

Introducción a la Visión Artificial con ROS

RAS-IEEE

José Carlos Rangel
jose.rangel@utp.ac.pa

FISC-UTP

15 de octubre de 2022



Introducción a la Visión Artificial con ROS

1 Introducción a la Visión Artificial

- Introducción a la Visión por Computadora
- Color
- Concepto de Imagen

Sección 1.1 | Introducción a la Visión por Computadora

Introducción a la Visión por Computadora I

Introducción a la Visión Artificial

- La Inteligencia Artificial (IA), busca que los computadores realicen trabajos para los cuales, cuando un humano los realice se requiere inteligencia.
- En la actualidad la IA como ciencia ha dado origen a diversas sub-ramas de desarrollo, las cuales se enfocan en cumplir o lograr la misión general de la IA pero enfocada solo en uno de los aspectos de la inteligencia humana.
- La Visión Artificial (VA) nace bajo este concepto y entre sus metas esta la emulación del comportamiento humano en lo relativo a la percepción visual o en otras palabras el sentido de la vista.



Introducción a la Visión por Computadora II

Introducción a la Visión Artificial

- Desde su concepción se considera uno de las entradas que mayor información puede proporcionar a un sistema.
- Por ende, es de suma importancia el desarrollo de métodos que permitan extraer la mayor cantidad de información de este tipo de entrada.
- La VA busca emular la visión humana y producir de una imagen de entrada, una salida de alto nivel, tal como lo haría una persona



Introducción a la Visión por Computadora III

Introducción a la Visión Artificial

- Como ciencia se apoya en otras para lograr su cometido, a la vez que es utilizada por otras áreas para cumplir metas globales.
- En la actualidad es una de las ramas de la IA que es más notada por las personas.
- En el mundo cotidiano está presente en sistemas de vigilancia, cámaras fotográficas hasta en coches y robots autónomos, en los cuales cumple una función primordial.



Introducción a la Visión por Computadora IV

Introducción a la Visión Artificial

Definición

Se puede definir la “Visión Artificial” como un campo de la “Inteligencia Artificial” que, mediante la utilización de las técnicas adecuadas, permite la obtención, procesamiento y análisis de cualquier tipo de información especial obtenida a través de imágenes digitales.



Introducción a la Visión Artificial

Sección 1.2 | Color

Color

1.2 Color

Color I

Color

- La luz y la reflexión de esta en las distintas superficies es producto de su longitud de onda.
- Cuando la luz entra en el sensor de imagen de una cámara, luz de las diferentes partes del espectro se descompone en las componentes **rojo, verde y azul** (RGB) que son los valores que se ven en un imagen.
- Esta descomposición guarda relación con los 3 tipos de **conos** presentes en la **retina** del ojo humano.
- Los sensores en cámaras y el ojo humano responden en cierta manera a estas diferentes longitudes onda.

Color II

Color

- La comparación de las respuestas de los distintos sensores arroja información de la energía en la longitud de onda de la luz entrante, esto es la información del color.
- El color se puede usar para identificar peculiaridades en las imágenes y para remover las sombras.
- Pero de igual manera como se vea un objeto en una imagen depende de la manera en la cual fue iluminado.



Reflexión y el Color de las cosas I

Color

- Cuando un haz de luz cae sobre un objeto, su superficie **absorbe algunas de las longitudes** de onda que la componen y refleja otras.
- Esto hace que solo las longitudes de **onda reflejadas** sean vistas por el ojo humano.
- Cuando se tiene por ejemplo un plátano, cuando este es visto bajo una luz blanca, parece amarillo, esto significa que el plátano tiene la capacidad de absorber las longitudes de onda que componen la luz blanca y solo refleja la amarilla.

Funcionamiento del Color

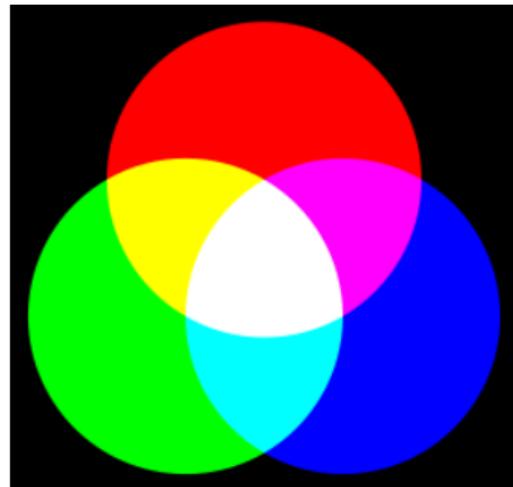
Teoría del Color



Modelo de Color I

Color

- Existen diversos modelos de color para las imágenes, el más común es el *Red, Green, Blue* o **Modelo RGB**.
- Este consiste en un **modelo aditivo** formado por los 3 colores primarios en la luz.
- En este modelo los 3 colores se van mezclando para producir una amplia gama de colores.



Modelo Aditivo con el resultado de las mezclas.

Canales I

Color

- En el modo *RGB* cada **color** es usualmente llamado **canal** y es representado con un entero con valor en el rango de [0 – 255].
- Estos derivan de los receptores de color del ojo humano.
- Una imagen *RGB* se compone entonces de 3 imágenes/mapas de color, una por cada color.
- Cada imagen representará la intensidad para cada uno de los colores del modelo.
- La unión (adición) de estas imágenes conformará el color verdadero de la imagen.



Canales II

Color

- En cada canal se tendrá un valor de color relacionado a cada pixel de la imagen.
- Si una imagen utiliza valores entre [0 – 255] para codificar el valor de intensidad en cada pixel, esta utiliza $8 - bits$ ($2^8 = 256$) por canal **para representar el color**.
- En este caso se dice que la imagen es de $24 - bits$ de profundidad.
- Este número **define la cantidad de colores** que se pueden utilizar para representar una imagen.



