

Reconocimiento de Objetos en Imágenes utilizando el Método de SIFT (Scale Invariant Feature Transform)

Juliana Gambini

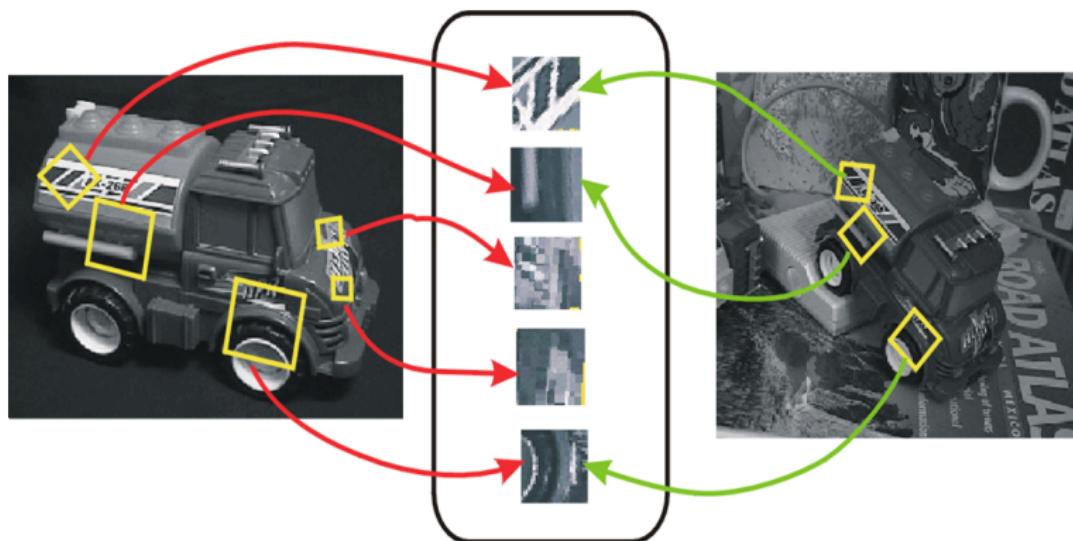
8 de junio de 2017

Organización

- 1 Objetivo
- 2 Resumen
- 3 Esp.Esc. Gaussiano
- 4 Keypoints
- 5 Orientaciones
- 6 Descriptores
- 7 Correspondencias

Objetivo

La idea es reconocer un objeto o una imagen dentro de otra imagen.



Descriptoros

El Método SIFT Transforma la imagen en un conjunto de descriptoros para luego hacer una correspondencia entre ellos.

- SIFT trabaja solamente sobre imágenes de niveles de gris.
- Los descriptoros son puntos que caracterizan la imagen.
- Son invariantes a escala, rotaciones y traslaciones.
- Son invariantes con respecto a cambios en la iluminación

Invarianza con respecto a la iluminación



Invarianza con respecto a la escala



Invarianza con respecto a la Rotación



Invarianza con respecto a la Rotación 3D o cambio en el punto de vista.



Invarianza con respecto a Transformaciones de Perspectiva.



SIFT

Este método transforma una imagen en un conjunto de descriptores que son robustos con respecto a transformaciones de traslación, rotación y escala.

Lowe en 2004

Fue desarrollado por Lowe en 2004

D. Lowe, "Distinctive image features from scale-invariant keypoints", International Journal of Computer Vision, vol. 60, pp. 91:110, 2004.

Organización

- 1 Objetivo
- 2 Resumen
- 3 Esp.Esc. Gaussiano
- 4 Keypoints
- 5 Orientaciones
- 6 Descriptores
- 7 Correspondencias

SIFT

Sucesivas operaciones: La primer parte está dedicada a encontrar puntos de interés o *keypoints*, la segunda a la extracción de características de cada *keypoint*

Organización

- 1 Objetivo
- 2 Resumen
- 3 Esp.Esc. Gaussiano**
- 4 Keypoints
- 5 Orientaciones
- 6 Descriptores
- 7 Correspondencias

