

Detección de bordes de imágenes mediante técnicas de Álgebra Lineal Numérica

Dr. Humberto Madrid de la Vega
Lic. Mario Alberto Barrientos Navarro

Centro de Investigación en Matemáticas Aplicadas
UAdeC, México

Seminario Semanal del Laboratorio de Cómputo Científico
Facultad de Ciencias. UNAM. Sept 30 de 2010

Detección de bordes



Figura: Un ejemplo de detección de bordes.

Imagen \rightarrow Grafo \rightarrow Matrices \rightarrow Valores y Vectores propios

Gráfica de una imagen

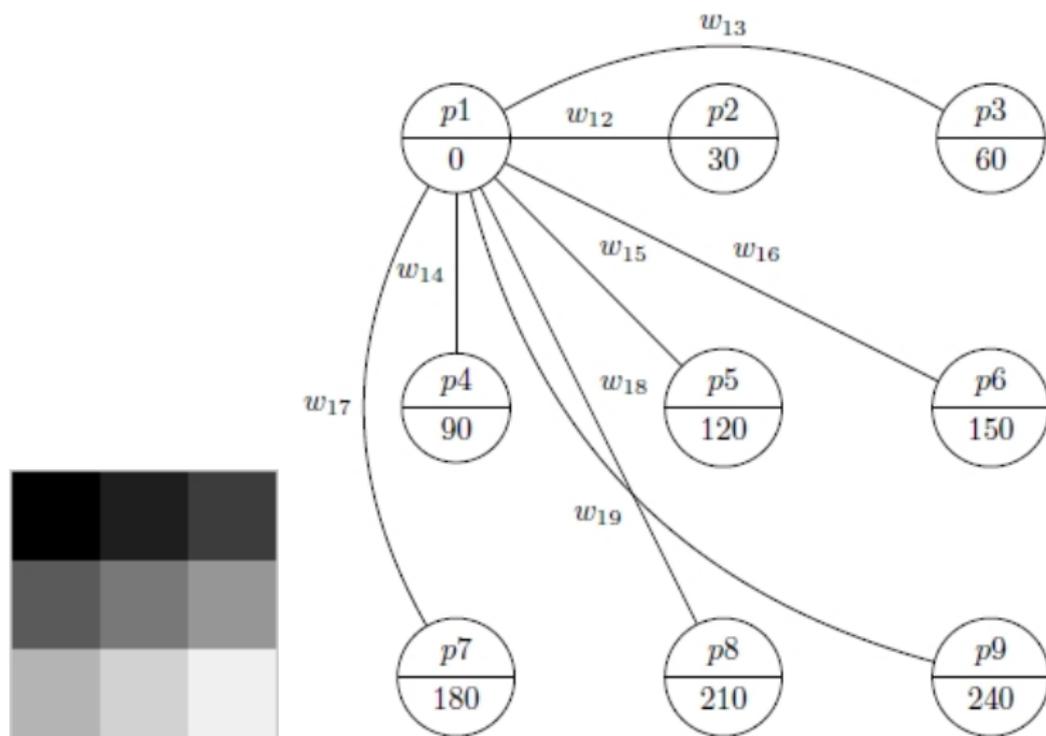


Figura: Contrucción de una gráfica a partir de una imagen.

Corte Normalizado

El corte normalizado (Shi-Malik, 2000) está dado por

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

mín $\text{Ncut}(A, B)$ puede obtenerse como:

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

- ▶ $\mathbf{W}(i, j) = w_{ij}$, $\mathbf{D}_i = \sum_j \mathbf{W}(i, j)$
- ▶ $\mathbf{L} = \mathbf{D}^{-\frac{1}{2}}(\mathbf{D} - \mathbf{W})\mathbf{D}^{-\frac{1}{2}}$ (Laplaciano) es simétrica semidefinida positiva ($\lambda_i \geq 0$).
- ▶ $\mathbf{z}_0 = \mathbf{D}^{\frac{1}{2}}\mathbf{1}$ es un vector propio de (1) con valor propio asociado cero, llamado vector propio más pequeño.
- ▶ El segundo vector propio más pequeño de \mathbf{L} es la solución con valores reales del problema de corte normalizado.