Canales III

Color

Canal *R*Canal *G*Canal *B*Canales *RGB*

Canales IV

Color



Canales *RG*



Canales *RB*



Canales *GB*



Canales *RGB*

Cámaras

1.2 Color

Sensor en las Cámaras Digitales I

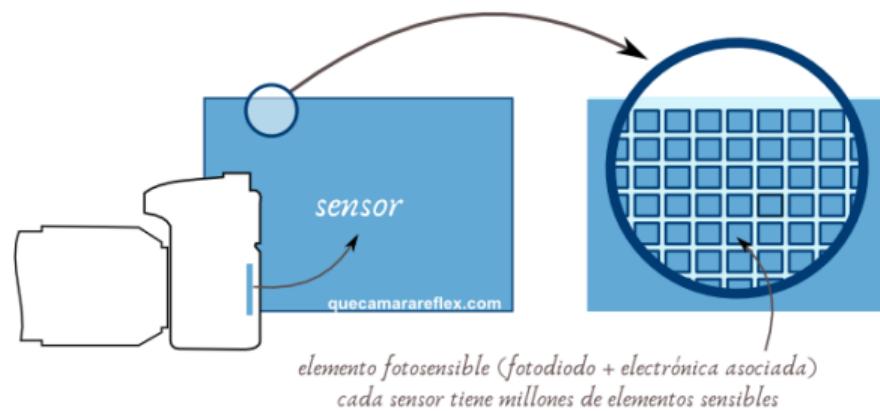
Color

- Los sensores están compuestos de millones de celdas fotosensibles microscópicas.
- Estas celdas se corresponden con los Megapíxeles de la cámara.
- Al realizar una fotografía cada celda del sensor se ocupa de capturar la luz proveniente de una parte de la toma.



Sensor en las Cámaras Digitales II

Color



Tomado de QuéCámaras

Sensor en las Cámaras Digitales III

Color

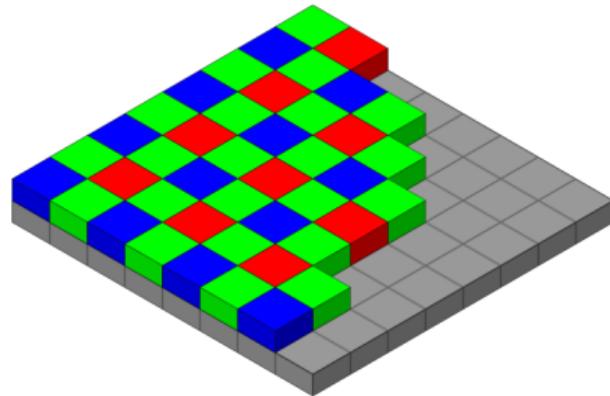
- Una celda es un fotodiodo que traduce la luz en electricidad y estas solo pueden medir la intensidad de luz que llega, más no el color de este haz de luz.
- Para analizar el color se utilizan filtros que descomponen la luz en sus tres componentes (**rojo**, **verde** y **azul**).
- Luego unas celdas reciben solo la componente roja, otras la verde y otras la azul.



Sensor en las Cámaras Digitales IV

Color

- Este filtro de separación recibe el nombre de Mosaico de Bayer



Tomado de [Wikipedia](#)



Sensor en las Cámaras Digitales V

Color

- El Mosaico tiene una distribución de 50% , 25% 25% de celdas Verdes, Rojas y Azules respectivamente.
- La predominancia del celdas verdes se debe a que el ojo humano es más sensible a este color.

Sección 1.3 | Concepto de Imagen

Formulación de una Imagen I

Concepto de Imagen

- Una imagen puede ser descrita como una función $2D$, $f(x, y)$
- (x, y) representan las coordenadas espaciales
- El valor de f en un punto (x, y) es proporcional al brillo o nivel de gris de la imagen (**Imagen en escala de Grises**).
- Cuando (x, y) y el brillo (f) son cantidades discretas finitas, la imagen es llamada **digital**.
- Por lo tanto $f(x, y)$ tomaría los siguientes valores:
 - $x \in [0, h - 1]$, donde h representa la altura de la imagen
 - $y \in [0, w - 1]$, donde w representa la anchura de la imagen
 - $f(x, y) \in [0, L - 1]$, donde $L = 256$ para una imagen de $8-bit$



Formulación de una Imagen II

Concepto de Imagen

- **Imagen de color** se puede representar de la misma manera, pero utilizando 3 funciones para representar los valores de los colores rojo, verde y azul.
- Cada una de estas funciones seguirá la misma formulación de $f(x, y)$ definida para la escala de grises.
- Se suelen identificar con las letras R , G , B y sus formulas pueden ser $fR(x, y)$, $fG(x, y)$ y $fB(x, y)$ respectivamente.



Formulación de una Imagen III

Concepto de Imagen

- Para **Imágenes en blanco y negro** se sigue la aproximación de una sola función.
- La única diferencia radica en que en este caso el valor de la función solo puede ser 0 (negro) o 255 (blanco)



Formulación de una Imagen IV

Concepto de Imagen

- Las **imágenes digitales** se pueden considerar como representaciones de escenas reales ya que utilizan cantidades discretas finitas.
- Para imágenes en escala de grises e imágenes en blanco y negro, se tiene una sola muestra para cada par (x, y) .
- Para imágenes a color se tendrán 3 muestras para cada par de coordenadas.