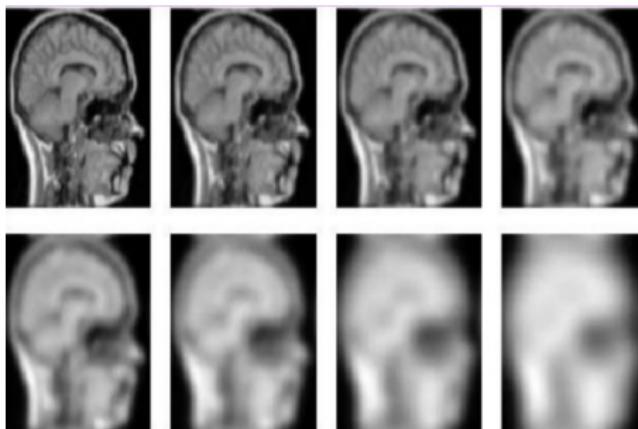
Construcción del Espacio Escala Gaussiano

Consiste en la aplicación de sucesivos filtros Gaussianos G_σ de diferente valor de escala σ . El algoritmo comienza con σ_0 y luego, $k\sigma_0$ para diferentes valores de k . En este caso cada filtro de Gauss debe tener tamaño

$$8\sigma + 1$$

El espacio escala Gaussiano se interpreta como un conjunto de fotos de la misma escena tomadas a diferente resolución.

Construcción del Espacio Escala Gaussiano



Conjunto de imágenes borroneadas

Paso 1

Con la aplicación de sucesivos filtros Gaussianos G_σ de diferente valor de escala σ se construye un conjunto de imágenes con diferentes niveles de borroneo.

paso 2

Luego, se baja la resolución de la imagen resultante y es ésta la imagen de entrada para construir otro conjunto de imágenes borroneadas.

Borronea, baja la resolución y vuelve a borrarnear



Octavas

Asociando a la Música

En música, una octava de notas $\{Do, Re, Mi, Fa, Sol, La, Si, Do\}$ posee doble frecuencia que la octava anterior. En el algoritmo original la resolución decrece por un factor de 2, por eso Lowe llamó «Octavas» a cada conjunto de imágenes con diferente nivel de borroneo pero con la misma resolución.

Upsampling y Subsampling

Sea I_{in} la imagen de entrada que posee una distancia entre pixels $\delta_{in} = 1$.

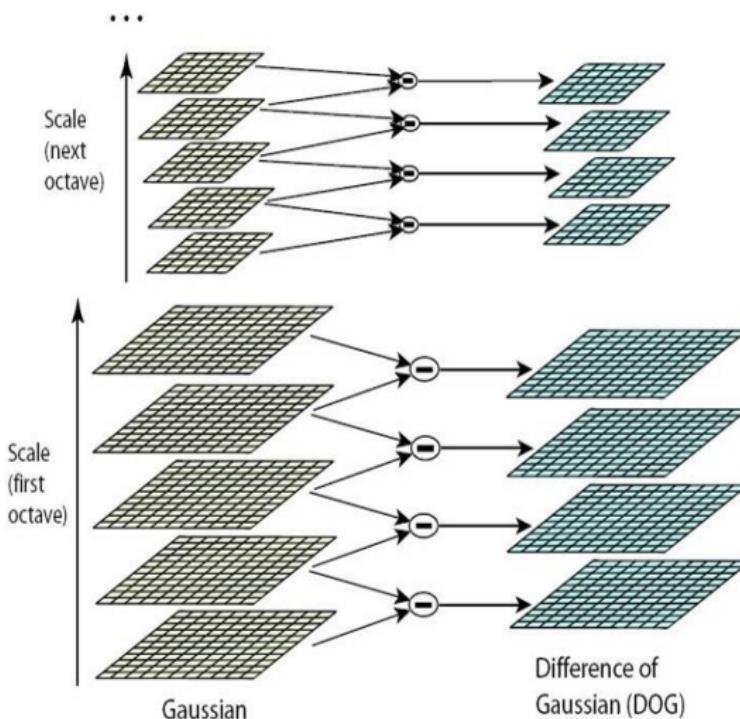
Upsampling

Agregamos pixels intermedios de tal manera que la distancia entre pixels es $\delta = 0,5$.

