). (b) Segmentation results (σ=5, r=3).

- "The edge weight W_{ij} between node i and j as the product of a feature similarity term and spatial proximity term"

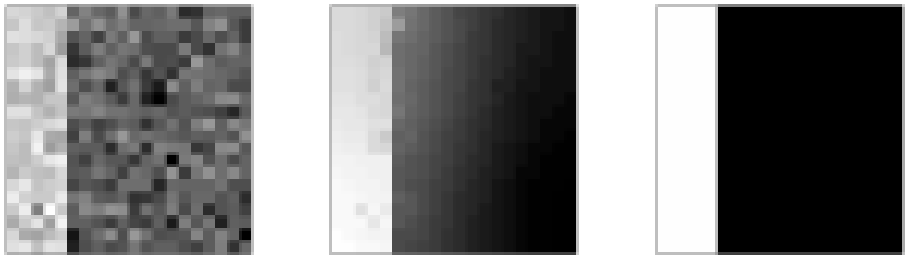
$$w_{ij} = e^{\frac{-\|F_{(i)} - F_{(j)}\|_2^2}{\sigma_I^2}} * \begin{cases} e^{\frac{-\|X_{(i)} - X_{(j)}\|_2^2}{\sigma_X^2}} & \text{if } \|X(i) - X(j)\|_2 < r \\ 0 & \text{otherwise.} \end{cases}$$

- "The value of σ is typically set to 10 to 20 percent of the total range of the feature distance function d(x)".
- "The exponential weighting function is chosen here for its relative simplicity"

- מבוא
- חלוקה אופטימאלית
  - בין קבוצות
  - בתוך קבוצה
  - בין ובתוך קבוצות
- יעילות חישובית
- רציפות/דיסקרטיות
- איטרטיביות
- האלגוריתם : 2-way
- **ניסויים**
- הרחבה : k-way
- Average cut/assoc.
- סיכום

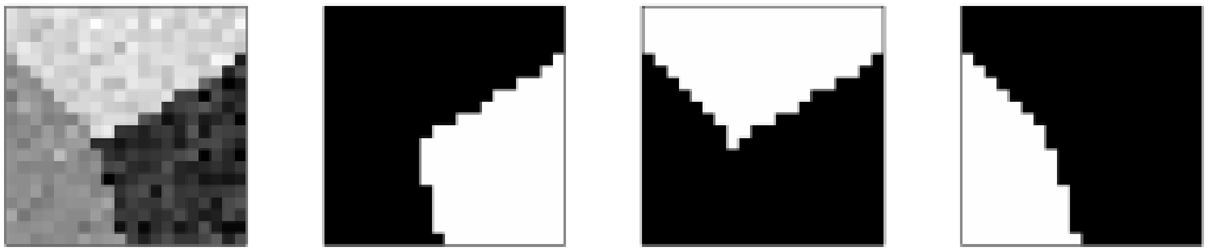
Experiments (II)

Synthetic images



(a) (b) (c)

(a) A synthetic image showing a noisy "step" image. Intensity varies from 0 to 1, and Gaussian noise with σ=0.2 is added. (b) shows the eigenvector with the second smallest eigenvalue and subplot (c) shows the resulting partition.



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

(a) A synthetic image showing three image patches forming a junction. Image intensity varies from 0 to 1 and Gaussian noise with σ=0.1 is added. (b)-(d) show the top three components of the partition.

Experiments (III)