Píxeles I

Concepto de Imagen

- Un **Pixel** es la mínima unidad que conforma una imagen digital.
- Cada imagen esta compuesta por un gran conjunto de estos elementos.
- Se corresponde con cada uno de los pares de coordenadas (x, y) con las cuales se formula una imagen.



Píxeles II

Concepto de Imagen



Píxeles III

Concepto de Imagen

- Cada unidad tendrá asociada un valor producido por la función de color de la imagen.
- Utilizando la notación $ANCHO \times ALTO \times CANALES$ ($w \times h \times channel$) se suelen expresar las características básicas de un archivo de imagen.



Canales de Color I

Concepto de Imagen

Tomando en cuenta lo anterior, una imagen tendrá asociado un modelo de color. Este se puede ver de forma similar a la formulación de un imagen, pero en cada canal se representará un solo color.

Canales de Color II

Concepto de Imagen



Histograma de una Imagen I

Concepto de Imagen

- El **histograma** de una imagen refleja la distribución tonal de la imagen.
- En él se grafica la cantidad de píxeles que existen para cada valor tonal.
- Este valor es usualmente llamado la **frecuencia**.
- Para una imagen en escala de grises el rango de los valores del histograma es $[0, K - 1]$, donde K es 2 elevado al número de bits para representar la imagen.
- En el caso de imágenes de $8 - \text{bits}$ se tendría que $K = 2^8 = 256$



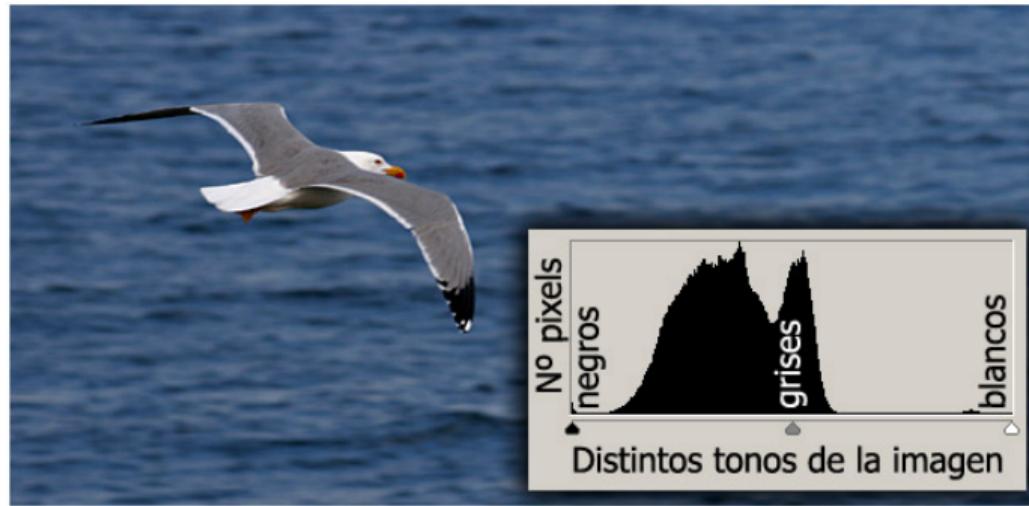
Histograma de una Imagen II

Concepto de Imagen

- Por lo tanto el histograma se podría definir de la siguiente manera:
 - * $h(i) = \text{NúmeroDePixellesConIntensidad}_i (i \in [0, 255])$
 - * Por lo cual, $h(80) = \text{NúmeroDePixellesConIntensidad}_{80}$
- Ejemplo de un histograma y su interpretación

Histograma de una Imagen III

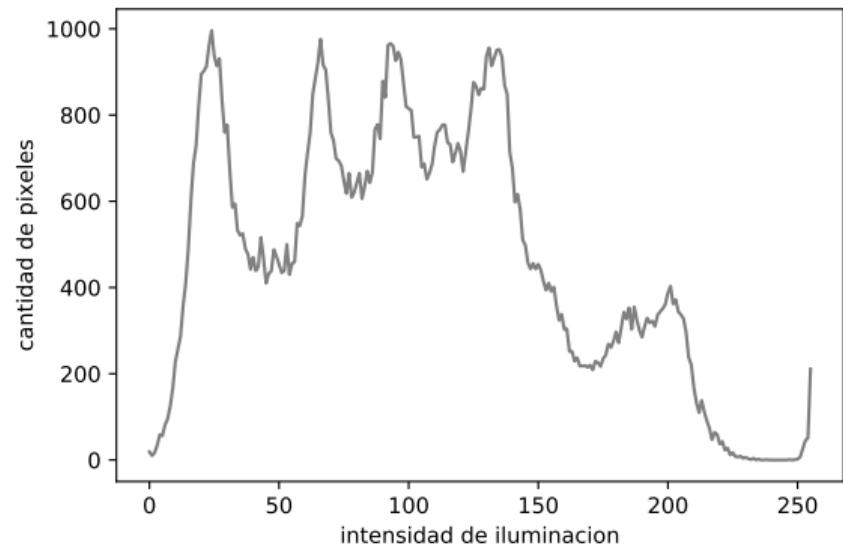
Concepto de Imagen



Ejemplos de Histograma I

Concepto de Imagen

Histograma para una imagen en escala de grises.