Función de pesos

$$w_{ij} = e^{\frac{-\|F(i)-F(j)\|^2}{\sigma_I}} * \begin{cases} e^{\frac{-\|X(i)-X(j)\|^2}{\sigma_X}}, & \text{si } \|X(i) - X(j)\| \leq r \\ 0 & \text{de otra forma} \end{cases}$$

- ▶ F tonos de los pixeles:

$$F = (0, 90, 180, 30, 120, 210, 60, 150, 240).$$

- ▶ X posición de los pixeles:

$$X(1) = (1 \ 1)$$

$$X(2) = (2 \ 1)$$

$$X(3) = (3 \ 1)$$

$$X(4) = (1 \ 2)$$

$$X(5) = (2 \ 2)$$

$$X(6) = (3 \ 2)$$

$$X(7) = (1 \ 3)$$

$$X(8) = (2 \ 3)$$

$$X(9) = (3 \ 3)$$

- ▶ σ_I y σ_X deben calibrarse.

Tamaño de vecindad

r definirá una vecindad alrededor de cada pixel. Sólo se calcularán los pesos de dicha vecindad. r grande significa contar con más información, pero también más trabajo. r pequeño reduce el trabajo, pero podríamos desechar información relevante

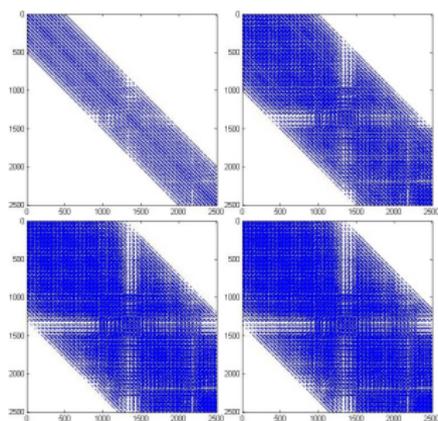


Figura: Matrices de adyacencia \mathbf{W} de una imagen de 50×50 construidas usando $r = 10$, $r = 20$, $r = 26$, y $r = 27$

Dificultades prácticas

- ▶ Si la imagen es de $m \times n$, \mathbf{L} es de $N \times N$, $N = mn$. Imagen de 640×480 requiere de \mathbf{L} de $307,200 \times 307,200$, es decir, 94,371,840,000 elementos.
- ▶ Los métodos para encontrar valores propios que requieren inversas o factorizaciones no funcionan para estas dimensiones.



Figura: Comparación de una imagen de 50×50 y una de 640×480 .

Método de la potencia

- ▶ Matriz \mathbf{A} de $n \times n$ con valores propios $|\lambda_1| > |\lambda_2| \geq |\lambda_3| \geq \dots \geq |\lambda_n|$ (λ_1 valor propio dominante).
- ▶ \mathbf{A} tiene n vectores propios linealmente independientes u_1, u_2, \dots, u_n .
- ▶ Vector \mathbf{x}_0 de dimensión N .

