

# Paralelización de un algoritmo de búsqueda de patrones en imágenes basado en la distancia de Hausdorff

Daniel Ortiz Nieto

Directores:

*Luis Miguel Sánchez, Antonio Berlanga*

21/10/2012

- Introducción
- Estado del Arte
  - ▶ Tecnologías de Paralelización
  - ▶ Métodos de detección de imágenes.
- Solución propuesta
  - ▶ Algoritmo
  - ▶ Implementaciones
- Pruebas
- Conclusiones
- Planificación

# Introducción

## Objetivos

- 1 Encontrar un concepto o propiedad matemática que permita el reconocimiento de imágenes que, además de cumplir con las condiciones deseables de robustez a la hora del análisis, posea una estructura de operación que facilite su optimización mediante la aplicación de técnicas de paralelización de las tareas necesarias.
- 2 Estudiar y seleccionar las mejores tecnologías de programación que nos permitan desarrollar una herramienta aplicando la idea anterior de la forma más óptima posible.
- 3 Idear un algoritmo con este concepto y codificarlo con las tecnologías seleccionadas tanto de forma secuencial como paralela.
- 4 Generar un plan de pruebas y comparar los resultados obtenidos para confirmar que el algoritmo cumple su función de manera óptima.

### ① Threads

- ▶ C++ 11
- ▶ OpenMP
- ▶ TBB

### ② SIMD

- ▶ CUDA
- ▶ OpenCL
- ▶ ArBB
- ▶ Cilk

# Estado del Arte

## Métodos de análisis de imágenes

- Representación de Invariantes
- Block Matching
- ① Esqueletos Topológicos
- ② Momentos Invariantes
- ③ Distancia Hausdorff
- ④ Mapeo Logarítmico Polar
- ⑤ Descriptores Geométricos
- ⑥ Perfiles de frontera
- ⑦ Descriptores de Fourier
- ⑧ Modelos de Forma activa
- ⑨ Transformada de Hough
- ⑩ Invariantes proyectivas 2D

# Solución propuesta

## Algoritmo

- La distancia Hausdorff.

Sean  $A = \{a_1, \dots, a_p\}$  y  $B = \{b_1, \dots, b_q\}$  dos conjuntos finitos no vacíos, subconjuntos de un espacio métrico  $(M, d)$ , se define la distancia Hausdorff  $d_H(A, B)$

$$d_H(A, B) = \max\{h(A, B), h(B, A)\} \quad (1)$$

Donde

$$h(A, B) = \max \min \|a - b\| \quad (2)$$

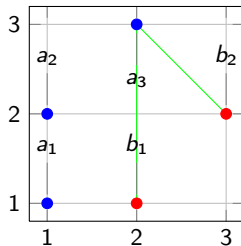
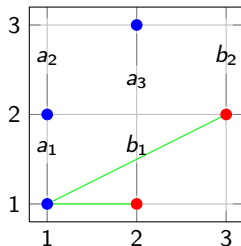
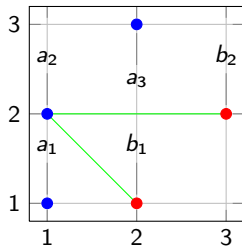
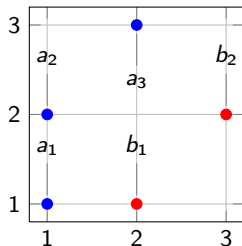
$$h(B, A) = \max \min \|b - a\| \quad (3)$$

Con  $a \in A$ ,  $b \in B$  y  $\|\cdot\|$  es una norma entre los puntos de  $A$  y  $B$ , como pueda ser la  $L_2$  o la norma Euclídea.

# Solución Propuesta

## Algoritmo

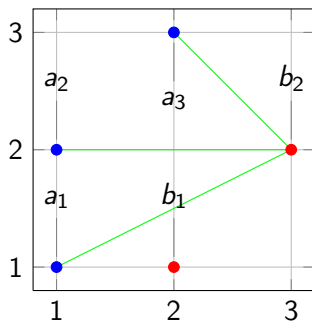
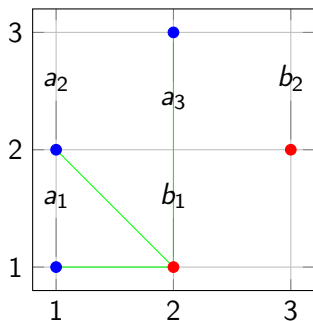
- La distancia Hausdorff.