Ejemplos de Histograma II

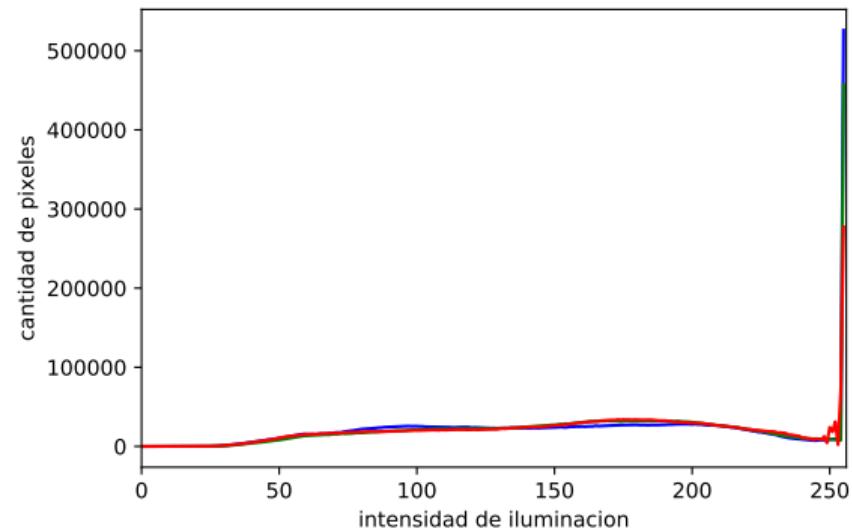
Concepto de Imagen

Diferentes Histogramas para Imágenes
a Color.