$$\mathbf{x}_1 = \mathbf{A}\mathbf{x}_0$$

$$\mathbf{x}_2 = \mathbf{A}\mathbf{x}_1 = \mathbf{A}^2\mathbf{x}_0$$

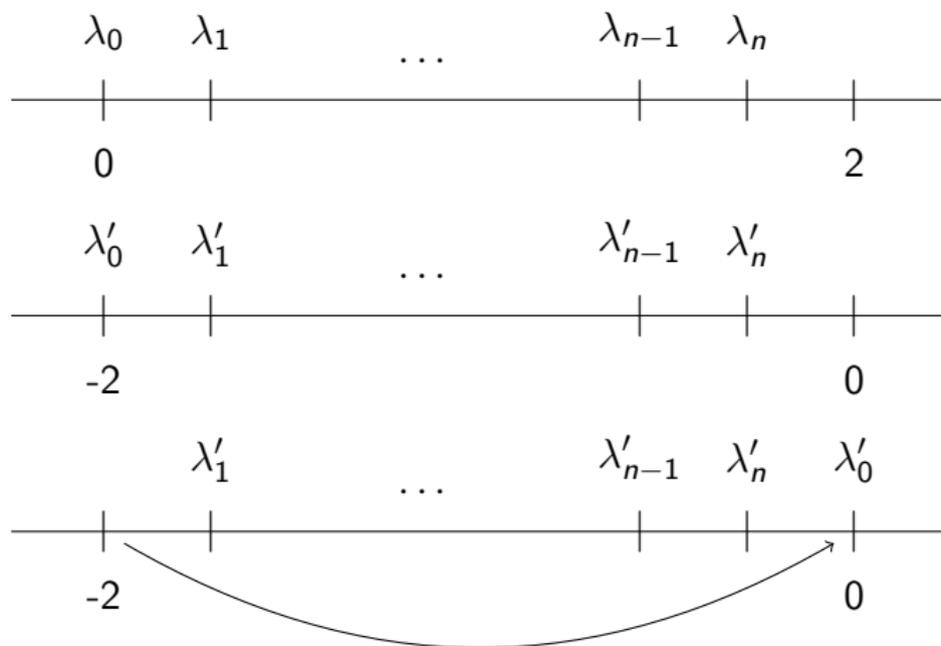
$$\vdots$$

$$\mathbf{x}_k = \mathbf{A}\mathbf{x}_{k-1} = \mathbf{A}^k\mathbf{x}_0$$

- ▶ \mathbf{x}_k es una aproximación al vector propio dominante
- ▶ Criterio de paro: error relativo $e = \|\mathbf{x}_{k-1} - \mathbf{x}_k\| / \|\mathbf{x}_k\|$

Técnicas auxiliares

- ▶ Normalización $\mathbf{x}'_k = \mathbf{x}_k / \|\mathbf{x}_k\|$.
- ▶ Desplazamiento $\mathbf{A}_s = \mathbf{A} - c\mathbf{I}$
- ▶ Deflación $\mathbf{A}_d = (\mathbf{I} - \mathbf{u}\mathbf{u}^T)\mathbf{A}$



Criterios de división

\mathbf{u}_2 , el vector propio subdominante de \mathbf{L} , no satisface la condición de tomar sólo dos valores para poder formar la bipartición. En base a \mathbf{u}_2 formaremos el vector indicador:

$$\mathbf{x}_i = \begin{cases} 1, & \text{si } \mathbf{u}_2(i) > \alpha \\ -1, & \text{si } \mathbf{u}_2(i) \leq \alpha \end{cases}$$

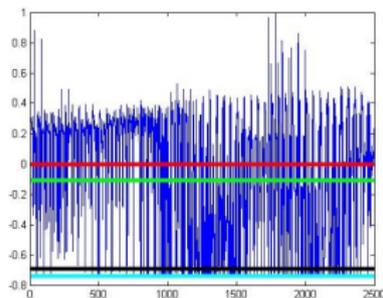


Figura: Entradas de un vector \mathbf{u}_2 . La línea roja indica el umbral $\alpha = 0$.

Tomar α como la media de los elementos de \mathbf{u}_2 o $\alpha = 0$ ha devuelto buenos resultados.

Ejemplos de segmentación

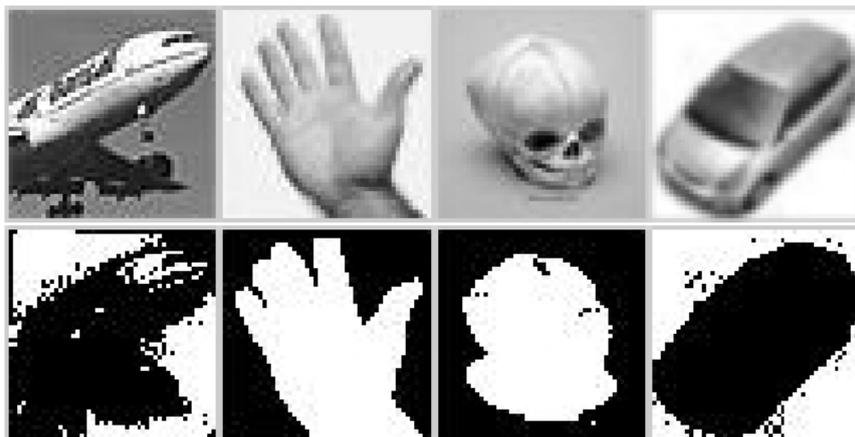


Figura: Algunos ejemplos de segmentación de imágenes de 50×50 , construidos con $r = 5$, $\sigma_I = 1$ y $\sigma_X = 1$, usando $\alpha = 0$ y tolerancia de $1e - 05$.

