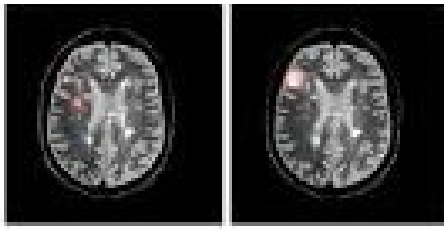
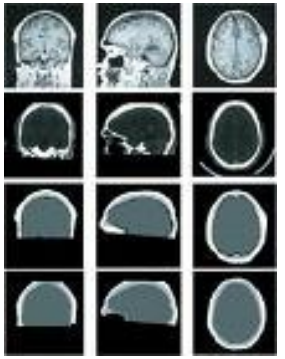
Phân Đoạn Ảnh
(Image Segmentation)

Giới thiệu

- Phân đoạn ảnh: là quá trình phân chia một ảnh số thành nhiều vùng khác nhau, trong đó mỗi vùng sẽ mang một ngữ nghĩa nhất định phục vụ cho các bước tiếp theo trong quá trình phân tích ảnh.
- Mục tiêu của phân đoạn ảnh là làm đơn giản hóa ảnh đầu vào sao cho chúng ta dễ dàng nhận biết vị trí và đường bao của mục tiêu cần quan tâm trên ảnh.
- Mỗi pixel trong một vùng của kết quả phân đoạn ảnh phải có tính chất giống nhau như về màu sắc, cường độ sáng, hay vân trong đó vân có thể là kết quả của một bộ lọc đặc trưng.
- Các vùng khác nhau sau khi phân đoạn ảnh phải có tính chất hoàn toàn khác biệt nhau
- Các ứng dụng của phân đoạn ảnh như
  - Tìm kiếm và phát hiện đối tượng trong ảnh
  - Trích xuất đối tượng có trong ảnh
  - Tiền xử lý cho các bài toán

  nhận dạng





Kết quả của phân đoạn ảnh

- Phân đoạn ảnh giúp giảm số lượng pixel lưu trữ trong ảnh
- Phân đoạn ảnh giúp quản lý ảnh thành từng vùng (hay từng đối tượng) thay vì từng pixel
- Phân đoạn ảnh ở mức tổng quát là sự phân biệt giữa foreground và background
- Trường hợp đơn giản nhất của phân đoạn ảnh là nhị phân hóa một ảnh mức xám

Nhóm dựa trên phương pháp gom nhóm

- Sử dụng phương pháp gom nhóm để gom các nhóm nhỏ thành các nhóm lớn hơn thỏa cùng một độ đo đặc trưng
- Gom nhóm sẽ được thực hiện quanh các hạt nhân được xác định ban đầu dựa trên các đặc trưng (mức xám, vân, màu)

Ưu điểm:

- Cho kết quả nhanh và thường ít bị ảnh hưởng bởi nhiễu

Nhược điểm:

- Xác định các điểm hạt nhân
- Khởi tạo các điểm hạt nhân khác nhau sẽ cho kết quả khác nhau

Tích hợp dựa trên intensity:

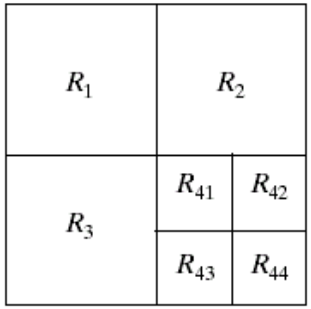
10	10	10	10	10	10	10
10	10	10	69	70	10	10
59	10	60	64	59	56	60
10	59	10	60	70	10	62
10	60	59	65	67	10	65
10	10	10	10	10	10	10
10	10	10	10	10	10	10

Tính chất homogeneity :

- **Giá trị intensity của các pixel không sai biệt nhau quá T đơn vị**
- **Xét các lân cận 4 hay 8 (trên, dưới, trái, phải và 2 cạnh chéo)**