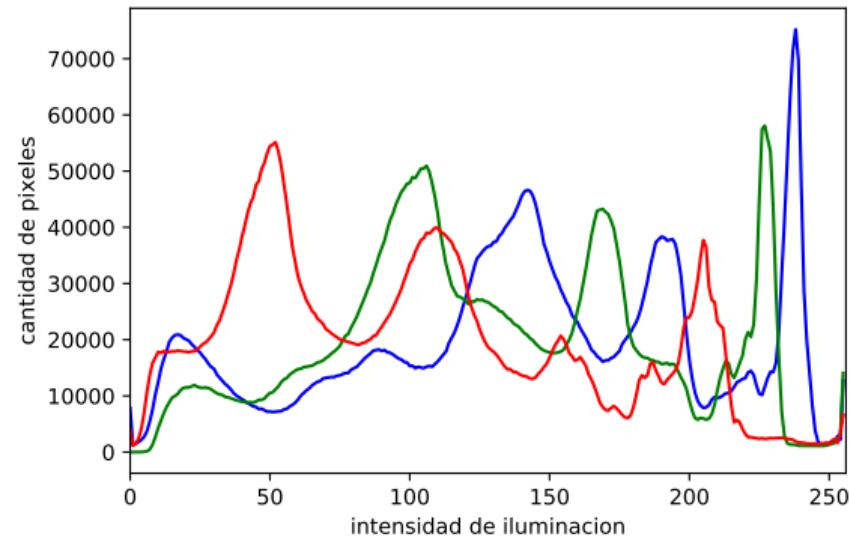
Ejemplos de Histograma III

Concepto de Imagen



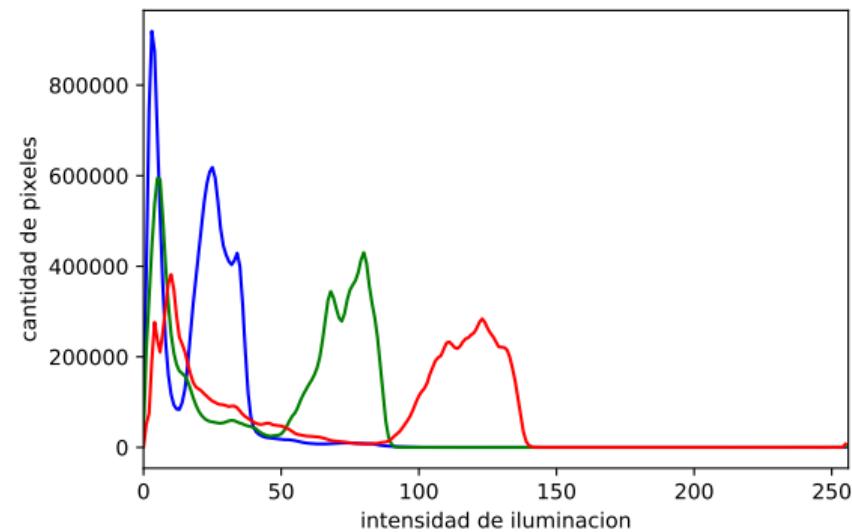
Ejemplos de Histograma IV

Concepto de Imagen



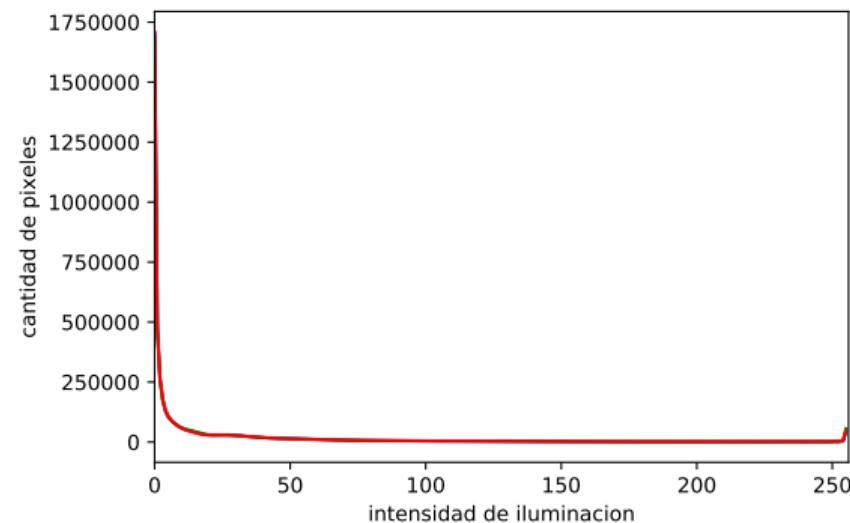
Ejemplos de Histograma V

Concepto de Imagen



Ejemplos de Histograma VI

Concepto de Imagen



Introducción a la Visión Artificial con ROS

2

Introducción a la Visión Artificial

- OpenCV
- Filtrado de imagen

Introducción a la Visión Artificial

Sección 2.9 | OpenCV

OpenCV I

Introducción a la Visión Artificial

- Es una librería *open source* de visión por computador, análisis de imagen y con algunas herramientas de aprendizaje automático.
- Para ello dispone de infinidad de algoritmos que permiten, con sólo escribir unas pocas líneas de código, identificar rostros, reconocer objetos, clasificarlos, detectar movimientos de manos, etc.
- Multiplataforma
- Puede programarse con C, C++, Python, Java, Matlab, entre otros.
- Se ha convertido en la librería por defecto para el desarrollo de aplicaciones con visión artificial.

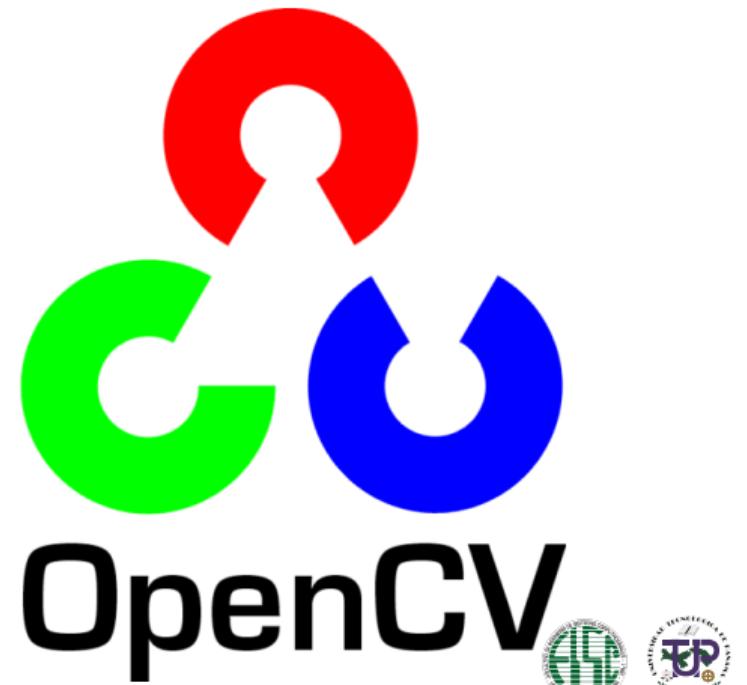


OpenCV II

Introducción a la Visión Artificial

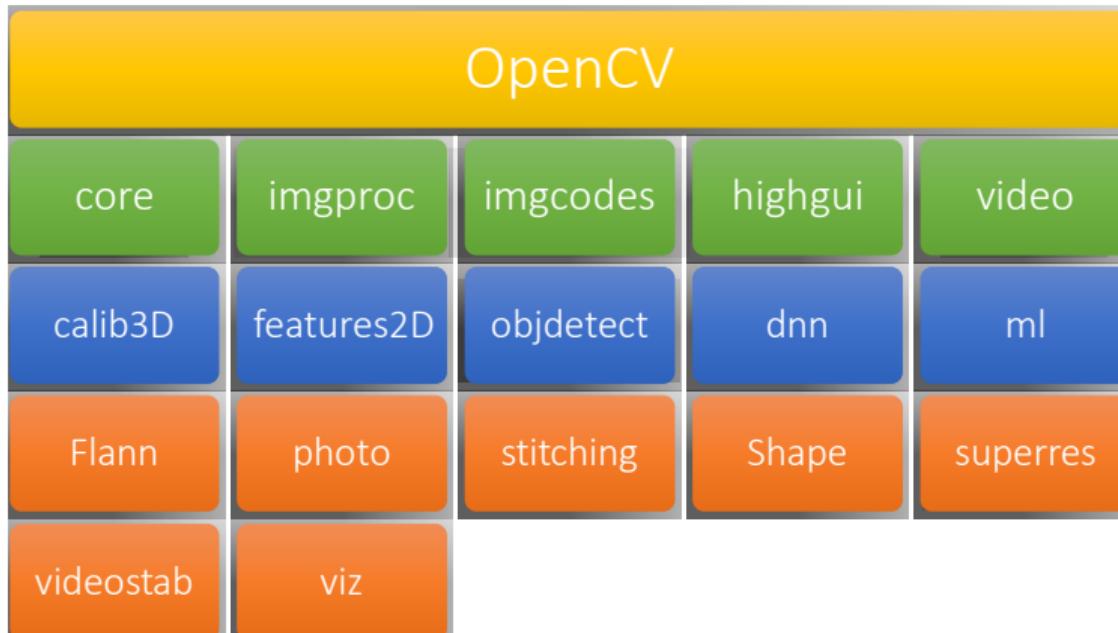
Incluye los siguientes módulos

- Procesamiento de Imágenes
- Análisis de Vídeo
- Calibración de Cámaras
- Reconstrucción 3D
- Características 2D
- Detección de Objetos
- Interfaces de alto nivel
- Vídeo I/O



OpenCV III

Introducción a la Visión Artificial



OpenCV IV

Introducción a la Visión Artificial

Dentro de sus aplicaciones se puede encontrar

- Interacción Humano Computador - HCI
- Identificación de Objetos
- Reconocimiento de Objetos
- Reconocimiento de Rostros
- Reconocimiento de Gestos
- Rastreo de Movimiento
- Procesamiento de Imágenes
- Robótica Móvil
- Realidad Aumentada
- Registro de imágenes de vista de calle
- Vigilancia Automatizada
- Análisis de Imágenes Médicas
- *Egomotion*
- ... entre otras