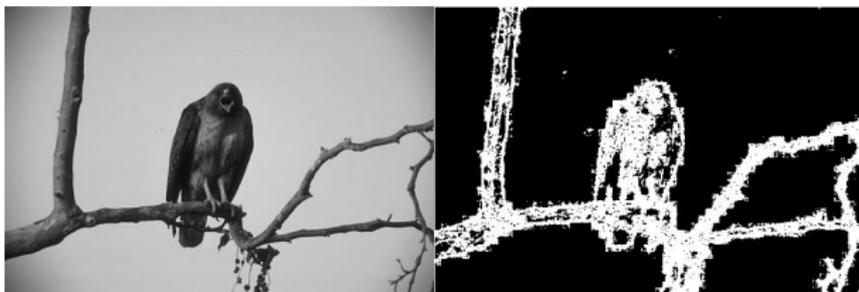
Resultados

El método de la potencia converge lentamente y cada iteración es muy costosa además de que el almacenamiento de L requiere de mucha memoria, provocando que sólo podamos trabajar con imágenes pequeñas y que tome una cantidad de tiempo considerable lograr la segmentación.

Modificaciones para detección de bordes

$$w_{ij} = e^{\frac{-\|F(i)-F(j)\|}{\sigma_I}} * \begin{cases} 1 & \text{si } \|X(i) - X(j)\| \leq 1 \\ 0 & \text{de otra forma} \end{cases} \quad (2)$$

Usar $r = 1$ provoca que la vecindad del pixel con la que se trabaja sea muy pequeña, y esta función de pesos hace que dos pixeles queden en conjuntos distintos sólo si hay una diferencia de tonos drástica. Estas características provocan que la segmentación coincida con los bordes de la imagen. Fijar $r = 1$ crea una estructura pentadiagonal en \mathbf{W} que podemos aprovechar para trabajar con imágenes de mayores dimensiones. Usamos $\sigma_I = 1255$ para este tipo de segmentación.



Filtro mediana

La aplicación del filtro mediana mejora considerablemente el resultado. Ejemplo obtenido tomando una vecindad de 5×5 para el filtro:



Figura: Arriba: imagen original y su segmentación. Abajo: imagen filtrada y la segmentación obtenida.

Resultados

La cantidad de recursos que se necesitan para detección de bordes es mucho menor a los que se requerían para el otro tipo de segmentación, por lo que el desempeño mejora enormemente. Otro aspecto interesante es que el método de la potencia converge en una sola iteración efectiva. Buscando una explicación para esto encontramos otro resultado interesante.

Modelo simplificado

Tomando

- ▶ $\mathbf{x}_0 = \mathbf{1}$
- ▶ $\mathbf{L}^* \approx \frac{4}{5}\mathbf{I} - \frac{1}{5}\mathbf{P}$

Obtendremos \mathbf{x}_1 de la forma

$$\mathbf{x}_1 = -2\mathbf{1} + 2c\bar{\mathbf{u}}_1$$

con $c = \sum_i (\bar{\mathbf{u}}_1)_i$, y se puede mostrar que \mathbf{x}_2 será un múltiplo de \mathbf{x}_1 , por lo que no es necesario seguir iterando. Por tanto $\mathbf{u}_2 = -2\mathbf{1} + 2c\bar{\mathbf{u}}_1$

¿Cuánto afectan estas aproximaciones?

Usando el criterio de división con $\alpha = 0$, la diferencia entre el modelo exacto y el original es casi imperceptible.

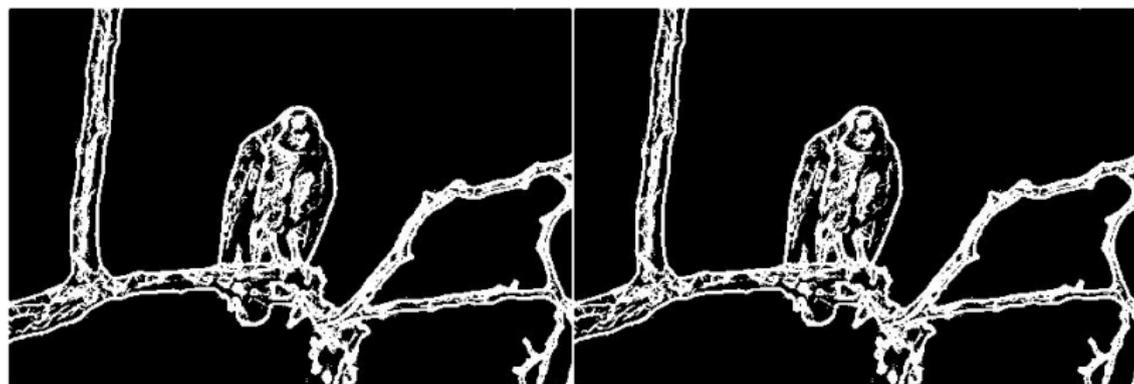


Figura: Segmentación obtenida usando el modelo original (izquierda) y el modelo simplificado (derecha)

