

Redes de Flujo: Segmentación de imágenes

Teoría de Algoritmos I (75.29 / 95.06)

Ing. Víctor Daniel Podberezski

✉ vpodberezski@fi.uba.ar

Enunciado

Contamos

con una imagen (formada por un conjunto de pixels)

Deseamos

Separar el fondo de la imagen de lo que está en primer plano



Definiciones

Llamaremos V al conjunto de pixels

que forman la imagen

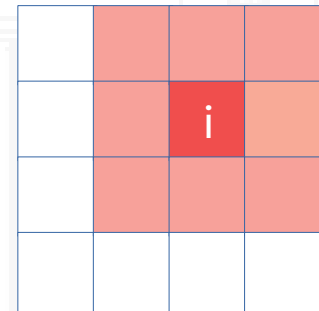
Forma una cuadrícula

Cada pixel i

Tiene un conjunto de pixels vecinos

8 para los pixels internos

(3 o 5 para los externos)



Representación utilizando un Grafo

Podemos verlo como un Grafo $G=(V,E)$

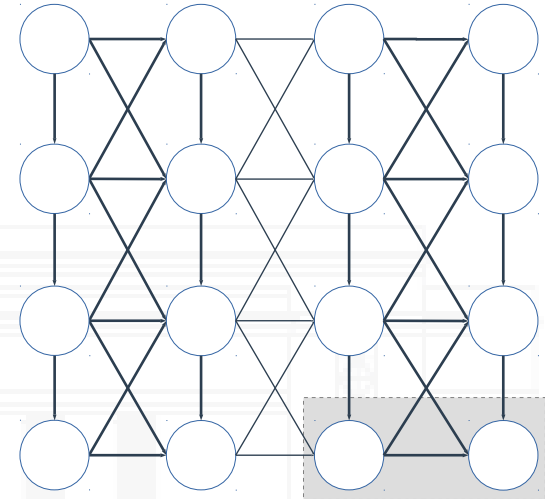
V los pixels

E los ejes que establecen la vecindad de cada pixel

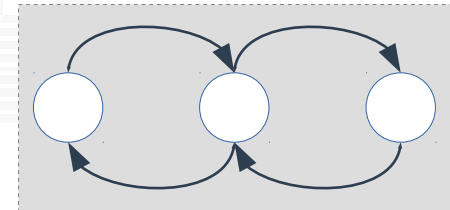
Cada relación de vecindad

Es bidireccional

Lo podemos pensar como 2 ejes direccionados



Los ejes de vecindad no se muestran duplicados por simplicidad



Probabilidad de pertenencia

Para cada pixel i tenemos

Valor “deseo” a_i de que pertenezca al primer plano

Valor “deseo” b_i de que pertenezca al fondo

(ambos valores positivos y disponibles para la segmentación)

Si tomamos un pixel como aislado

$a_i > b_i \rightarrow$ pertenece al primer plano

Sino pertenece al fondo

... pero queremos que la decisión dependa también de sus vecinos

Penalidad por cambio

Si muchos de los vecinos del pixel i

Pertenecen al fondo, es más “deseable” que i también

Por cada par de pixels (i,j) vecinos

Existe una penalidad p_{ij} de que pertenezcan a diferente segmento (fondo / 1er plano)

El valor de p_{ij} es mayor a cero

Un problema de maximización

Si llamamos

A al conjunto de pixels de la imagen que pertenecen al primer plano

B al conjunto de pixels de la imagen que pertenecen al fondo

Podemos calcular

La “deseabilidad” de la segmentación A/B como:

$$q(A, B) = \sum_{i \in A} a_i + \sum_{j \in B} b_j - \sum_{\substack{i \in A \\ j \in B \\ i, j \text{ vecinos}}} p_{ij}$$

Queremos

Seleccionar la elección del segmento A y B que maximice $q(A, B)$

Transformación en minimización

Si llamamos

$$Q = \sum_{i \in E} a_i + b_i$$

Observamos que

$$\sum_{i \in A} a_i + \sum_{j \in B} b_j = Q - \sum_{i \in A} b_i - \sum_{j \in B} a_j$$

Por lo tanto

$$q(A, B) = Q - \sum_{i \in A} b_i - \sum_{j \in B} a_j - \sum_{\substack{i \in A \\ j \in B \\ i, j \text{ vecinos}}} p_{ij}$$

Transformación en minimización (cont.)