OpenCV V

Introducción a la Visión Artificial

Características

- Puede procesar al rededor de 30 frames por segundo
- Se puede integrar con diferentes entornos de desarrollo
- Optimizada para la ejecución
- Gran comunidad de desarrolladores que también dan soporte
- Soporte para el *Machine Learning*
- Cuenta con más de 2500 algoritmos optimizados y del estado del arte en visión computacional

Introducción a la Visión Artificial

Sección 2.10 | Filtrado de imagen

Filtrado de imagen I

Introducción a la Visión Artificial

Definición

Se puede definir como el procedimiento en el cual teniendo una imagen, esta es modificada utilizando una o varios métodos, que generalmente implican una ecuación matemática, para obtener una nueva versión de la imagen original, pero resaltando ciertas características que permiten un mejor procesamiento de esta. Este valor modificador recibe el nombre de **filtro**.



Filtrado de imagen II

Introducción a la Visión Artificial

Definición

Se trata de métodos para resaltar o suprimir, de forma selectiva, información contenida en una imagen a diferentes escalas espaciales, para destacar algunos elementos de la imagen, o también para ocultar valores anómalos.



Filtrado de imagen III

Introducción a la Visión Artificial

Filtro (Definición)

Un filtro se puede ver como un dispositivo tal que al pasar una señal compleja separa o selecciona las sub-señales componentes que satisfacen ciertas propiedades.



Filtrado de imagen IV

Introducción a la Visión Artificial

- Los filtros se pueden clasificar en 2 grandes grupos
 - ① **Filtros lineales:** si satisface los principios de linealidad y superposición. Es decir, el filtro lineal se caracteriza por el hecho de que el resultado de cada pixel se obtiene como combinación lineal de sus vecinos.
 - ② **Filtros no-lineales:** si no satisface los principios de linealidad y superposición. Como consecuencia de esto último, el resultado de un filtro no-lineal no se puede obtener mediante la convolución de la señal (imagen original) con la función de respuesta al impulso unidad.



Filtrado de imagen V

Introducción a la Visión Artificial

- Los Filtros Lineales se pueden dividir en 2 sub-grupos

- 1 Filtros lineales que operan en el dominio espacial. Que hacen uso de la operación de **convolución**, manipulando directamente los píxeles de la imagen original.
- 2 Filtros lineales que operan en el dominio frecuencial. Que hacen uso de la **Transformada de Fourier** de la imagen.



Filtrado de imagen VI

Introducción a la Visión Artificial

- Según el rango o intervalo de frecuencias seleccionado un filtro puede ser:

Filtro paso-bajo: cuando deja pasar todas las frecuencias espaciales iguales o inferiores a una dada, incluyendo al componente continuo (origen de coordenadas, índice de la luminancia media de la imagen o nivel medio de brillo) y elimina las superiores a ella.

Filtro paso-alto: cuando deja pasar todas las frecuencias espaciales iguales o mayores que una frecuencia dada y elimina las inferiores a ella.

Filtro paso-banda: cuando sólo deja pasar un rango de frecuencias espaciales y elimina todas las que se hallan fuera de tal rango.

Convoluciones

2.10 Filtrado de imagen

Convoluciones I

Introducción a la Visión Artificial

- El filtrado de imágenes consiste en la modificación de las matrices correspondientes a la imagen digitalizada mediante el procedimiento que se halla diseñado.
- Uno de los procedimientos más utilizado es la llamada **convolución de matrices**.
- Convolución es el tratamiento de una matriz por otra que se llama “**kernel**”.



Convoluciones II

Introducción a la Visión Artificial

- Son útiles para extraer la información relevante de una imagen.
- Es un procesamiento matemático sobre los píxeles de una imagen, con la intención de resaltar ciertos tipos de relaciones.
- Matemáticamente una **Máscara de Convolución** o filtros espaciales, son ventanas (o matrices) con valores dados por el tipo de relación que se desea resaltar.

Convoluciones III

Introducción a la Visión Artificial

- * Un ejemplo típico de estas ventanas puede ser el siguiente, el cual es un filtro especial de pasabajos.

$$\frac{1}{9} \times \begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Convoluciones IV

Introducción a la Visión Artificial

- * El proceso de aplicar un filtro espacial a una imagen se explica mediante el proceso de convolución, la cual esta dada por la siguiente ecuación:

$$f(x, y) * g(x, y) = \int_{-\infty}^{\infty} \int_{-\infty}^{\infty} f(\alpha, \beta)g(x - \alpha, y - \beta)d\alpha d\beta$$

- * Donde:
 - x y y son las coordenadas del pixel
 - α y β son variables para controlar el desplazamiento de la matriz de convolución



