

# Algoritmos BMA

Adolfo Otero

José Otero

José Antonio Corrales

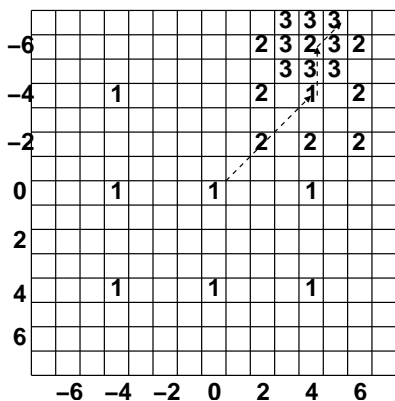
Departamento de Informática.

Área de Lenguajes y Sistemas Informáticos

Universidad de Oviedo

**Búsqueda en tres pasos:** refinamiento sucesivo de la búsqueda.

- Tamaño de paso decreciente logarítmicamente (se divide por 2).
- Tamaño inicial entero más próximo mayor que  $\frac{d}{2}$ .
- En cada paso se chequean 9 posiciones.
- Se continúa por la posición con la mínima BDM (Block Distortion Measure).
- Finaliza cuando el tamaño es 1.

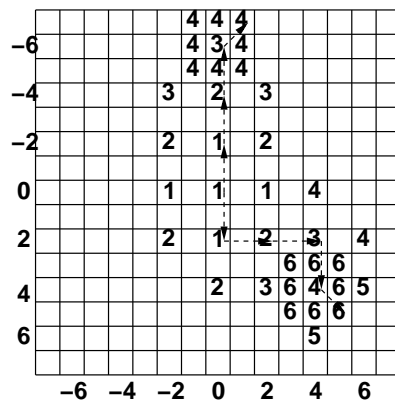


## Medidas de distorsión

- Media del valor absoluto de la diferencia  
 $\frac{1}{mn} \sum_{p=1}^m \sum_{q=1}^n |A[p, q] - B[p, q]|$
- Error cuadrático medio  
 $\frac{1}{mn} \sum_{p=1}^m \sum_{q=1}^n (A[p, q] - B[p, q])^2$
- Diferencia de clasificación de Pel  
Los pixels se clasifican como emparejados o no en función de la diferencia en intensidad. Se contabilizan los emparejamientos.  
 $\sum_{p=1}^m \sum_{q=1}^n \text{ord}(|A[p, q] - B[p, q]| \leq t)$
- Proyección integral  
 $\sum_{p=1}^m | \sum_{q=1}^n A[p, q] - \sum_{q=1}^n B[p, q] | + \sum_{q=1}^n | \sum_{p=1}^m A[p, q] - \sum_{p=1}^m B[p, q] |$

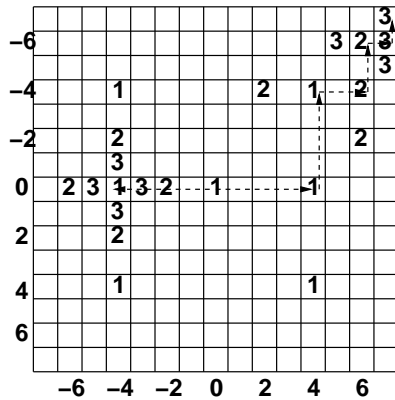
## Búsqueda logarítmica bidimensional

- Patrón de búsqueda en cruz '+'
- Tamaño del paso inicial entero más próximo mayor que  $\frac{d}{4}$
- El tamaño se reduce a la mitad si:
  - Posición de mínima BDM es la central.
  - Límite del espacio de búsqueda.
- Cuando se alcanza el tamaño mínimo se busca en las 8 posiciones que rodean a la central.



### Búsqueda ortogonal

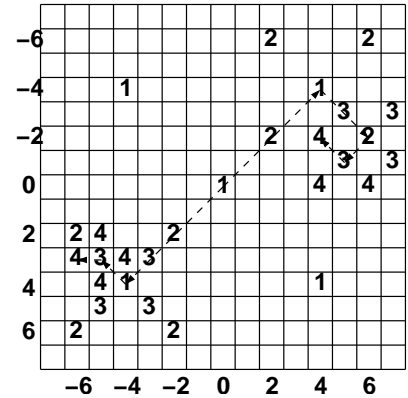
- Búsqueda horizontal/vertical.
- Tamaño de paso decreciente logarítmicamente.
- Tamaño inicial entero más próximo menor que  $\frac{d}{2}$ .