# Solución Propuesta

## Algoritmo

- La distancia Hausdorff.

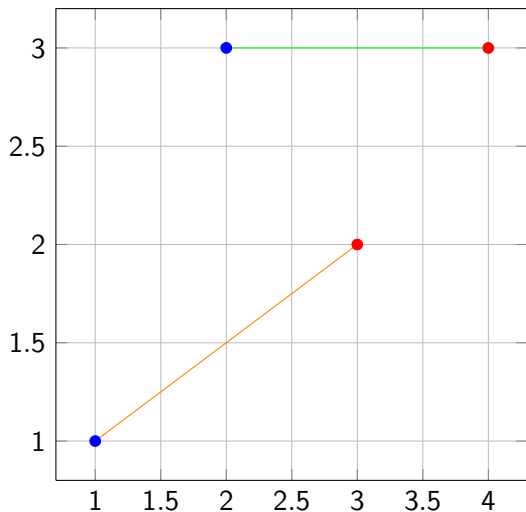




# Solución Propuesta

## Algoritmo

- La distancia Hausdorff.



# Solución Propuesta

## Algoritmo

**Data:**  $A = \text{bitmap1}(m \times n)$ ,  $B = \text{bitmap2}(p \times q)$ .

**Result:** Distancia  $d_H$  entre A y B.

**begin**

$A \leftarrow \text{leer}(\text{bitmap1})$

$B \leftarrow \text{leer}(\text{bitmap2})$

$A' \leftarrow \text{detectarBordes}(A)$

$B' \leftarrow \text{detectarBordes}(B)$

$d_{A,B} \leftarrow -1$

$d_{B,A} \leftarrow -1$

$d_{\text{temp}} \leftarrow -1$

$d_H \leftarrow 0$

**for**  $a \in A'$  **do**

**if**  $\text{color}(a) = \text{blanco}$  **then**

$A'_c \leftarrow \text{coordenadasXY}(a)$

**for**  $b \in B'$  **do**

**if**  $\text{color}(b) = \text{blanco}$  **then**

$B'_c \leftarrow \text{coordenadasXY}(b)$

# Solución Propuesta

## Algoritmo

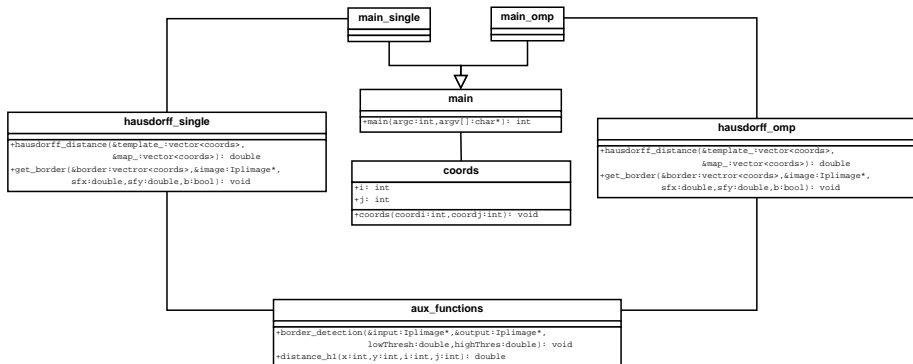
```
begin
  for  $a' \in A'_c$  do
    for  $b' \in B'_c$  do
      if  $d_{temp} = -1$  then
         $d_{temp} \leftarrow \|a' - b'\|$ 
       $d_{temp} \leftarrow \min(d_{temp}, \|a' - b'\|)$ ;
     $d_{A,B} \leftarrow \max(d_{temp}, d_{A,B})$ ;
   $d_{temp} \leftarrow -1$ ;
  for  $b' \in B'_c$  do
    for  $a' \in A'_c$  do
      if  $d_{temp} = -1$  then
         $d_{temp} \leftarrow \|b' - a'\|$ 
       $d_{temp} \leftarrow \min(d_{temp}, \|b' - a'\|)$ ;
     $d_{B,A} \leftarrow \max(d_{temp}, d_{B,A})$ ;
   $d_H \leftarrow \max(d_{A,B}, d_{B,A})$ 
```