Subsampling

Eliminamos pixels intermedios de tal manera que la distancia entre pixels es $\delta = 2$.

Esquema: Construcción de octavas



Organización

- 1 Objetivo
- 2 Resumen
- 3 Esp.Esc. Gaussiano
- 4 Keypoints
- 5 Orientaciones
- 6 Descriptores
- 7 Correspondencias

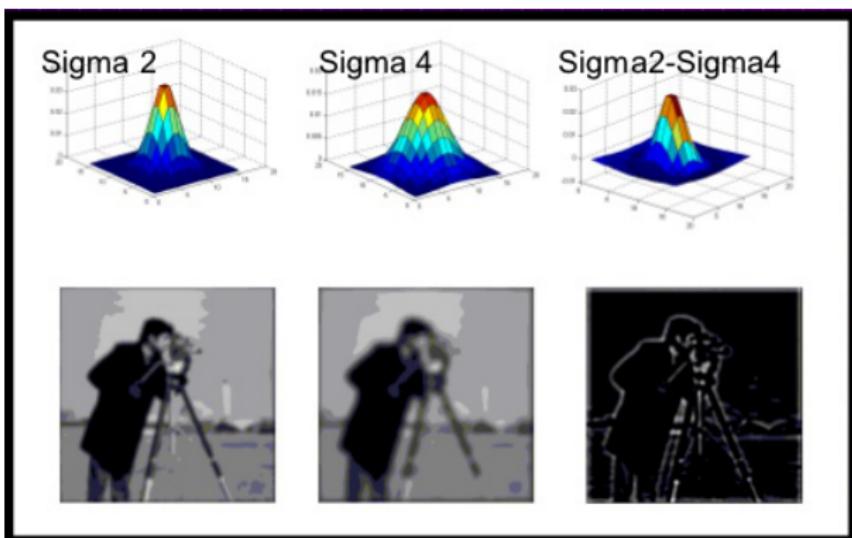
Diferencias Gaussianas

$$D(x, y, \sigma) = (G(x, y, k\sigma) - G(x, y, \sigma)) * I(x, y)$$

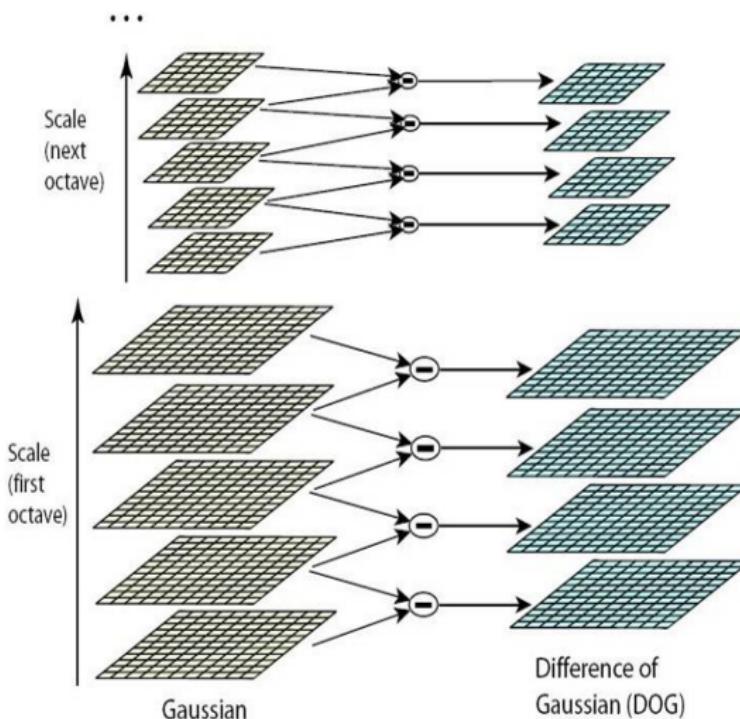
o también

$$D \circ G$$

Extremos de las diferencias de Gaussianas



Esquema: Diferencias de Gaussianas



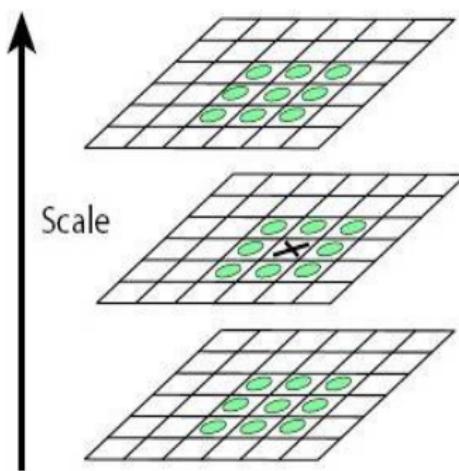
Extracción de los *keypoints* candidatos