### Búsqueda en cruz

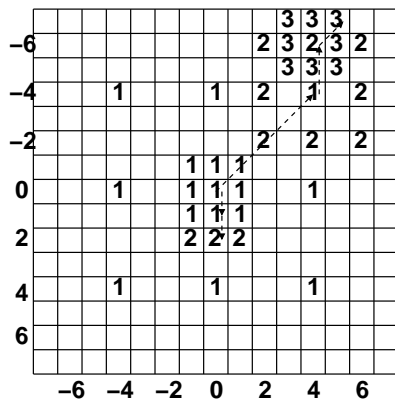
- Patrón en cruz 'x'.
- Tamaño de paso decreciente.
- Tamaño inicial entero  $\frac{d}{2}$ .
- Si paso mínimo y posición de mínima BDM:
  - Centro
  - Esquina sup izq
  - Esquina inf dch

Se cambia el patrón de búsqueda a una '+'



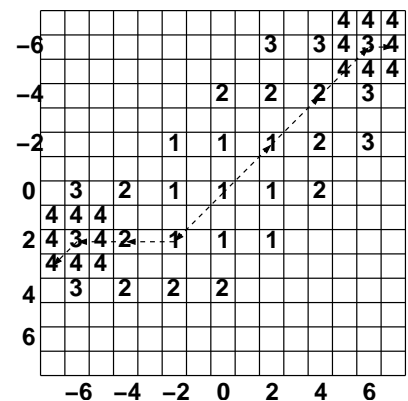
### Búsqueda en tres pasos para pequeños desplazamientos

- Búsqueda inicial en 8 puntos a  $\frac{d}{2}$ , en el centro y en los 8 adyacentes.
- Si la BDM se produce en los 8 adyacentes al centro, nueva búsqueda 3x3 alrededor.
- Si se produce en los 8 más alejados, búsqueda en tres pasos normal.



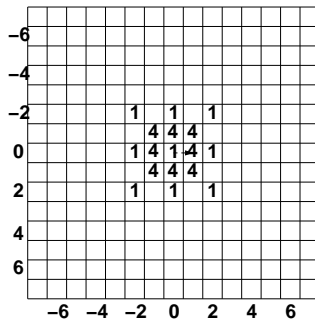
### Búsqueda en cuatro pasos

- Tamaño inicial de paso pequeño, entero más próximo mayor que  $\frac{d}{2}$ .
- Puede parar en el paso 2 o 3 si la BDM mínima se da en el centro.

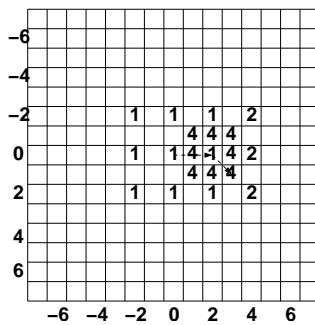


## Búsqueda en cuatro pasos II

- Parada en el segundo paso.

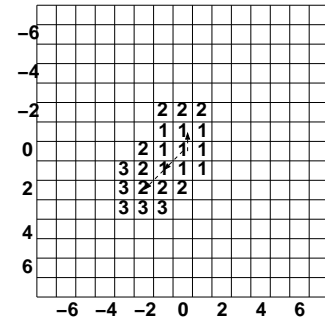


- Parada en el tercer paso.



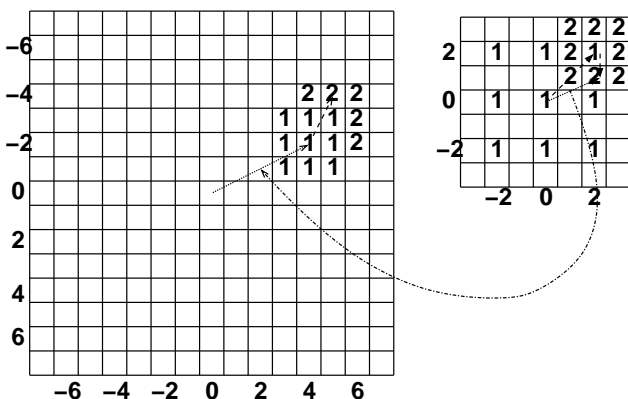
## Búsqueda basada en descenso del gradiente

- Tamaño del paso = 1.
- No hay restricción al número de pasos.
- Finaliza si la mínima BDM se produce en el centro.
- En caso contrario siguiente búsqueda centrada en el punto con BDM mínima.



## Búsqueda jerárquica

- Búsqueda inicial a baja resolución.
- Extrapolación del desplazamiento a alta resolución.
- Nueva búsqueda alrededor de la estimación a baja resolución.
- Se pueden utilizar distintos métodos en cada caso.



## Búsqueda voraz

- Intentan reducir el número de evaluaciones del criterio que define BDM.
- En lugar de desplazarse al punto con BDM mínima se desplazan al primero que presenta una BDM menor que el anterior.
- El tamaño del paso se reduce cuando se encuentra un mínimo, es decir, no se encuentra ninguno menor.
- Terminan cuando el tamaño de paso es 1.

## Algoritmo BMA voraz A

- Tamaño de paso inicial  $s =$  entero más próximo mayor que  $\frac{d}{2}$ .
- Actualización:  $s = \frac{s}{2}$
- En primer lugar se mide BDM en  $(0,0)$ .
- A continuación  $(+s, 0)$ , si BDM no es menor  $(0, +s)$ ,  $(-s, 0)$ ,  $(0, -s)$  (sentido antihorario).

