



---

## Práctica 2: Programación Dinámica

---

### 1. CONSIDERACIONES GENERALES

- La entrega de la práctica se realizará en moodle, enviando un fichero `practica1.zip` en la tarea habilitada para ello.
- Se entregará un fichero `practica2.zip` que contiene un directorio denominado `practica2_NIA1_NIA2` (siendo NIA1 y NIA2 los números identificadores de cada estudiante asignados por la Universidad de Zaragoza, y NIA1 será el NIA menor. En el caso de un grupo de práctica formado por un único alumno, el directorio tendrá como nombre `practica2_NIA` (con el identificador de ese alumno).
- El directorio incluirá los siguientes ficheros de texto:
  - Descripción general del directorio: cómo está organizado, instrucciones de instalación, compilación y ejecución, instrucciones para repetir las pruebas, etc. (tiene que llamarse **LEEME**).
  - Listados del código debidamente comentados. Deberán seguir una estructura lógica para poder encontrar y navegar adecuadamente cada una de las partes de la práctica.
  - Un programa para la *shell* denominado `ejecutar.sh` que automatice la compilación y ejecución de los programas entregados con los casos de prueba.
  - Los ficheros auxiliares de entrada necesarios para ejecutar las pruebas del punto anterior.
- El directorio incluirá también un informe con la presentación y análisis de resultados (fichero PDF, máximo 3 páginas sin portada). Indicar: nombre, apellidos y NIA de cada miembro del grupo de práctica.
- **Fechas límite de entrega para la primera convocatoria:**

Grupo	Fecha y hora
Jueves A	27/02/2025 8:00AM
Viernes A	28/02/2025 8:00AM
Jueves B	06/03/2025 8:00AM
Viernes B	07/03/2025 8:00AM

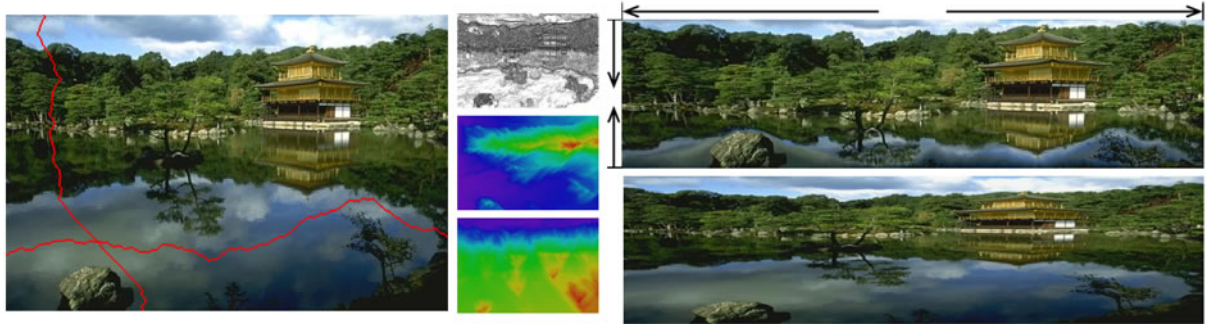
## 1.1. EVALUACIÓN

- En la calificación se tendrán en cuenta los siguientes aspectos: documentación, diseño e implementación, diseño de casos de prueba, análisis de las pruebas realizadas y facilidad para la repetición de las pruebas por los profesores.
- Se aplicarán las reglas de tratamiento de casos de plagio explicadas en la presentación de la asignatura.

La valoración máxima de la práctica es de **10 puntos**.

## 2. ENUNCIADO

En [AS07]<sup>1</sup> se describió un método que podemos traducir como reducción de costuras (*seam carving*) que utiliza programación dinámica para reducir la anchura o la altura de una imagen, teniendo en cuenta su contenido. En lugar de hacer un escalado o eliminar píxeles en los bordes, la reducción de costuras se basa en eliminar de la imagen los píxeles cuya ausencia será menos llamativa. Este método permite mantener el contenido más importante reduciendo el tamaño de la imagen.



Para ello, el primer paso que se tiene que realizar es calcular una función de energía para la imagen. Existen muchas formas de hacer esto, pero podemos utilizar la siguiente simplificación.

Un píxel está rodeado de otros 8 en el caso general:

A	B	C
D	E	F
G	H	I

Podemos utilizar la fórmula para el píxel  $E$ :

$$energia(E) = \sqrt{energia_x^2 + energia_y^2}$$

donde

$$energia_x = a + 2d + g - c - 2f - i$$

$$energia_y = a + 2b + c - g - 2h - i$$

Cada letra minúscula representa el brillo (suma de la componente roja, azul y verde) del píxel.

Para los píxeles que están en los bordes de la imagen podemos suponer que la imagen está rodeada de un marco de píxeles de color negro (brillo 0).

<sup>1</sup><https://doi.org/10.1145/1276377.1276390>

### 3. COSTURAS

Reduciremos la anchura de la imagen eliminando costuras verticales. Una costura vertical es un conjunto de píxeles conectados verticalmente, uno de cada fila de la imagen. Cualquier par de píxeles de dos filas consecutivas se considera que están conectados cuando están en la misma columna o en la misma diagonal. La energía de una costura es la suma de las energías de los píxeles que la forman.