Los puntos de interés (*keypoints*)

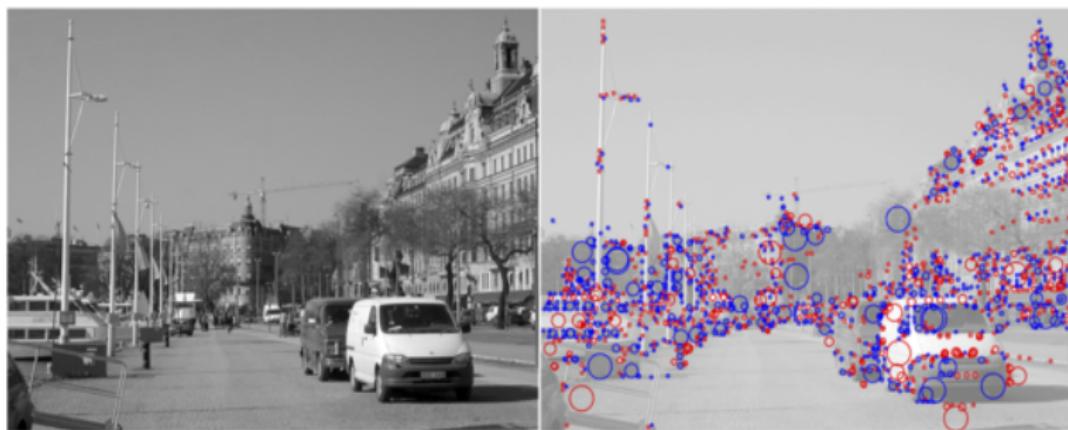
de las diferencias Gaussianas se definen como los extremos en 3D

Cada elemento se compara con sus vecinos en 3D, los vecinos son $26 = 3 \times 3 \times 3 - 1$.

Extremos 3D



Extremos 3D



Puntos de interés detectados en una imagen usando el método de los extremos de las diferencias de gaussianas. El radio de los círculos ilustra la escala. El color rojo indica que fueron mínimos, los azules que fueron máximos.

Refinamiento

Sin embargo, esta forma de encontrar los extremos es poco confiable porque resultan muchos extremos falsos. Entonces hay que hacer un refinamiento que permita rechazar los puntos con bajo contraste o aquellos que se encuentran sobre un borde.

Los puntos de borde no sirven

Porque no son invariantes con respecto a la escala

Filtrado de *keypoints* sobre bordes

Los puntos de borde no son estructuras que le interesen a SIFT porque no son invariantes a escala.

El hessiano de $D \circ G$ provee una caracterización para estos puntos indeseables.

Curvaturas principales

Las curvaturas principales de una superficie, miden qué tanto se "dobra" la superficie en ese punto y se define como los autovalores de la matriz de derivadas segundas de la superficie o el Hessiano.

En términos de los autovalores del Hessiano, un punto es de borde cuando los autovalores difieren mucho entre si o cuando $\lambda_1 \gg \lambda_2$

¿Cómo se descartan los puntos de borde? La misma idea que Harris

Para no tener que calcular los autovalores, se calcula $Tr(H)$ y $det(H)$. Como sabemos que $Tr(H) = \lambda_1 + \lambda_2$ y $det(H) = \lambda_1 \cdot \lambda_2$ con estos valores tenemos involucrados los autavoles. Supongamos que $\lambda_1 = r\lambda_2$

- r es el radio entre curvaturas principales.
- El Hessiano en cada punto se calcula por diferencias finitas.