Podemos transformar el problema de maximización

$$q(A, B) = Q - \sum_{i \in A} b_i - \sum_{j \in B} a_j - \sum_{\substack{i \in A \\ j \in B \\ i, j \text{ vecinos}}} p_{ij}$$

A un problema de minimización de

$$q'(A, B) = \sum_{i \in A} b_i + \sum_{j \in B} a_j + \sum_{\substack{i \in A \\ j \in B \\ i, j \text{ vecinos}}} p_{ij}$$

Transformación en un problema de corte mínimo

Utilizaremos

un nodo fuente “s” como representación del segmento fondo

Un nodo sumidero “t” como representación del segmento primer plano

Se creará

Un eje s-i por cada nodo “pixel” con capacidad a_i

Un eje i-t por cada nodo “pixel” con capacidad b_i

Un eje i-j por cada pixel vecino con capacidad p_{ij} (esto genera 1 eje ida y otro vuelta entre cada par de pixeles vecinos)

Red de flujos

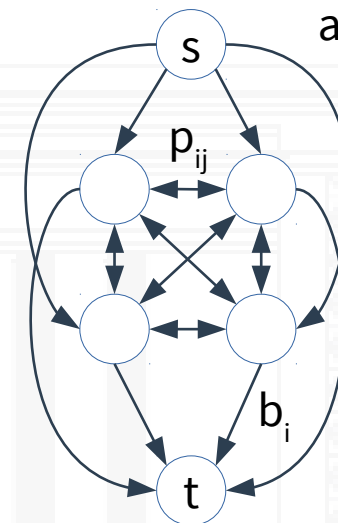
Con los cambios propuestos

Tenemos un $G' = (V', E')$ red de flujo

Con

$V' = \text{pixels} + \{s, t\}$

$E' = \text{ejes vecindad} + \text{ejes } s-i + \text{ejes } i-t$



Modelo simplificado
de 4 píxeles de red de
flujos

Corte s-t

Si analizamos

un coste A-B generico, con $s \in A$ y $t \in B$

Llamaremos $c(A,B)$

A la capacidad del corte A-B

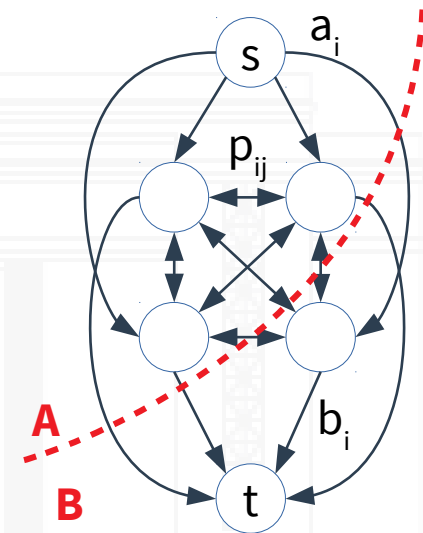
(suma de las capacidades de los ejes que van de nodos de A a nodos de B)

Los ejes de $c(A,B)$ corresponden a

Ejes s-j, con $j \in B$ (con capacidad a_j)

Ejes i-t con $i \in A$ (con capacidad b_i)

Ejes i,j con $i \in A$ y $j \in B$ (con capacidad p_{ij}) ← los ejes de vecindad



Corte s-t (cont.)

Sumando las capacidades tenemos

Que el resultado equivale a la formula de $q'(A,B)$

$$\sum_{i \in A} b_i + \sum_{j \in B} a_j + \sum_{\substack{i \in A \\ j \in B \\ i, j \text{ vecinos}}} p_{ij} = q'(A, B)$$

Podemos hacer cortes arbitrariamente

Algunos tendrán capacidad mayor y otros menor

Pero nos interesa el menor de los posibles

Corte mínimo s-t

El cálculo del corte mínimo

Esta vinculado al cálculo del flujo máximo

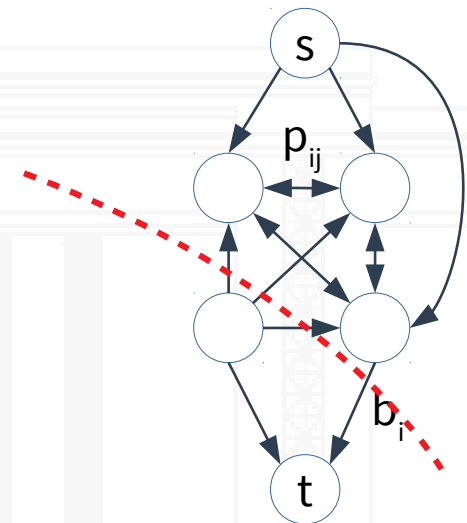
Se resuelve el problema del flujo máximo