Phân đoạn dựa trên phân chia và tích hợp vùng

- Phân chia vùng
  - Phân chia vùng từ một tập các điểm hạt nhân.
  - Một cách khác là xem ảnh là một vùng và tiến hành phân chia các vùng theo một tỉ lệ xác định.
- Tích hợp các vùng
  - Quá trình này đi ngược lại với quá trình phân chia trên.
  - Bắt đầu với vùng có kích thước nhỏ và tiến hành gom nhóm các vùng nhỏ thành vùng lớn hơn nếu các vùng nhỏ có cùng đặc trưng (màu, mức xám hay vân).
  - Thông thường quá trình phân chia và tích hợp được tiến hành xen kẽ qua từng bước lặp.

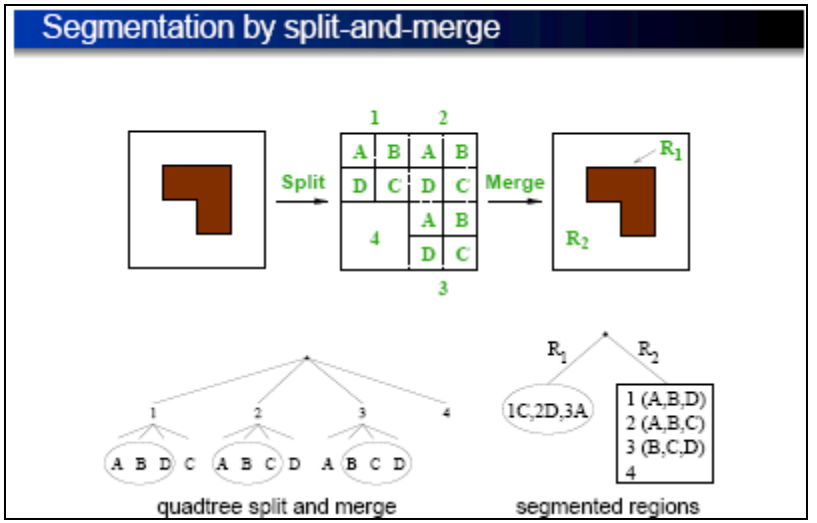


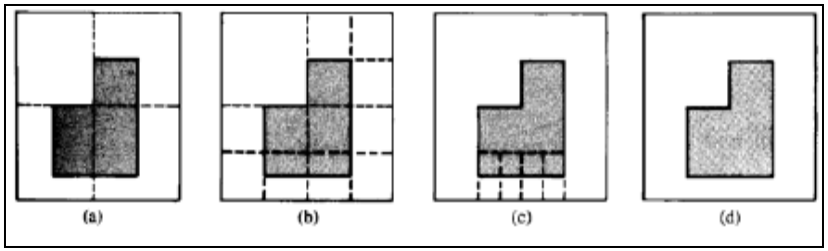
Ví dụ về cách phân chia – tích hợp

- Thuật toán có 2 tiến trình:
  - top-down: Chia ảnh thành các vùng đồng nhất, dựa trên cây tứ phân
  - bottom-up: Kết hợp các vùng lân cận có tính chất tương tự nhau
- Thuật toán:

  Top-down
  - Chia một ảnh thành 4 phần và mỗi phần thành 4 vùng nếu các phần chưa đồng nhất
  - Quá trình chia sẽ dừng nếu thu được tất cả các phần đồng nhất : *P(R) = TRUE*, kết quả nhận được là cây tứ phân

  Bottom-up
  - Tại mỗi mức của cây tiến hành tích hợp các vùng lân cận R_i *and* R_j nếu 2 vùng này đồng nhất $P(R_i,R_j) = TRUE$
- Dừng thuật toán khi không thể phân chia hay hợp nhất thêm vùng nào nữa