¿Cómo se descartan los puntos de borde? La misma idea que Harris (cont.)

Entonces,

$$\frac{(Tr(H))^2}{det(H)} = \frac{(\lambda_1 + \lambda_2)^2}{\lambda_1 \cdot \lambda_2} = \frac{(r\lambda_2 + \lambda_2)^2}{r\lambda_2^2} = \frac{(r+1)^2}{r}$$

esta cantidad es mínima cuando $\lambda_1 = \lambda_2$. Entonces para chequear si el radio entre curvaturas principales es menor que un umbral r_u , solo tenemos que ver que

$$\frac{(Tr(H))^2}{det(H)} < \frac{(r_u + 1)^2}{r_u}$$

Por lo tanto vamos a descartar los keypoints tal que

$$\frac{(Tr(H))^2}{det(H)} > \frac{(r_u + 1)^2}{r_u}$$

Lowe utilizó $r_u = 10$

Quedan los más importantes



Organización

1 Objetivo

2 Resumen

3 Esp.Esc. Gaussiano

4 Keypoints

5 Orientaciones

6 Descriptores

7 Correspondencias

Asignación de Orientaciones

SIFT asigna orientaciones a cada *keypoint* utilizando una estimación del gradiente y del ángulo del gradiente, calculados por diferencias finitas con los pixels alrededor de cada *keypoint* en la imagen.

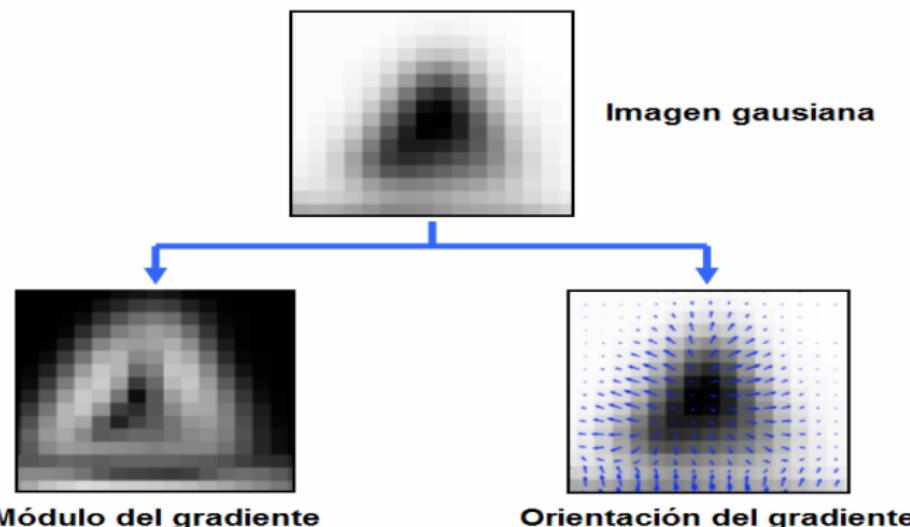
Definimos una región de 16×16 pixels alrededor del punto donde vamos a calcular la orientación, y a cada uno de estos pixels se les calcula su gradiente.

Asignación de Orientaciones

Sea (x, y, σ) un *keypoint*, entonces se estiman la magnitud y el ángulo del gradiente usando la imagen donde el *keypoint* fue encontrado .

Asignación de Orientaciones

Esto se realiza en todos los pixels alrededor del keypoint en una ventana de 16×16



