

# Informe Práctica 2: Programación Dinámica

Jorge Soria Romeo  
872016

Alberto Francisco Solaz García  
873373

February 2025

## 1. Introducción

En este informe analizamos la eficiencia y efectividad del algoritmo de reducción de costuras que hemos implementado en C++ utilizando programación dinámica y memoización. Su propósito es reducir la anchura o altura de una imagen minimizando la pérdida de información.

## 2. Eficiencia

El funcionamiento de nuestro algoritmo se describe mediante el flow diagram de la figura 1. A continuación describimos su coste asintótico en notación big-O en función del ancho de la imagen  $W$ , la altura de la imagen  $H$  y el número de costuras a eliminar  $N$ :

1. **Analizar argumentos de la línea de comandos**  $\mathcal{O}(1)$
2. **Cargar imagen usando CImg**  $\mathcal{O}(W \times H)$
3. **Convertir imagen a estructura personalizada**  $\mathcal{O}(W \times H)$
4. **Para cada una de las  $N$  costuras a eliminar** ( $N$  iteraciones):
  - **Inicializar tabla de costes y matriz de memoización**  $\mathcal{O}(W \times H)$
  - **Calcular coste con programación dinámica**  $\mathcal{O}(W \times H)$  (no se repiten cálculos)
  - **Encontrar la costura mínima**  $\mathcal{O}(W)$
  - **Eliminar la costura de la imagen**  $\mathcal{O}(H)$
  - **Guardar imagen intermedia (si aplica)**  $\mathcal{O}(W \times H)$
5. **Convertir imagen de nuevo a CImg**  $\mathcal{O}(W \times H)$
6. **Guardar imagen final redimensionada**  $\mathcal{O}(W \times H)$

Para el caso peor donde  $N = W$ , el coste en memoria y tiempo en total es de:

$$\mathcal{O}(W^2 \times H)$$

Para el caso general, donde  $N < W$ , el coste en memoria y de tiempo se ajusta proporcionalmente:

$$\Theta(N \times W \times H)$$

## 3. Recursividad y tablas

A continuación, se muestran las funciones de recursividad empleadas:

$e(x, y)$ : energía del píxel $(x, y)$
$c(x, y)$ : energía de la costura mínima que parte del píxel $(x, y)$
$c(x, y) = \min\{c(x, y + 1), c(x - 1, y + 1), c(x + 1, y + 1)\} + e(x, y)$

Distinguimos el siguiente caso base:

$$c(x, \text{altoImagen}) = e(x, \text{altoImagen})$$

La tabla es una matriz de tamaño  $N \times M$ , donde  $N$  representa el ancho de la imagen y  $M$  su alto. Cada elemento de la tabla  $(i, j)$  corresponde a  $c(i, j)$ . La primera fila que podemos rellenar es aquella en la que  $j = \text{altoImagen}$ , es decir, todos los elementos  $c(i, \text{altoImagen})$  para  $1 \leq i \leq \text{anchoImagen}$ .

*	*	...	*	*
*	*	...	*	*
*	*	...	*	*
$e(1, M)$	$e(2, M)$	$\dots$	$e(N - 1, M)$	$e(N, M)$