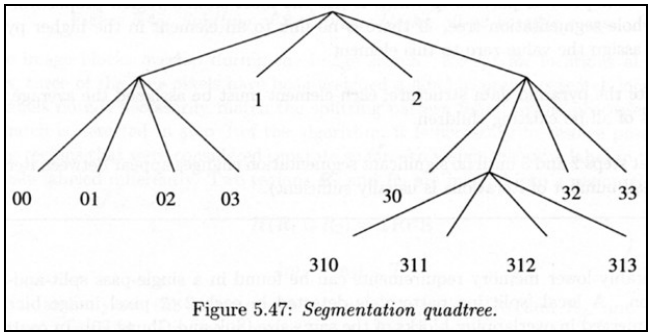
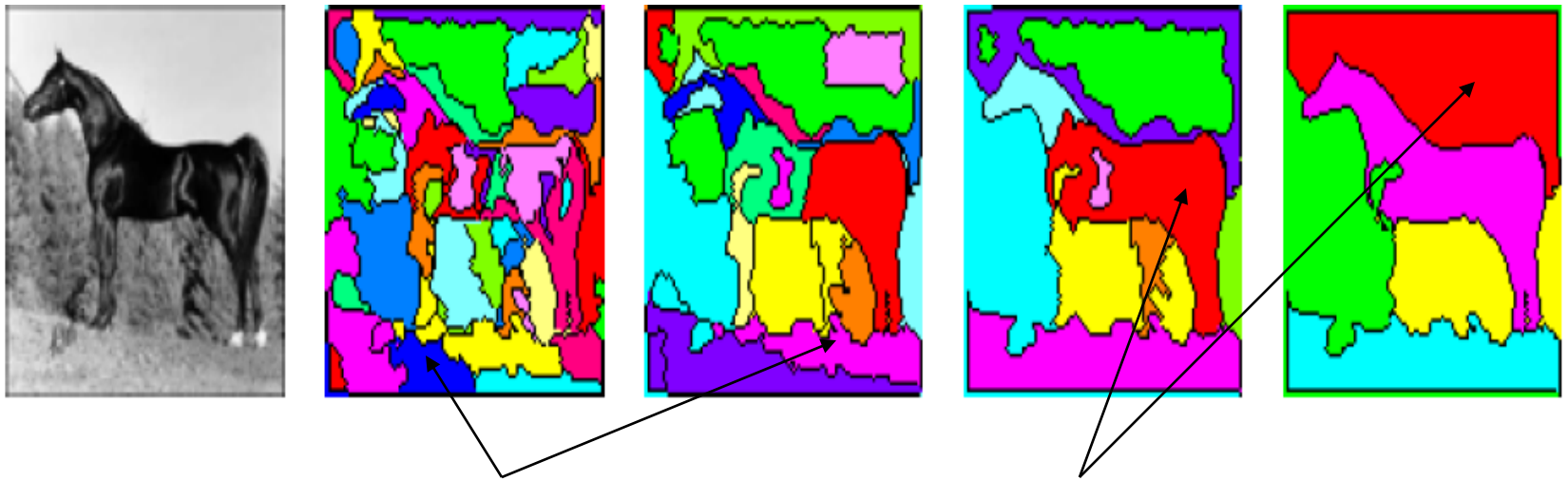


Figure 5.47: *Segmentation quadtree.*

Một số vấn đề trong cách tiếp cận

- Lựa chọn hạt nhân như thế nào ? Thông thường hạt nhân là điểm trung tâm của các tập điểm chiếm ưu thế trong ảnh.
- Làm thế nào để xác định tính tương đồng? Thông thường tính tương đồng phải có hai thể hiện về không gian và về nội dung.
- Làm sao để xác định điều kiện dừng cho thuật toán ? Trong cách tiếp cận này điều kiện dừng đóng vai trò quan trọng vì không phải lúc nào thuật toán cũng có thể hội tụ đến kết quả duy nhất.