Histograma de Orientaciones

Histograma de orientaciones

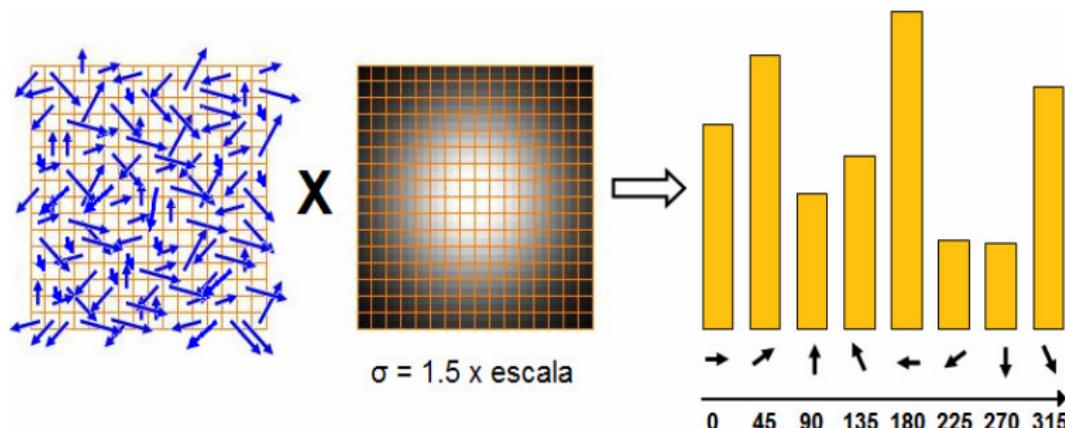
En este paso se agrupa la información en forma de histograma, uno para cada *keypoint*. Cada histograma posee 36 bins, con cada bin cubriendo 10 grados, de forma tal que cubre los 360°

Ponderación

A medida que se agrega una orientación al histograma, éste se multiplica por su módulo.

Ponderaciones

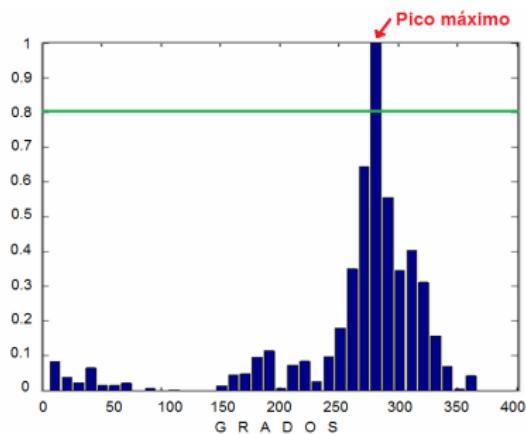
Las ponderaciones se realizan con el objetivo de darle más importancia a las orientaciones con módulos grandes.



Asignación de Orientaciones

Los picos más altos de cada histograma son las direcciones dominantes de los gradientes y por lo tanto la orientación que se le asigna al *keypoint* en cuestión.

Picos



Organización

- 1 Objetivo
- 2 Resumen
- 3 Esp.Esc. Gaussiano
- 4 Keypoints
- 5 Orientaciones
- 6 Descriptores
- 7 Correspondencias

Construcción de Descriptores

Hasta este punto, para cada keypoint se tiene almacenado la ubicación, la octava a la que pertenece, la escala y la orientación.

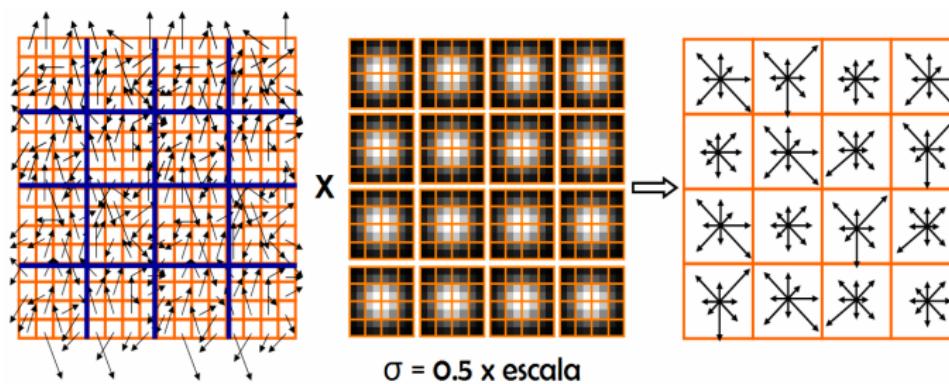
Con toda esta información, el siguiente paso es obtener un descriptor.

Construcción de Descriptores

Este paso se realiza sobre la imagen donde el *keypoint* fue encontrado. El proceso se realiza en regiones de 16×16 alrededor del *keypoint*.