Convoluciones V

Introducción a la Visión Artificial

- * De igual manera el resultado de la convolución para un pixel de la imagen resultado se puede definir utilizando la siguiente formula:

$$p_{conv}(x, y) = \sum_{s=-a}^a \sum_{t=-b}^b g(s, t)f(x + s, y + t)$$

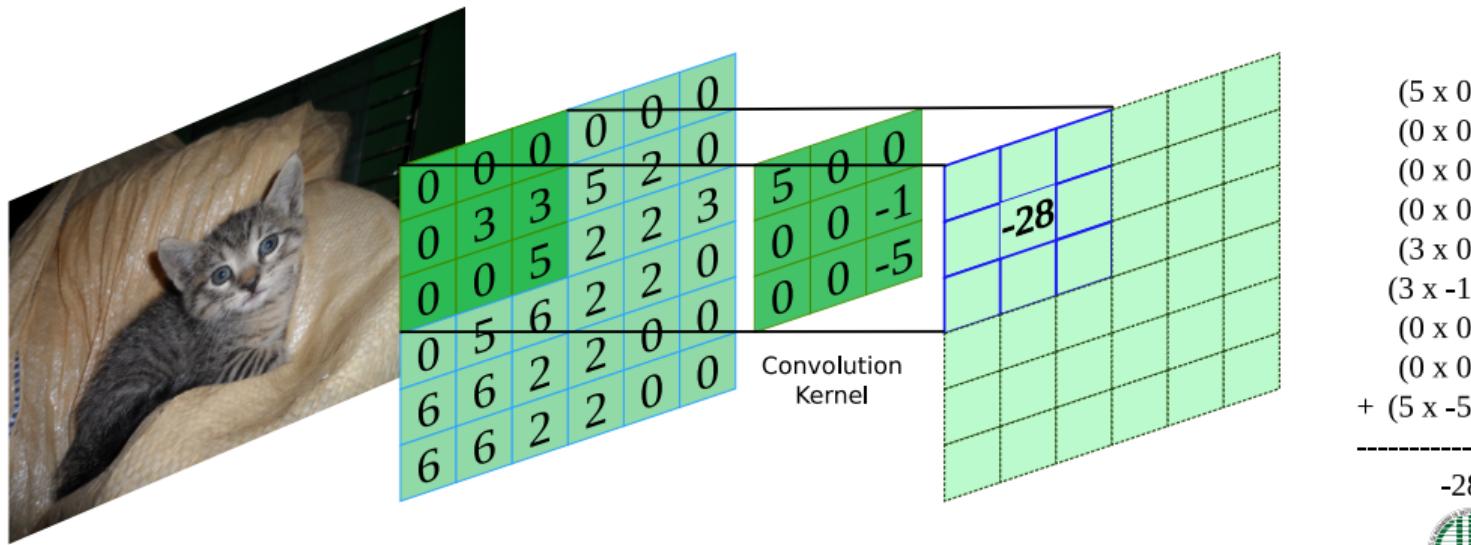
- * Por lo cual se puede decir que el resultado para un pixel, es la sumatoria de los productos de los valores del filtro (g) por los valores de los píxeles cercanos al este.



Convoluciones VI

Introducción a la Visión Artificial

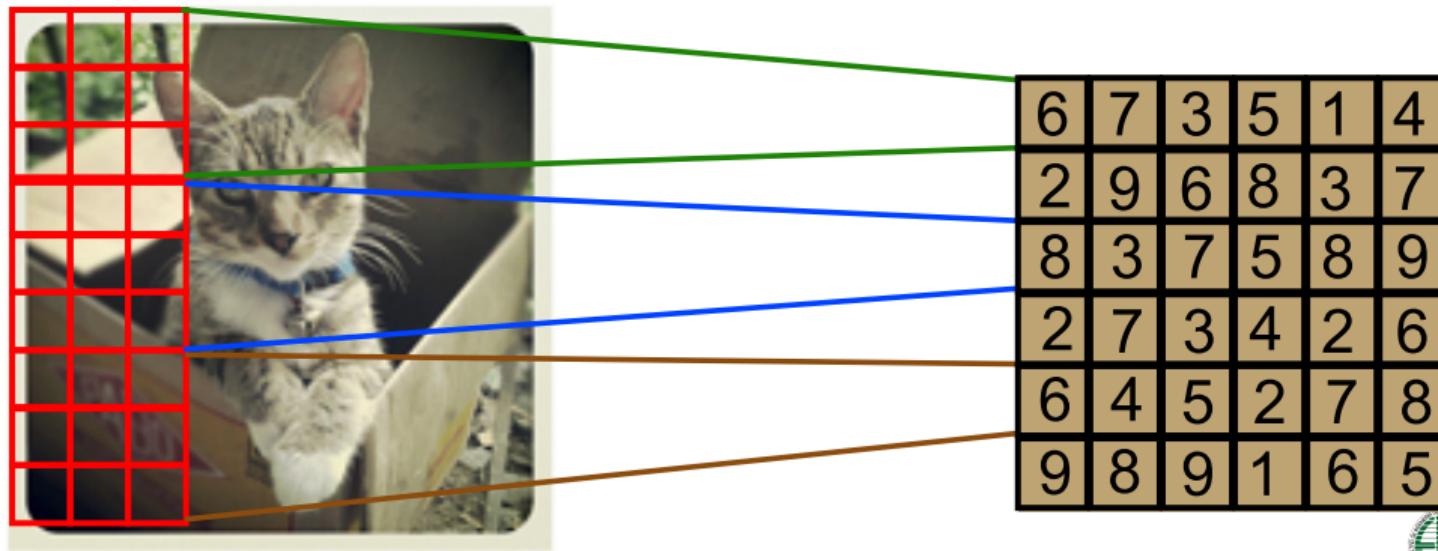
- De manera gráfica una convolución se podría representar de la siguiente manera



Convoluciones VII

Introducción a la Visión Artificial

- Los píxeles de la imagen resultante se originarían según el siguiente esquema.