### Algoritmo BMA voraz B

- Similar al anterior con tamaño inicial  $s =$  entero más próximo mayor que  $\frac{d}{4}$ .
- Actualización:  $s = \frac{s}{2}$

### Algoritmo BMA voraz C

- Similar al anterior con tamaño inicial  $s =$  entero más próximo mayor que  $\frac{d}{4}$ .
- Actualización:  $s = \frac{s}{4}$

### Inconvenientes de los algoritmos A, B y C

- La búsqueda continúa siempre a la derecha del punto anterior, independientemente del resultado obtenido en el paso anterior.
- Sería más eficiente buscar primero en la dirección del desplazamiento obtenido en el paso anterior.

### Algoritmo BMA voraz F

- El orden en el que se realizan las búsquedas tiene impacto sobre la eficiencia.
- Optimizado para escenas con movimiento fundamentalmente en horizontal.
- Búsqueda a la derecha y a la izquierda del punto central.
- Tamaño inicial  $s =$  entero más próximo mayor que  $\frac{d}{4}$ .
- Actualización:  $s = \frac{s}{4}$

### Algoritmo BMA voraz Adaptativo

- Decide el orden de la búsqueda en función del movimiento obtenido en otras áreas de la imagen o en otros frames.

### Algoritmo BMA voraz D

- Tamaño inicial  $s =$  entero más próximo mayor que  $\frac{d}{4}$ .
- Se examina en primer lugar  $(0, 0)$ , después  $(+s, 0)$ .
- Si  $(+s, 0)$  posee una menor BDM que  $(0, 0)$ , se sigue hacia  $(+2s, 0)$ .
- Se mantiene la dirección seguida en el paso anterior.
- Si la BDM obtenida es mayor que la del paso anterior, se pasa a la siguiente dirección en sentido antihorario.
- Si el centro proporciona una BDM mínima, se divide el tamaño del paso entre 4.

### Algoritmo BMA voraz E

- Similar al anterior.
- Tamaño inicial  $s =$  entero más próximo mayor que  $\frac{d}{2}$ .
- Actualización:  $s = \frac{s}{2}$

### Otros algoritmos BMA

#### Espacio de búsqueda adaptativo

- Intenta prevenir el estancamiento en mínimos locales.
- El tamaño del espacio de búsqueda depende de la diferencia entre la BDM mínima y la anterior.
- A mayor diferencia, menor espacio de búsqueda.

#### Algoritmos dependientes

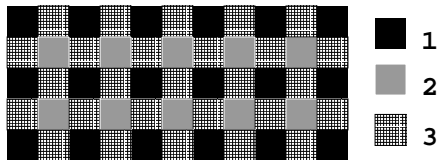
- El movimiento detectado en la escena se debe:
  - Al movimiento de los objetos.
  - Al movimiento de la cámara.
- Entre frames el movimiento puede no cambiar demasiado.
- Dentro de los frames, pixels próximos se mueven de forma similar.
- Idea: utilizar el movimiento medido en otros pixels/otros frames.

## Dependencia espacial

- Utiliza el movimiento de pixels vecinos para predecir el punto en donde se producirá la BDM mínima.
  - Media ponderada (a la distancia).
  - Media de los pixels con movimiento más parecido.
- Cada pixel posee 8 vecinos, pero en función del orden del cálculo, pueden no estar disponibles los resultados para algunos.
- Si se utilizan datos de pixels a un lado del actual, puede haber problemas en los contornos de los objetos.



Algoritmo de tres pasadas:



## Espacios de búsqueda multidimensionales

- Los algoritmos precedentes no tienen en cuenta:
  - Cambios en la iluminación.
  - Rotación / movimientos en direcciones no paralelas al plano de la imagen.
- Soluciones:
  - Utilizar primitivas distintas a la intensidad de gris.
  - Añadir una dimensión más a la búsqueda: factor de iluminación, escalado, rotación, deformación.

## Iluminación

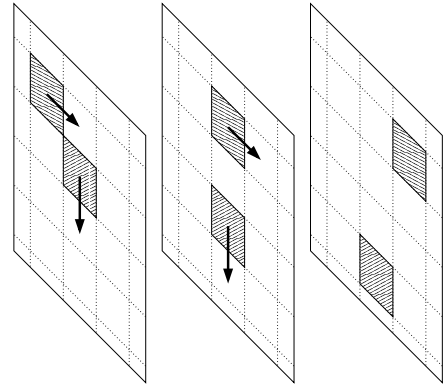
Se busca la posición de mínima BDM y además mínima distancia en iluminación.



## Escalado, Rotación, Distorsión

## Dependencia temporal

- Introduce la memoria en el algoritmo.
- Supone que los pixels mantienen (aproximadamente) su velocidad en el tiempo.
- Se utiliza esta hipótesis para buscar alrededor de la posición que tendría el punto si se mantuviese la velocidad.



Bibliografía de estas transparencias:

Colin E. Manning. Master Thesis.

<http://newmediarepublic.com/dvideo/>

Keng Cheung. PhD Thesis.

<http://www.image.cityu.hk/ckcheung/thesis>

Referencias de los algoritmos:

<http://newmediarepublic.com/dvideo/compression/references.html>