Construcción de Descriptoros

La ventana de 16×16 pixels se divide en sub regiones de 4×4 pixels. Luego, se construye otro histograma con las orientaciones pero este solo de 8 bins para cada ventana de 4×4 .



Construcción de Descriptores

Luego se construye el descriptor contenido los valores de todas las orientaciones de los histogramas anteriores
El descriptor tendrá un tamaño $t = 128 = 8 * 16$, 8 es la cantidad de orientaciones y 16 es el tamaño de la ventana.

Organización

- 1 Objetivo
- 2 Resumen
- 3 Esp.Esc. Gaussiano
- 4 Keypoints
- 5 Orientaciones
- 6 Descriptores
- 7 Correspondencias

Cálculo de correspondencias

Un descriptor no es más que un conjunto de 128 elementos que indican las orientaciones más importantes alrededor de un punto de interés.

Si dos imágenes contienen descriptores muy similares, es probable que ambos estén describiendo una misma zona y por tanto sean correspondientes.

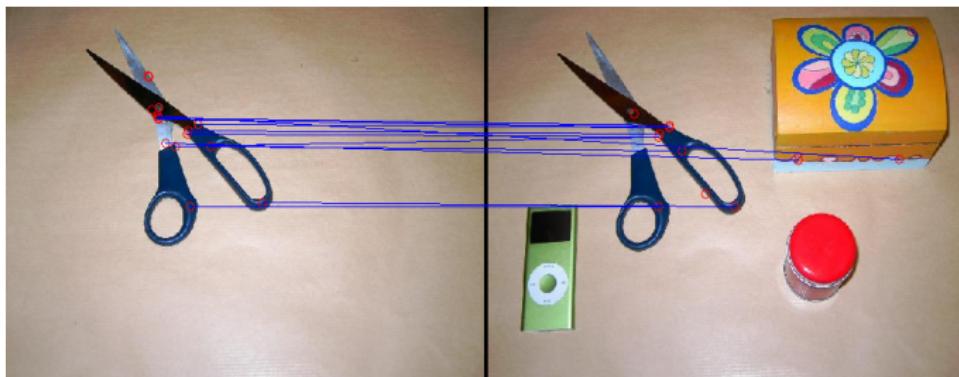
Cálculo de Correspondencias

Para calcular el grado de similitud entre ellos existen varias alternativas, una posible es la distancia euclídea.

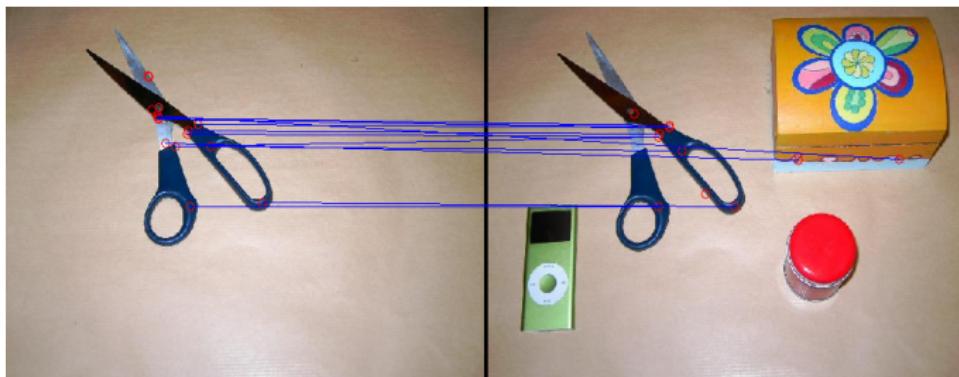
$$dif = \sqrt{\sum_{i=1}^{128} (a_i - b_i)^2} \quad (1)$$

donde a_i y b_i son los descriptores i de las imágenes A y B , respectivamente. Y luego se considera que las imágenes son iguales si este valor es menor que un umbral U_{abs} . Este umbral depende de la imagen.