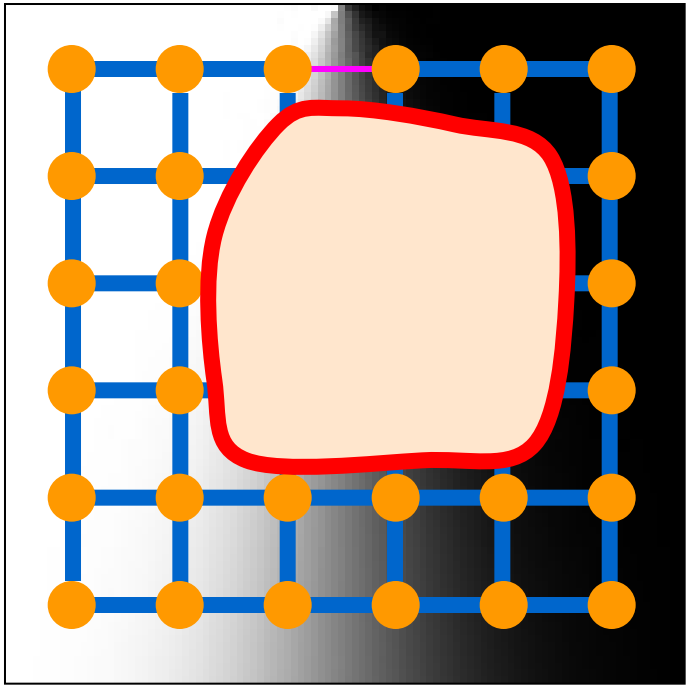
Đánh giá phân đoạn ảnh



Tính đồng nhất với các pixel cùng nhóm + Tính sai biệt/phân biệt với các pixel ở bên ngoài nhóm = **Saliency**

Graph-Cut

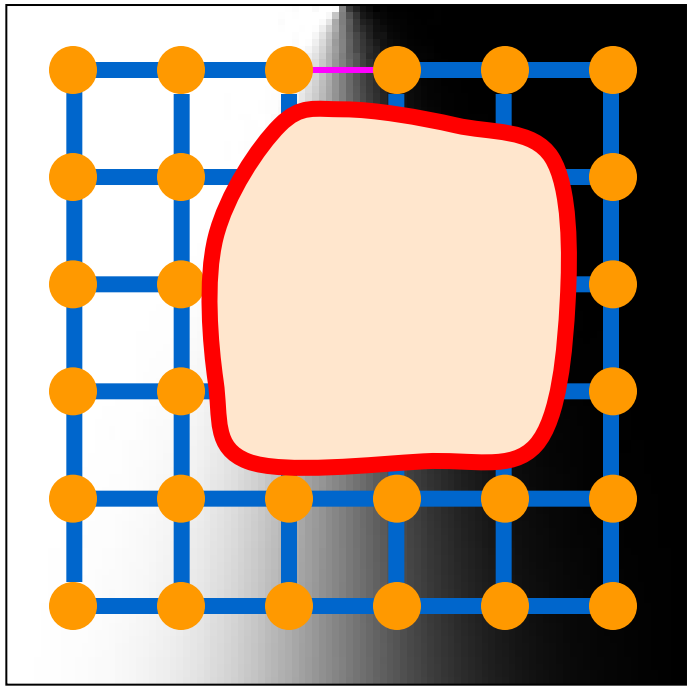
$$u_i = \begin{cases} 1 & i \in S \\ 0 & i \notin S \end{cases}$$