# Solución Propuesta

## Codificación

- C++
  - ▶ OpenCV
- OpenMP

## Diagrama de Clases:

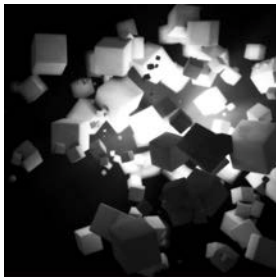


# Solución Propuesta

## Funcionamiento



(a) Original



(b) Escala de grises



(c) Contornos

Patrón del primer grupo de pruebas.



(d) Letra normal.



(e) Letra con 'agujero'.

- 10 Parejas desde 50x50 hasta 1125x1125 pixeles.

# Pruebas

## Conjunto B



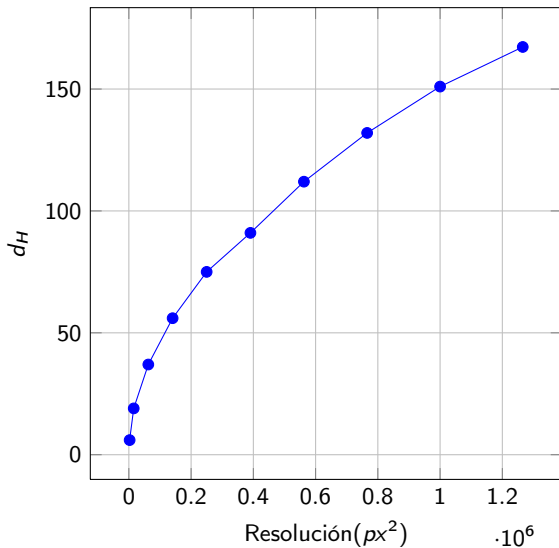
- Misma resolución, 1000x1000, complejidad creciente.

- **Tipo de CPU:** Intel Westmere EX.
- **Reloj de la CPU:** 1.87 Ghz.
- **Sockets:** 4.
- **núcleos por Socket:** 6.
- **Hilos por núcleo:** 2.
- **Total Hilos de ejecución disponibles:**  $2 \cdot 6 \cdot 4 = 48$ .
- **Niveles de Caché:** 3.
- **Tamaño por nivel de Caché:** Nivel 1: 32 kb. Nivel 2: 256kb. Nivel 3: 18MB.
- **Sistema Operativo:** Ubuntu 10.10.



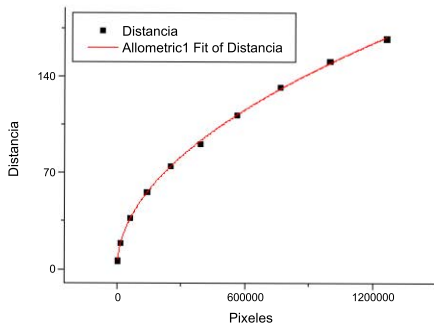
# Pruebas

## Resultados del grupo de pruebas A



# Análisis

## Ajuste de los resultados

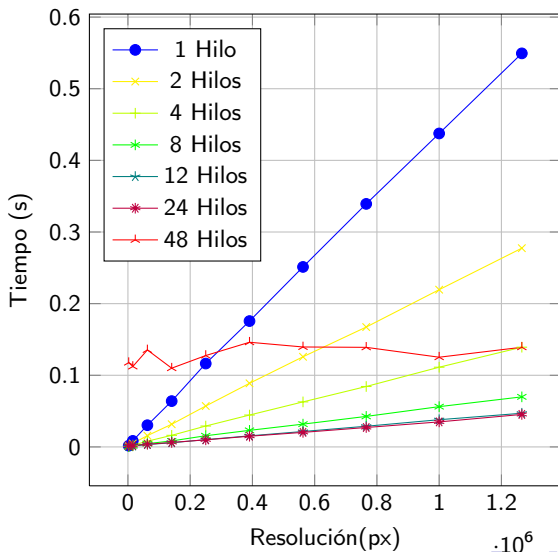


Ajuste:

- Ecuación de ajuste:  $y = a \cdot x^b$
- $a = 0.14 \pm 0.011$
- $b = 0.504 \pm 0.005$
- Coeficiente de regresión ( $r^2$ ) = 0.9995