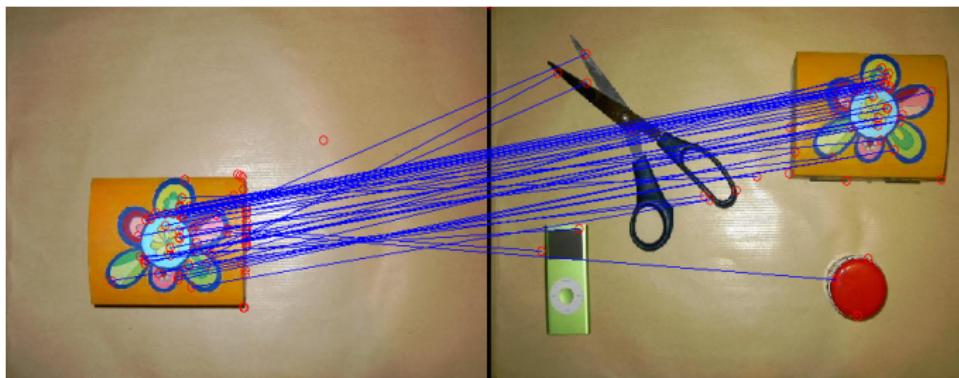
Ejemplos: Reconocimiento de objetos



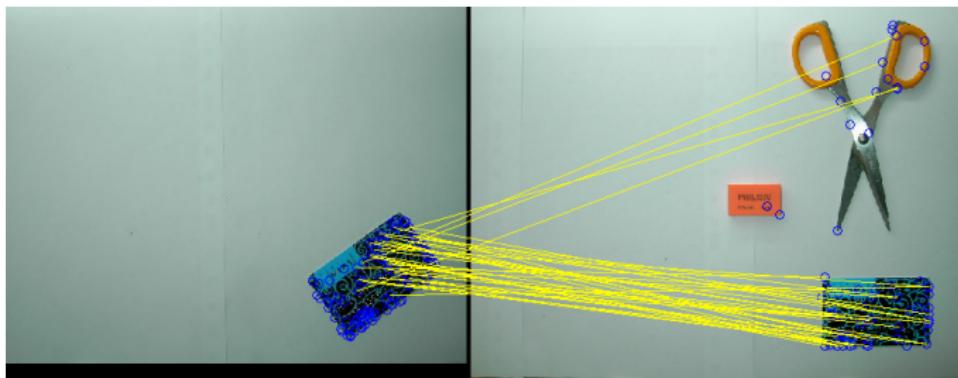
Ejemplos: Reconocimiento de objetos



Ejemplos: Reconocimiento de objetos

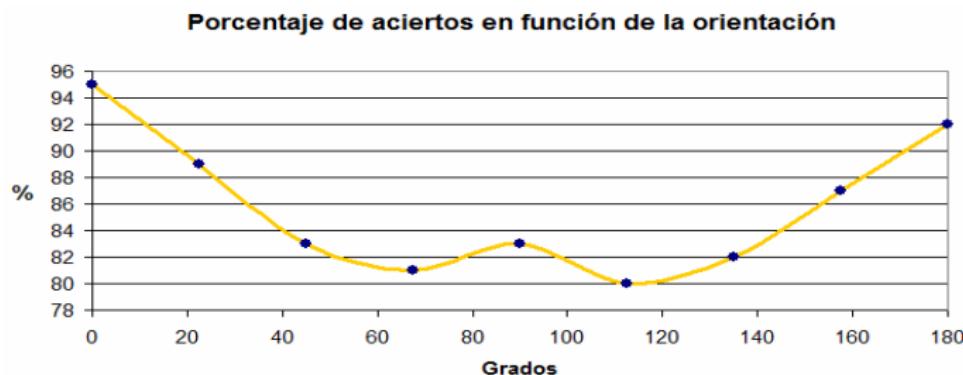


Reconocimiento de un objeto rotado



Invarianza con respecto a la rotación

Al realizar la prueba con diferentes grados de inclinación, en el intervalo $[0, \dots, 180]$, obtenemos los resultados del gráfico de la siguiente figura.



Resumen del Algoritmo

- Computar el Espacio-Escala Gaussiano.
- Computar las Diferencias Gaussianas (DoG).
- Encontrar los *keypoint* candidatos.
- Filtrar los *keypoints* ubicados sobre algún borde.
- Asignar una orientación de referencia a cada *keypoint*.
- Construir los descriptores.