Graph-Cut

$$E(S) = \sum_{i \neq j} w_{ij} (u_i - u_j)^2$$

$$u_i = \begin{cases} 1 & i \in S \\ 0 & i \notin S \end{cases}$$



Graph-Cut

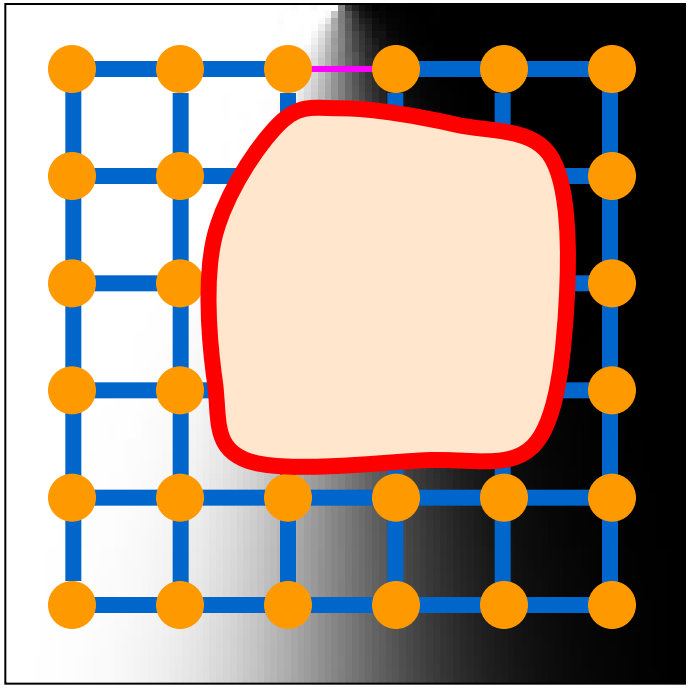
$$E(S) = \sum_{i \neq j} w_{ij}(u_i - u_j)^2$$

$$u_i = \begin{cases} 1 & i \in S \\ 0 & i \notin S \end{cases}$$

$$N(S) = \sum w_{ij} u_i u_j$$

E(S) : mối liên kết giữa các node trong S và các node lân cận không nằm trong S.

N(S) : mối liên kết giữa các node trong S.



Graph-Cut

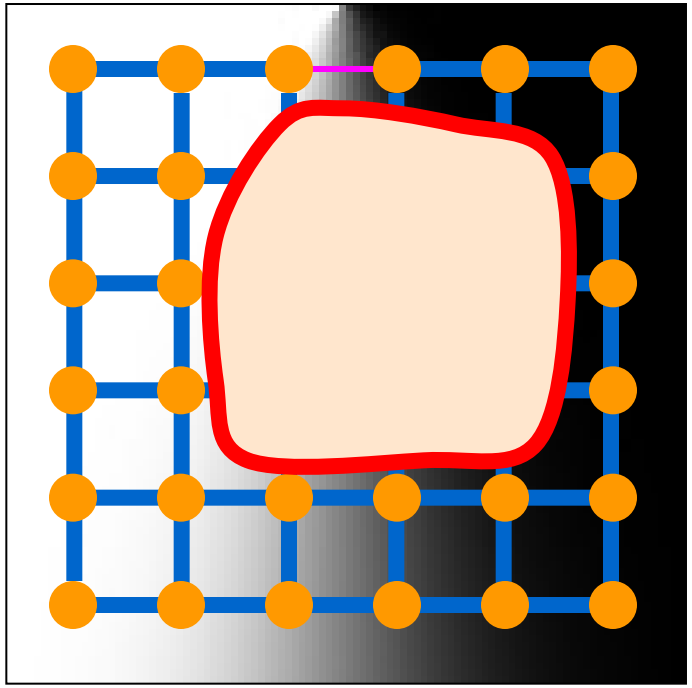
$$E(S) = \sum_{i \neq j} w_{ij} (u_i - u_j)^2$$