Weather radar

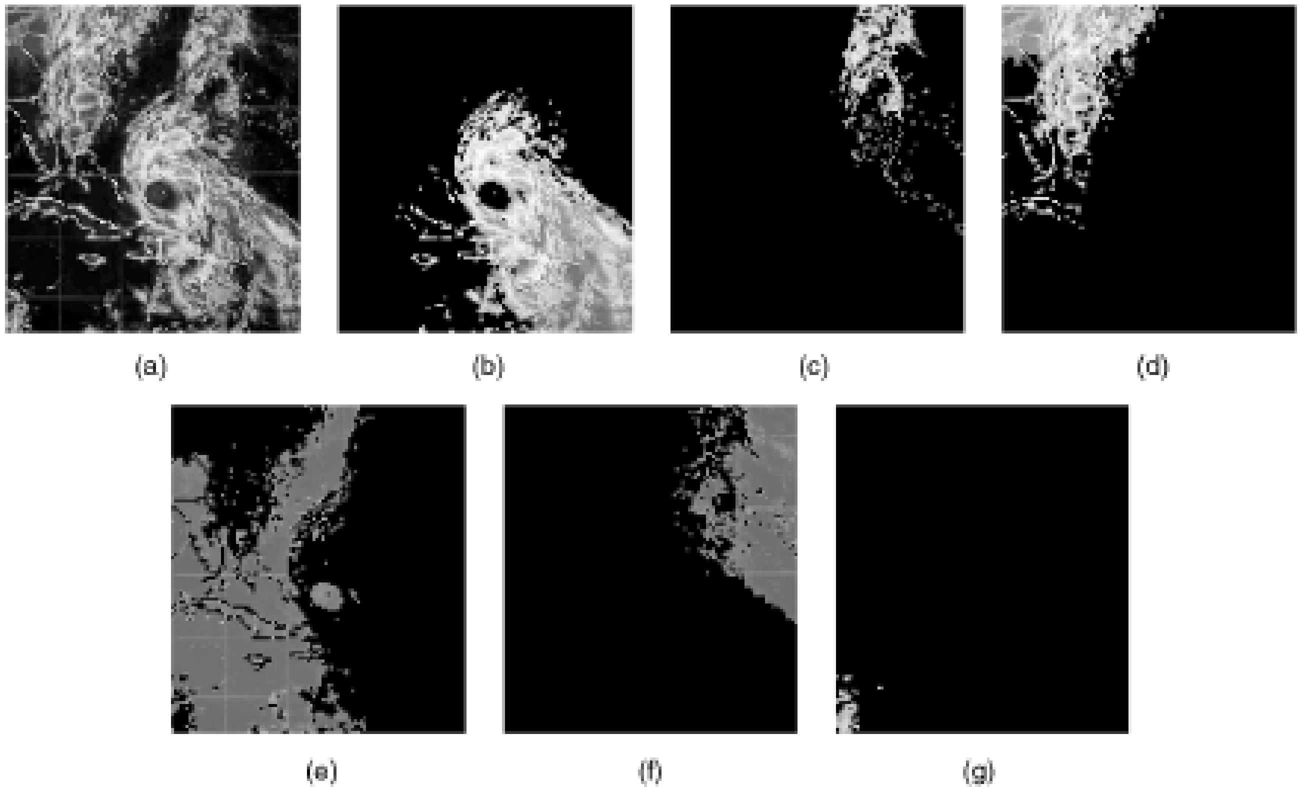
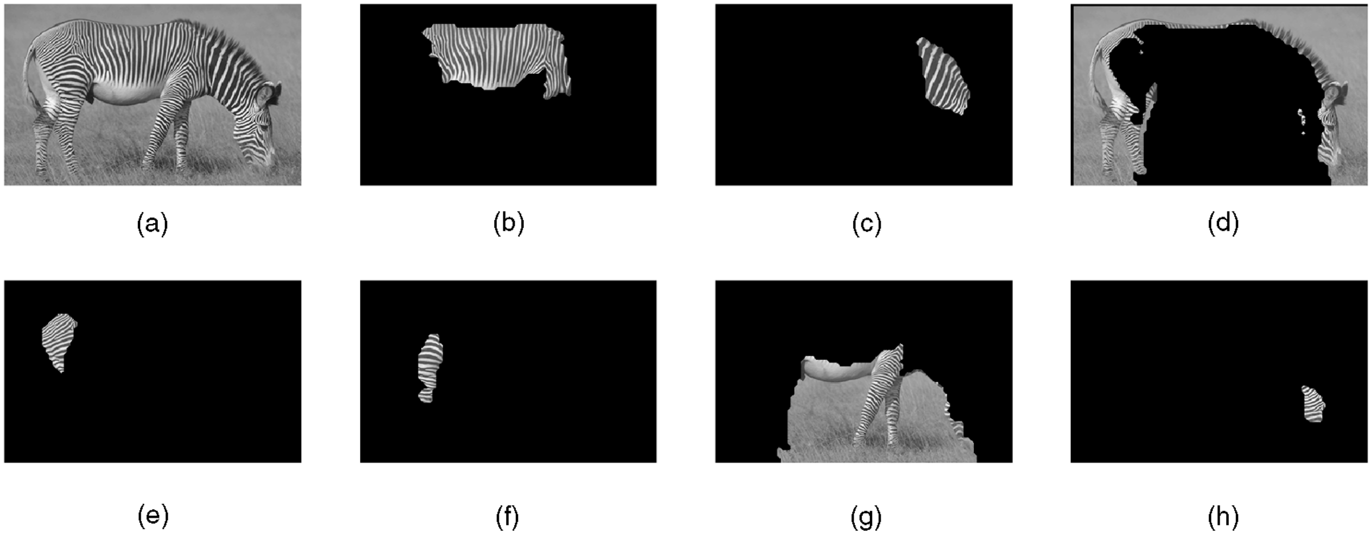