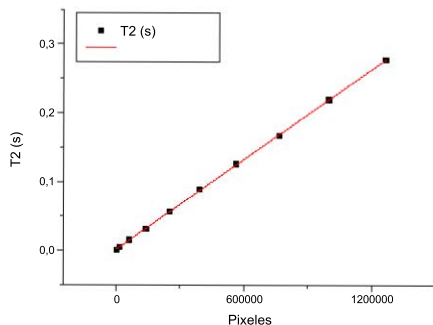
# Pruebas

## Resultados del grupo de pruebas A



# Análisis

Ajuste para los resultados con 2 hilos.

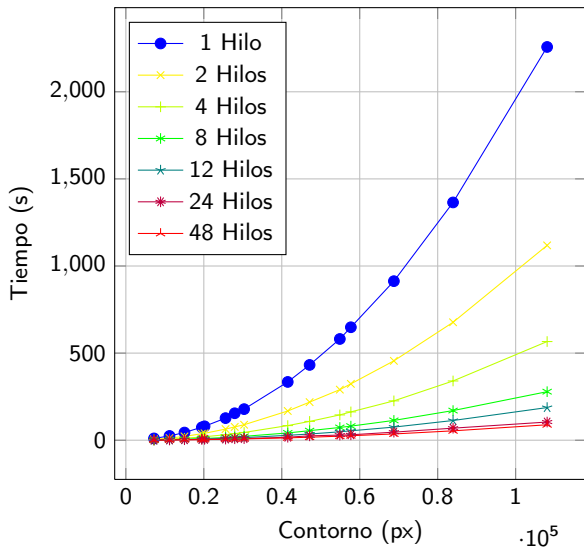


Ajuste:

- Ecuación de ajuste:  $y = m \cdot x + b$
- $m = 0,00397 \pm 0,00139$
- $b = 4,3407 \cdot 10^{-7} \pm 2,27832 \cdot 10^{-9}$
- Coeficiente de regresión ( $r^2$ ) = 0,99986

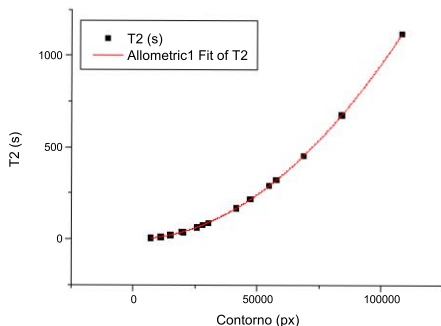
# Resultados

## Resultados del grupo de pruebas B



# Análisis

Ajuste para los resultados con 2 hilos.

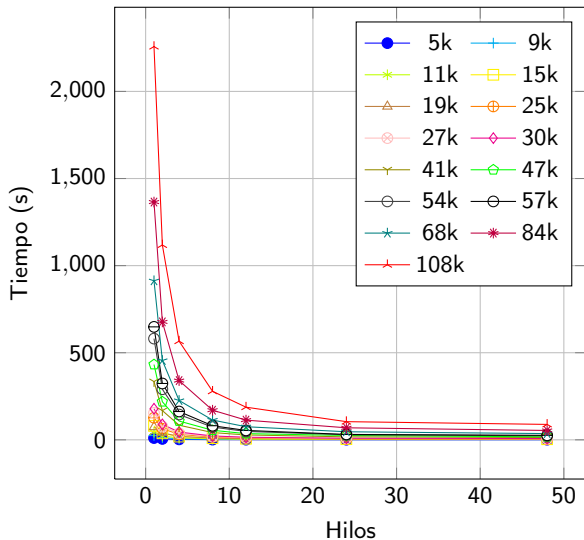


Ajuste:

- Ecuación de ajuste:  $y = a \cdot x^b$
- $a = 1,16422 \cdot 10^{-7} \pm 4,69871 \cdot 10^{-9}$
- $b = 1,98312 \pm 0,00354$
- Coeficiente de regresión ( $r^2$ ) = 0,99998