La reducción de costuras consiste en encontrar la costura vertical de menor energía y eliminarla de la imagen. Cada vez que se elimina una de estas costuras la imagen cambia y, por lo tanto, será necesario recalcular las energías si se desea reiterar el proceso.

Algunas alternativas más sencillas podrían ser (aproximaciones voraces):

- Eliminar directamente el píxel de menor energía de cada fila.
- Calcular las energías de cada columna -en lugar de un camino- de la imagen y se elimina la que tenga el menor valor.

Para pensar: ¿Qué sucedería en estos casos? (es fácil probarlo)

Con la aproximación propuesta, las regiones con cambios bruscos como bordes y zonas con más detalle de la imagen, tendrán energías más altas.

Para calcular una costura óptima o de mínima energía se puede utilizar un algoritmo de programación dinámica. Este enfoque calcula el coste para llegar a un píxel dado en la fila  $i$  sumando su energía al coste mínimo necesario para llegar a sus vecinos en la fila  $i - 1$ . Estos valores se utilizarán para eliminar la costura de menor energía de la imagen.

La ecuación en recurrencias (de manera esquemática) es, si  $c(i, j)$  es la energía de la costura mínima que llega al píxel  $(i, j)$ :

$$\begin{aligned} c(i, j) &= \min\{c(i-1, j-1), c(i-1, j), c(i-1, j+1)\} + e(i, j) \\ c(i, 1) &= e(i, 1) \end{aligned}$$

Esto es, el coste mínimo para llegar hasta el píxel  $(i, j)$  se calcula como el mínimo entre los costes de la fila anterior,  $i - 1$ , en la columna anterior  $j - 1$ , la misma columna,  $j$ , y la columna siguiente  $j + 1$  al que se suma la energía del píxel.

El coste mínimo para llegar desde el píxel  $(i, 1)$  en la primera fila de la imagen es la energía del propio píxel (hay una única costura con un píxel de longitud).

**TAREA 1. DISEÑO** Se pide diseñar un algoritmo de **programación dinámica** que, a partir de una imagen reduzca su anchura en un número de píxeles repitiendo el proceso de reducción de costuras tantas veces como se le indique. Idealmente, deberíamos poder resolver la ecuación en recurrencias directamente y después tratar de calcular los valores de manera más eficiente.

**TAREA 2. IMPLEMENTACIÓN** Se pide desarrollar un programa que implemente el algoritmo de solución propuesto. La forma de ejecutar el programa será la siguiente:

```
lab000:> costuras 5 archivo resultados
```

donde **costuras** es el programa ejecutable, **5** es un entero que indica el número de columnas en que se reducirá la imagen, **archivo** es una imagen en el formato que elijamos para realizar el procesamiento y **resultados** es un directorio donde se almacenarán los resultados correspondientes (imágenes intermedias y final; preferiblemente se utilizará para las pruebas el directorio `/tmp` o alguno similar que no aumente el espacio de almacenamiento del usuario).

Para la manipulación de las imágenes, cálculo de los valores relativos a los píxeles y otras operaciones relacionadas se recomienda buscar una biblioteca disponible en el lenguaje de programación que elijamos, no es necesario implementar esa parte.

TAREA 3. EXPERIMENTACIÓN Analizar la corrección y eficiencia (coste y tiempo de ejecución) del algoritmo implementado a través de un conjunto de pruebas. Probar con diversas imágenes para ver el efecto de la reducción en las mismas.

### 3.1. BIBLIOGRAFÍA → EN MOODLE

1. Transparencias de la asignatura.

## REFERENCIAS

[AS07] Shai Avidan and Ariel Shamir. Seam carving for content-aware image resizing. *ACM Trans. Graph.*, 26(3), July 2007.

### ALGUNOS COMENTARIOS ADICIONALES

- El proceso funciona bien con imágenes que tienen una parte del contenido con energía alta y otras partes donde hay energía baja, que son las que ese eliminarán. Si la imagen contiene zonas de baja energía que son importantes, o toda la imagen tiene mucha energía, el resultado no será muy bueno.
- Es posible probar diferentes funciones de cálculo de energía, que proporcionarán diferentes resultados a la hora de reducir las imágenes.
- La reducción del tamaño de las imágenes se puede hacer, de forma completamente análoga, en altura en lugar de en anchura.
- El mismo proceso se puede utilizar para agrandar las imágenes. En este caso calcularemos las  $k$  costuras de menor energía y añadiendo píxeles en esos lugares con la media de los valores de sus vecinos. Este método funciona bien para valores razonablemente pequeños de  $k$  y depende del contenido de la imagen, ya que los píxeles añadidos no contendrán las texturas y variaciones naturales de una imagen normal.
- También es posible utilizar estas ideas para eliminar objetos de la imagen: bastaría con cambiar la energía de la zona que queremos eliminar de forma que en el proceso de eliminación de costuras sus componentes sean las elegidas para la supresión.