Fig. 8. (a) shows a 126X106 weather radar image. (b)-(g) show the components of the partition with Ncut value less than 0.08. Parameter setting: σ_I=0.007, σ_x=15.0, r=10.

Experiments (IV)

Texture segmentation

“Note that the measure we have used is orientation-variant and, therefore, parts of the zebra skin with different stripe orientation should be marked as separate regions”.



(a) shows an image of a zebra. The remaining images show the major components of the partition. The texture features used correspond to convolutions with DOOG filters at six orientations and five scales.

הרחבה לאלגוריתם: k-way

- המוטיבציה: לא לאבד את האינפורמציה הנמצאת בווקטורים העצמיים 3 והלאה (חיסרון של קריטריון היציבות).
- שתי אפשרויות:

1. א) ליצור over segmentation של הגרף על ידי שימוש ב n הווקטורים העצמיים הראשונים.

ב) להריץ על התוצאה אלגוריתם k-means שיאחד סיגמנטים ל k קבוצות (cumulative Aggregation), כך שנביא למינימום את:

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

2. א) ליצור k קבוצות בעזרת אלגוריתם k-means (k < 100)

ב) להגדיר כל קבוצה כצומת בגרף המצומצם; להגדיר את משקל הקשת בין שני צמתים כ- $W(i,j) = assoc(A_i, A_j)$

ג) להריץ אלגוריתם 2-way רגיל (או לבדוק סדרתית את החלוקות, אם המטריצה מספיק קטנה).

Average association

- נרצה למצוא מקסימום ל

$$\frac{assoc(A,A)}{|A|}+\frac{assoc(B,B)}{|B|}$$

* lAl - מספר הצמתים בקבוצה A

בעזרת הווקטורים העצמיים שמתאימים לפתרון בעיה זו.

- נטייה ליצירת קבוצות קטנות ו"צפופת" (clustering מול סגמנטציה).
- מתאים אם יש prior על התפלגות נורמאלית.

Average cut

- נרצה למצוא מינימום ל

$$\frac{cut(A,B)}{|A|}+\frac{cut(A,B)}{|B|}$$

- התוצאות דומות לאלו של *Ncut*.
- אבל : אין קורלציה שלילית עם Average association (קיימת ב *Ncut*) ולכן אין בטחון במידת הקרבה שבתוך הקבוצות.

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 $W x = (1-\bar{\lambda}) D x$	$(D-W)\,x = \bar{\lambda} x$

- מבוא
- חלוקה אופטימאלית
  - בין קבוצות
  - בתוך קבוצה
  - בין ובתוך קבוצות
- יעילות חישובית
- רציפות/דיסקרטיות
- איטרטיביות
- האלגוריתם : 2-way
- ניסויים
- הרחבה : k-way
- **Average cut/assoc.**
- סיכום

סיכום

- אלגוריתם פשוט יחסית.
- אין הנחת Prior.
- התוצאות הראשוניות מעודדות (היו בשנת 2000. המחקר בברקלי נמשך. יישומים??).
- ניתן לשלב אותו בקלות עם אלגוריתמים אחרים.

תכונות:

- דורש זיכרון רב.
- זמן חישוב $O(n \cdot \sqrt{n})$ בקירוב.
- נטייה ליצור סגמנטים שווי גודל.
- בעיה עם רקע צבעוני (textured backgrounds).

פרמטרים תלויי אדם:

- פונקצית הדמיון:
  - אילו תכונות נמדדות
  - מהו המרחק המקסימאלי בין פיקסלים באותו סגמנט.
- תנאי הסף לטרנספורמציה של הווקטור העצמי מהמרחב הרציף לבדיד.
- קריטריון ההחלקה - תנאי הסף לקריאה רקורסיבית.

Experiments (IV)

Motion segmentation

(a) (b)

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

Given an image sequence, a weighted graph is constructed by taking each pixel in the image sequence as a node and connecting pixels that are in the spatiotemporal neighborhood of each other.