$$u_i = \begin{cases} 1 & i \in S \\ 0 & i \notin S \end{cases}$$

$$N(S) = \sum w_{ij} u_i u_j$$

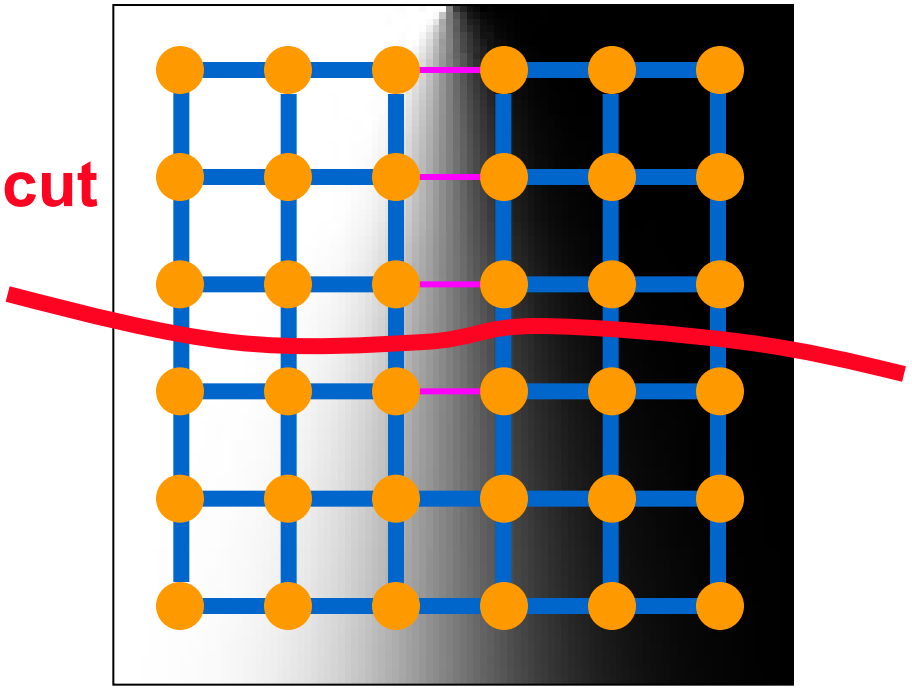
Minimize:

$$\Gamma(S) = \frac{E(S)}{N(S)}$$



Graph-Cut

High-energy cut



Minimize:

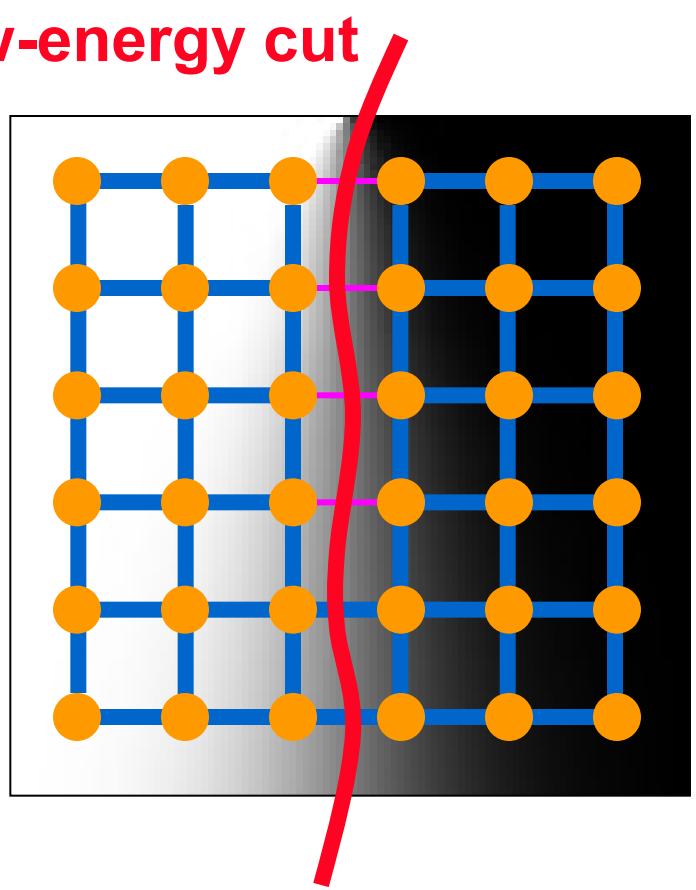
$$\Gamma(S) = \frac{E(S)}{N(S)}$$

Graph-Cut

Minimize:

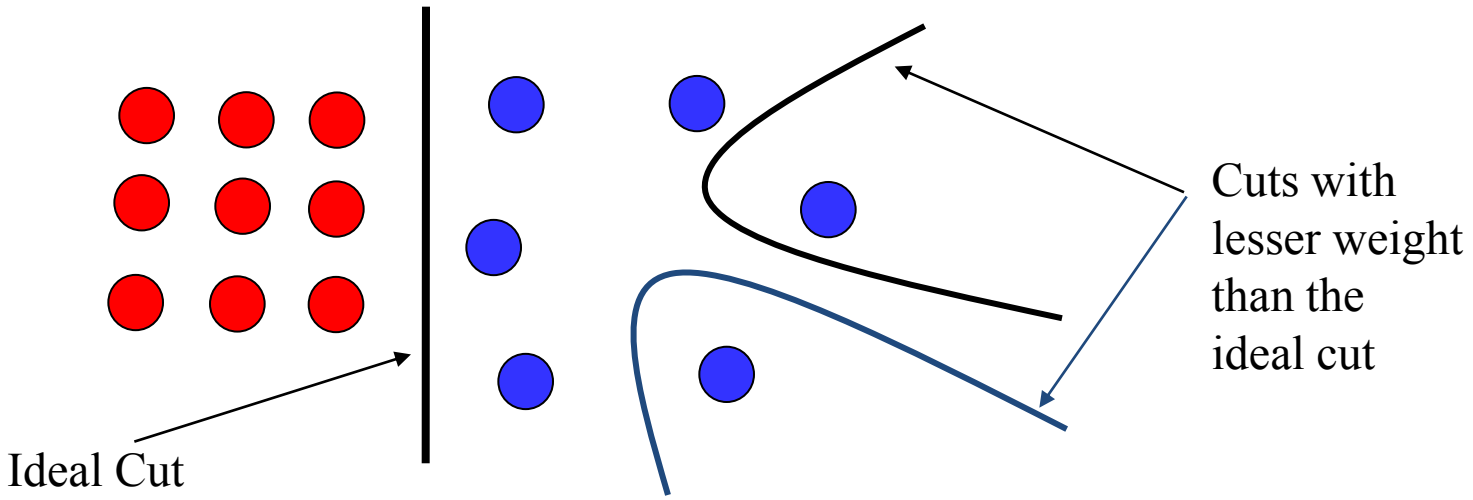
$$\Gamma(S) = \frac{E(S)}{N(S)}$$

Việc duyệt và tìm hết tất cả phân đoạn S để thỏa điều kiện trên là bài toán NP-hard



Normalized cut

- Khuyết điểm Graph-Cut: hơn nữa việc cực tiểu hóa trọng số trên có thể gây ra oversegmentation (là phân chia quá nhiều phân đoạn có kích thước nhỏ)



Normalized cut

- Để hạn chế nhược điểm trên có thể chuẩn hóa sự phân chia bằng trọng số của các cạnh tồn tại trong phân đoạn
- The *normalized cut* cost is:

$$\frac{w(A,B)}{vol(A)} + \frac{w(A,B)}{vol(B)}$$

w(*A*, *B*) = Tổng trọng số các cạnh nối giữa A và B, trong đó B chính là phần bù của A, công thức trên giúp cân bằng volumn giữa A và B khi tiến hành lựa chọn segment.

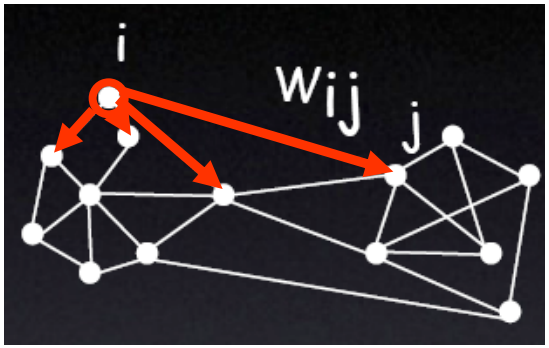
Vol(x) = V*olume* của *x.*

J. Shi and J. Malik. Normalized cuts and image segmentation. PAMI 2000

Trọng số cạnh

- Ma trận trọng số: $W = \lfloor w_{i,j} \rfloor$

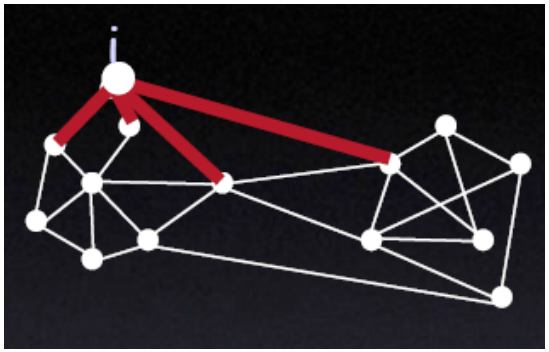
$$w_{i,j} = e^{\frac{-\left\| X_{(i)} - X_{(j)} \right\|_2^2}{\sigma_X^2}}$$