#### 4. Diagrama de flujo

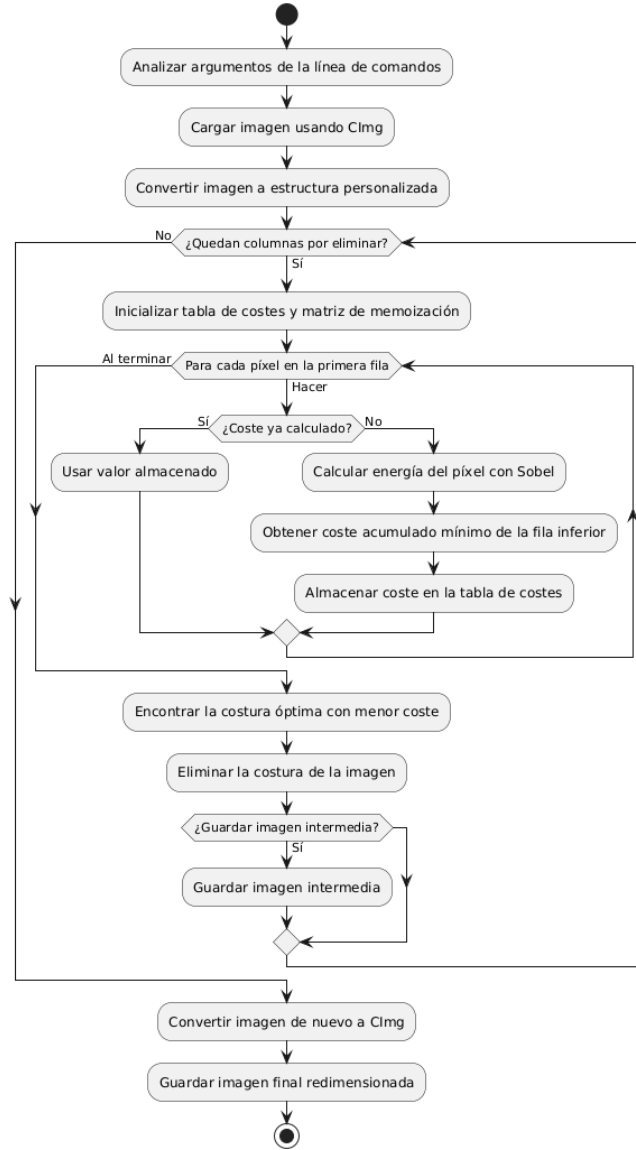


Figura 1: Flow diagram de nuestro algoritmo de eliminación de costuras.

## 5. Pruebas experimentales

Para verificar el correcto funcionamiento de nuestro algoritmo, se puede ejecutar de manera cómoda el script `ejecutar.sh`, el cual reduce al mínimo la anchura de cada imagen de un conjunto de prueba de imágenes con características diversas y almacena, de forma ordenada, tanto el resultado final como cada imagen intermedia en el directorio `/tmp`.

La secuencia de imágenes de la figura 2 muestra experimentalmente la efectividad de nuestro algoritmo de reducción de anchura mediante *seam carving*. A medida que se eliminan columnas, se observa que las regiones de baja energía, como las costuras compuestas por el pelo blanco del gato, desaparecen primero, mientras que las áreas de alta energía, como la silueta del gato y en especial su cola, permanecen visibles durante más tiempo. Esto confirma que el algoritmo prioriza la preservación de detalles estructuralmente significativos, minimizando la distorsión en las zonas de mayor contraste y relevancia visual.

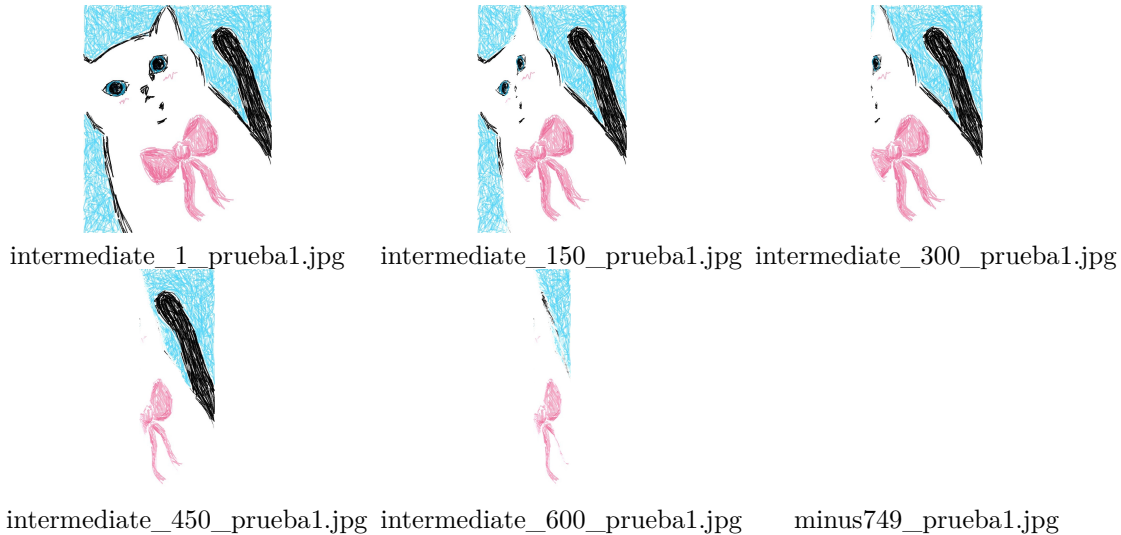


Figura 2: Evolución de la eliminación de costuras de una imagen con partes de energía alta y baja

El algoritmo funciona bien cuando existen zonas diferenciadas por su energía: preserva áreas de alta energía y elimina aquellas de baja energía. Sin embargo, cuando zonas importantes poseen baja energía o toda la imagen tiene alta energía, el rendimiento se ve afectado por las limitaciones propias del método. Aun así, considerando estas restricciones, los resultados obtenidos son satisfactorios: nuestro algoritmo es efectivo. Diferentes funciones de cálculo para la energía proporcionarían diferentes resultados.

Nombre	Ancho	Alto	Tiempo	Descripción
prueba1.jpg	705 px	907 px	22.287 s	Partes con energía alta y baja
prueba2.jpg	256 px	384 px	1.128 s	Ejemplo del enunciado
prueba3.jpg	640 px	480 px	8.711 s	Energía distribuida heterogéneamente
prueba4.png	267 px	373 px	5.382 s	Filas de energía homogénea

Cuadro 1: Pruebas experimentales realizadas en lab000.cps.unizar.es