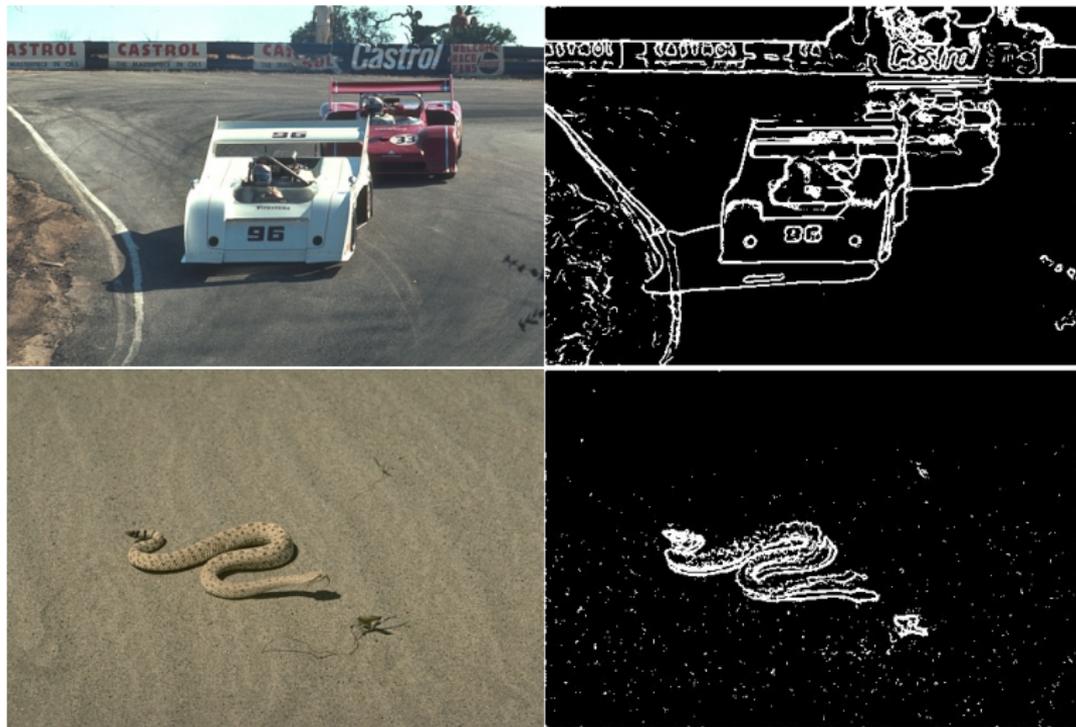
Ventajas del modelo simplificado

- ▶ La segmentación se obtiene de forma cerrada (fórmula).
- ▶ El nuevo algoritmo es de orden lineal $O(N)$
- ▶ La cantidad de elementos a almacenar puede reducirse a un vector de dimension $N = mn$, sin necesidad de almacenamiento sparse.

Ejemplos (1)



Ejemplos (2)



Trabajo futuro

- ▶ Inclusión de técnicas estadísticas para reducir aun más el costo computacional del procedimiento.
- ▶ Aplicación del método a sistemas de detección de movimiento y conteo de automóviles y personas.

Bibliografía



Jianbo Shi y Jitendra Malik.

Normalized Cuts and Image Segmentation.

IEEE Transactions on Pattern Analysis and Machine Intelligence, 2000.



Gene Golub y Charles Van-Loan.

Matrix Computations.

John Hopkins Press, 1996.



Gonzalo R. Arce.

Nonlinear signal processing: a statistical approach.

Wiley, 2005.



D. Martin, C. Fowlkes, D. Tal y J. Malik.

A Database of Human Segmented Natural Images and its Application to Evaluating Segmentation Algorithms and Measuring Ecological Statistics.

Proc. 8th Int'l Conf. Computer Vision, 2001.

<http://www.eecs.berkeley.edu/Research/Projects/CS/vision/bsds/>