Bậc của đỉnh

- Bậc của đỉnh:

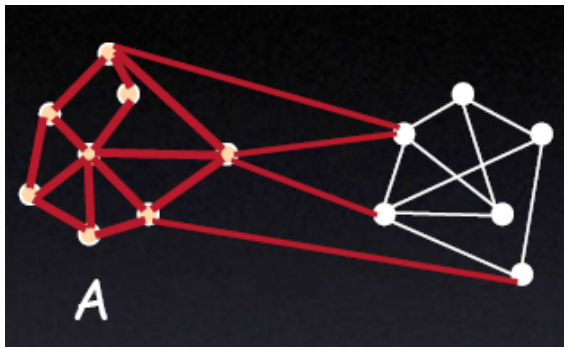
$$d_i = \sum_j w_{i,j}$$



Volume

- Volume của một vùng:

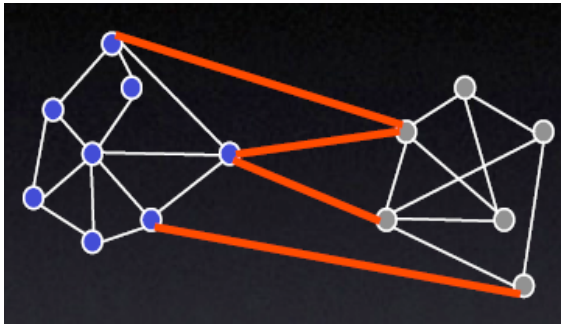
$$vol(A) = \sum_{i \in A} d_i, A \subseteq V$$



Trọng số của một cách phân đoạn

- Trọng số của một cách phân đoạn

$$w(A,B) = \sum_{i \in A, j \in B} w_{i,j}$$

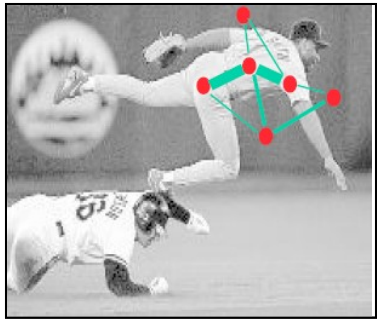


Biểu diễn cụ thể

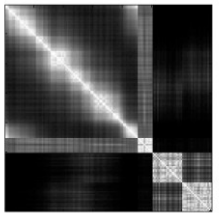
Partition matrix X:

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

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



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: $L = D - W$

Hàm đánh giá trọng số

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

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

Normalized cut

- Tính *W* của một ảnh cho trước
- Tính *D* dựa trên công thức $D(i, i) = \Sigma_j W(i, j)$
- Khi đó normalized-cut được viết lại như sau

$$\frac{y^T(D-W)y}{y^T Dy}$$

J. Shi and J. Malik. Normalized cuts and image segmentation. PAMI 2000

Normalized cut

- Chúng ta có thể giải bài toán trên bằng cách tính y sao cho $(D - W)y = \lambda Dy$
- Lời giải y chính là bài toán tìm eigenvector sao cho đẳng thức trên đạt giá trị nhỏ nhất với y không phải là nghiệm hiển nhiên

Normalized cut algorithm

1. Represent the image as a weighted graph $G = (V,E)$, compute the weight of each edge, and summarize the information in D and W
2. Solve $(D - W)y = \lambda Dy$ for the eigenvector with the second smallest eigenvalue
3. Use the entries of the eigenvector to bipartition the graph

To find more than two clusters:

- Recursively bipartition the graph
- Run k-means clustering on values of several eigenvectors

Ví dụ

Normalized Cuts