Al terminar la ejecución no existen flujos de aumento

Quedan separados nodos alcanzables desde s y no alcanzables (que se unen a t)

La frontera serán ejes con capacidad cero de A a B

Estos ejes conforman el corte mínimo



Pasos segmentación de imagen

1. Dada la imagen y los parámetros p_{ij} , a_i y b_i para todo pixel

Construir la red de flujo según método

2. Resolver problema de flujo máximo

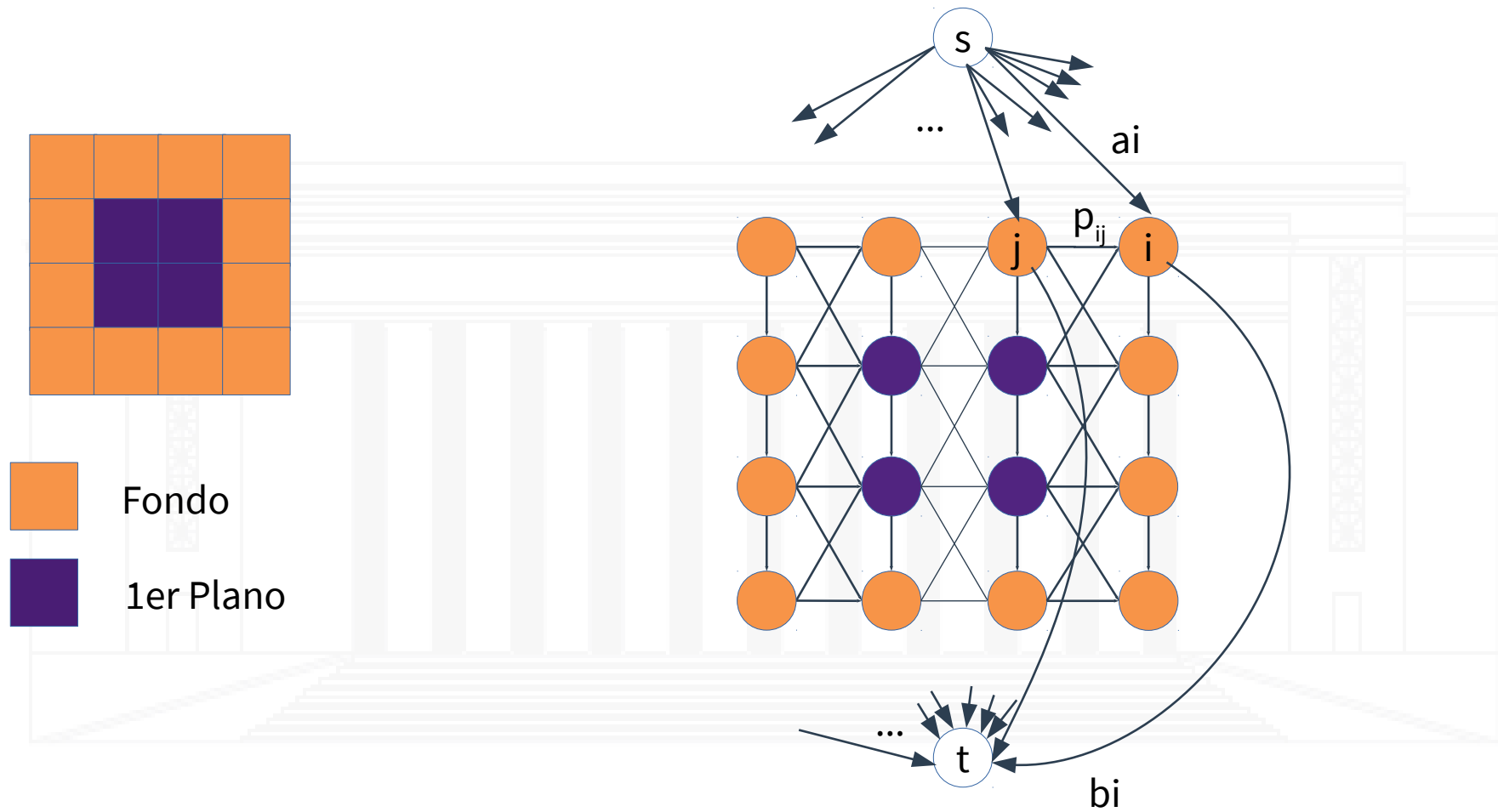
Mediante Ford-Fulkerson

3. Desde el grafo residual final, realizar BFS desde S.

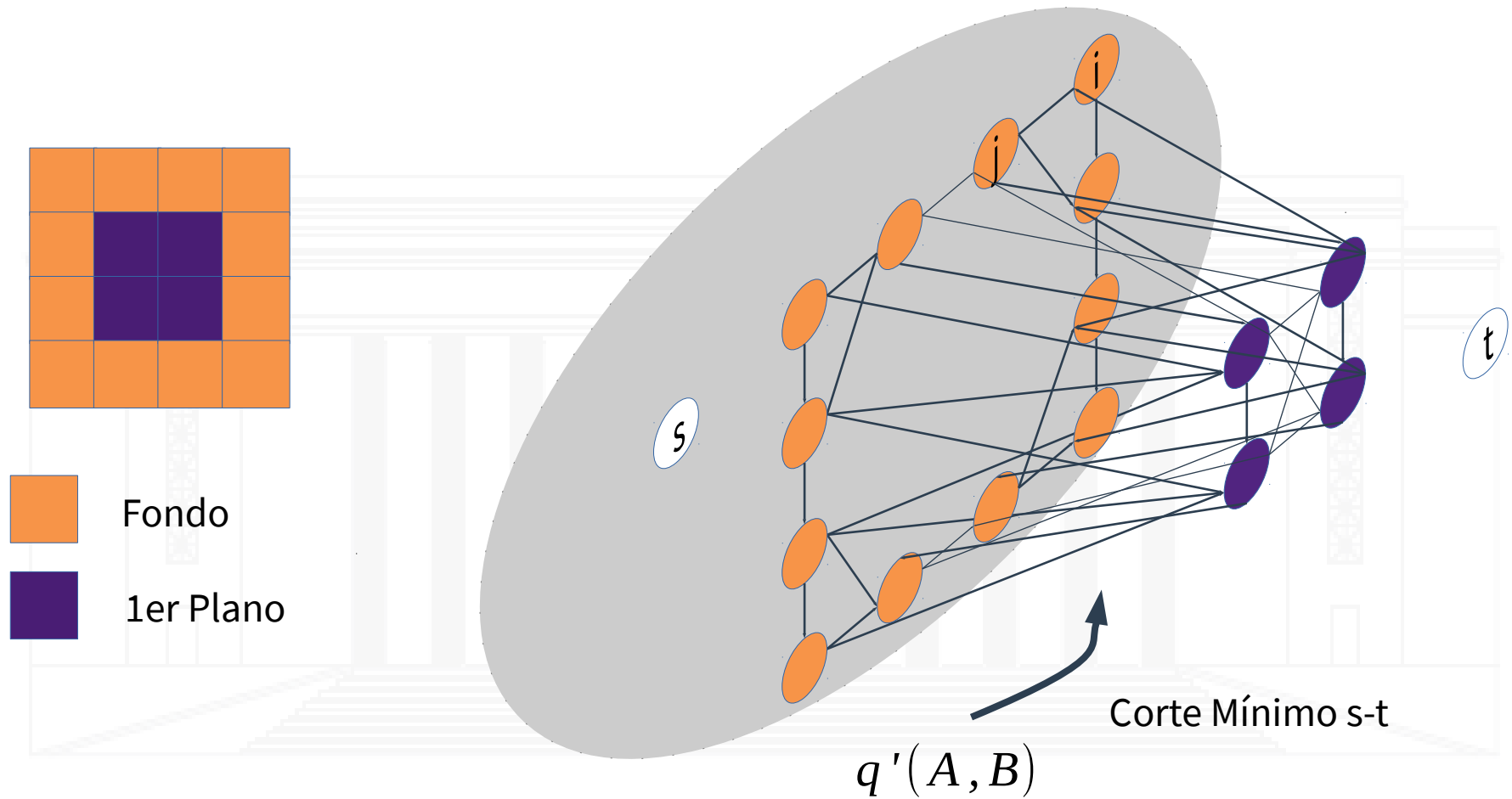
Todos los nodos alcanzables serán los pixels del segmento “fondo”

El resto pertenece a los pixels del segmento “primer plano”

Visualización de la solución



Visualización de la solución





Presentación realizada en Mayo de 2020