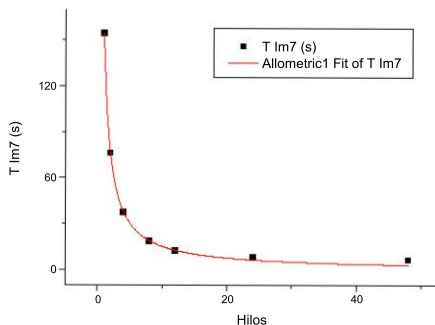
# Resultados

## Resultados del grupo de pruebas B



# Análisis

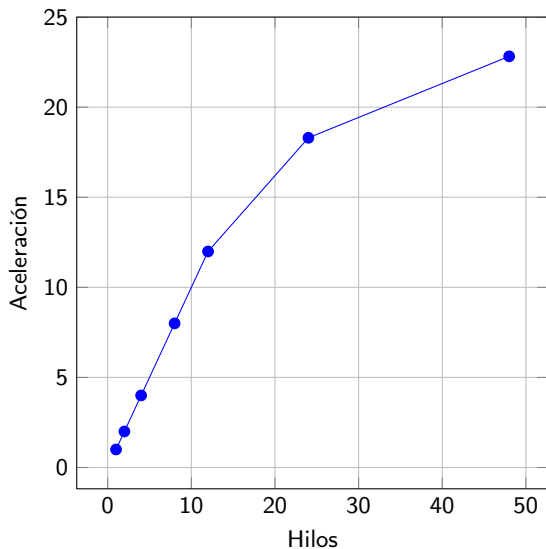
Ajuste para los resultados de la imagen 27k.



Ajuste:

- Ecuación de ajuste:  $y = a \cdot x^b$
- $a = 154,36974 \pm 1,70587$
- $b = -1,00386 \pm 0,02098$
- Coeficiente de regresión ( $r^2$ ) = 0,99894





# Conclusiones

- 1 La distancia Hausdorff ha demostrado ser una buena herramienta para el análisis de imágenes:
  - ▶ Proporciona un algoritmo cuya paralelización permite obtener mejoras lineales en el tiempo de ejecución.
  - ▶ Variación del tiempo de ejecución con el tamaño de los contornos:  
 $t_{th} = a \cdot x^2$ .
  - ▶ Tiempo de ejecución en relación con el número de hilos:  $t_{th} = t_1 \cdot th^{-1}$ .
- 2 La modificación de la distancia con la resolución:  $d_H = 0.14 \cdot \sqrt{x}$ .
- 3 El uso de contornos para el análisis de imágenes es prometedor por su ahorro en memoria y tiempo de computación.
- 4 El compilador de OpenMP ha resultado ser altamente eficiente en su tarea de paralelización.
- 5 Igualmente OpenCV ha proporcionado el soporte requerido de forma eficiente y sencilla.

- Realizar una implementación SIMD usando CUDA para observar el comportamiento con un nivel mayor de paralelización en imágenes de más resolución.
- Modificar el algoritmo para disminuir el número de puntos de los contornos, de forma que hagan falta menos elementos para el cálculo de la distancia.
- Estudiar si tomar sólo los vértices es una opción viable.
- Ampliar el programa incluyendo una función que permita descartar imágenes en base a la distancia obtenida.
- Cálculo automático del límite para el que se consideran suficientemente iguales los contornos comparados.
- Creación de un algoritmo que convolucione un contorno sobre una imagen mayor y permita encontrar coincidencias, permitiendo así el reconocimiento de patrones pequeños en imágenes grandes.
- Estudio de viabilidad de uso de la distancia Hausdorff para el reconocimiento de figuras en 3D y su posible aplicación para su uso en sistemas de visión artificial.

- Estado del arte:
  - ▶ Tecnologías de paralelización: 15 días.
  - ▶ Técnicas de análisis de imágenes: 20 días.
- Desarrollo:
  - ▶ Estudio y creación de un algoritmo basado en la distancia Hausdorff: 10 días.
  - ▶ Diseño e implementación secuencial del programa: 15 días.
  - ▶ Implementación de la paralelización con OMP: 4 días.
- Pruebas y resultados:
  - ▶ Diseño del conjunto de pruebas: 1 día.
  - ▶ Ejecución de las pruebas: 2 días.
  - ▶ Análisis de los resultados y conclusiones: 3 días.
- Total: 75 días / 600 horas.

A 40€/h esto nos da un total de **24.000€**

Fin de la presentación.