Convoluciones VIII

Introducción a la Visión Artificial

Ejemplo de Convolución

- Teniendo la imagen $f(x, y)$ y el filtro $g(x, y)$, respectivamente.

$$f(x, y) = \begin{bmatrix} 17 & 24 & 1 & 8 & 15 \\ 23 & 5 & 7 & 14 & 16 \\ 4 & 6 & 13 & 20 & 22 \\ 10 & 12 & 19 & 21 & 3 \\ 11 & 18 & 25 & 2 & 9 \end{bmatrix} \quad g(x, y) = \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$



Convoluciones IX

Introducción a la Visión Artificial

Ejemplo de Convolución

- El resultado de la Convolución $f * g$ sería el siguiente

$$f * g = \begin{bmatrix} 11 & 10 & 12 \\ 11 & 13 & 15 \\ 13 & 15 & 14 \end{bmatrix}$$

- * Se trabajan con imágenes discretizadas, por lo tanto, cuando se obtiene un valor decimal, este se trunca.



Convoluciones X

Introducción a la Visión Artificial

Ejemplo de Convolución

- Realizando los cálculos para el elemento de la fila 2, columna 1 de la imagen resultado, se obtendría lo siguiente

$$\begin{bmatrix} 23 & 5 & 7 \\ 4 & 6 & 13 \\ 10 & 12 & 19 \end{bmatrix} * \begin{bmatrix} 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \\ 1/9 & 1/9 & 1/9 \end{bmatrix}$$

$$\frac{23 + 5 + 7 + 4 + 6 + 13 + 10 + 12 + 19}{9} = \frac{99}{9} = 11$$



Convoluciones XI

Introducción a la Visión Artificial

- Ejemplos de máscaras de filtros más utilizadas.

Enfoque

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 5 & -1 \\ 0 & -1 & 0 \end{bmatrix}$$

Desenfoque

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & 1 & 1 \\ 1 & 1 & 1 \end{bmatrix}$$

Realce de Bordes

$$\begin{bmatrix} 0 & 0 & 0 \\ -1 & 1 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Repujado

$$\begin{bmatrix} -2 & -1 & 0 \\ -1 & 1 & 1 \\ 0 & 1 & 2 \end{bmatrix}$$



Convoluciones XII

Introducción a la Visión Artificial

Detección de Bordes

$$\begin{bmatrix} 0 & 1 & 0 \\ 1 & -4 & 1 \\ 0 & 1 & 0 \end{bmatrix}$$

Filtro de Tipo Sobel

$$\begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix}$$

Filtro de Tipo Sharpen

$$\begin{bmatrix} 1 & -2 & 1 \\ -2 & 5 & -2 \\ 1 & -2 & 1 \end{bmatrix}$$

Filtro Norte

$$\begin{bmatrix} 1 & 1 & 1 \\ 1 & -2 & 1 \\ -1 & -1 & -1 \end{bmatrix}$$

Filtro Este

$$\begin{bmatrix} -1 & 1 & 1 \\ -1 & -2 & 1 \\ -1 & 1 & 1 \end{bmatrix}$$

Filtro de Tipo Gauss

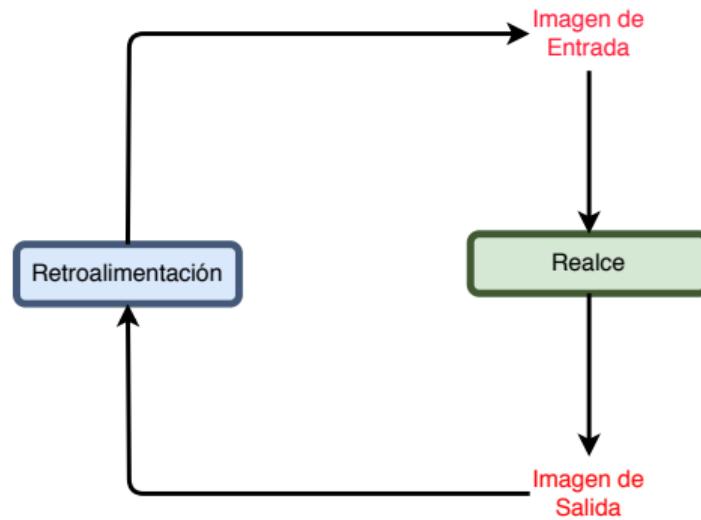
$$\begin{bmatrix} 1 & 2 & 3 & 1 & 1 \\ 2 & 7 & 11 & 7 & 2 \\ 3 & 11 & 17 & 11 & 3 \\ 2 & 7 & 11 & 7 & 1 \\ 1 & 2 & 3 & 2 & 1 \end{bmatrix}$$



Convoluciones XIII

Introducción a la Visión Artificial

Ejemplo de Aplicación de una Máscara de Detección de Bordes



Convoluciones XIV

Introducción a la Visión Artificial

Ejemplo de Aplicación de una Máscara de Convolución de Gauss

Original	Gauss
----------	-------

A black and white close-up photograph of a woman's face and shoulder. She is wearing a large, light-colored straw hat with a wide brim and a single long feather tucked behind her ear. Her hair is dark and styled down. She has a gentle expression, looking slightly off-camera. The background is dark and out of focus.

A black and white close-up photograph of a woman's face and upper shoulders. She is wearing a light-colored, wide-brimmed hat that casts a shadow over her eyes. Her hair is dark and wavy, visible at the bottom left. She has a neutral or slightly pensive expression, looking towards the right of the frame. The background is blurred, showing what appears to be an interior room with furniture.

Filtros en el Dominio de la Frecuencia

2.10 Filtrado de imagen

Filtros en el Dominio de la Frecuencia I

Introducción a la Visión Artificial

Definición

Este tipo de filtro aplica la Transformada de Fourier a una imagen, con el fin de obtener su representación en el dominio de la frecuencia. Esto con la finalidad de aplicar filtros que permitan realizar el suavizado, realce o detección de bordes.



Filtros en el Dominio de la Frecuencia II

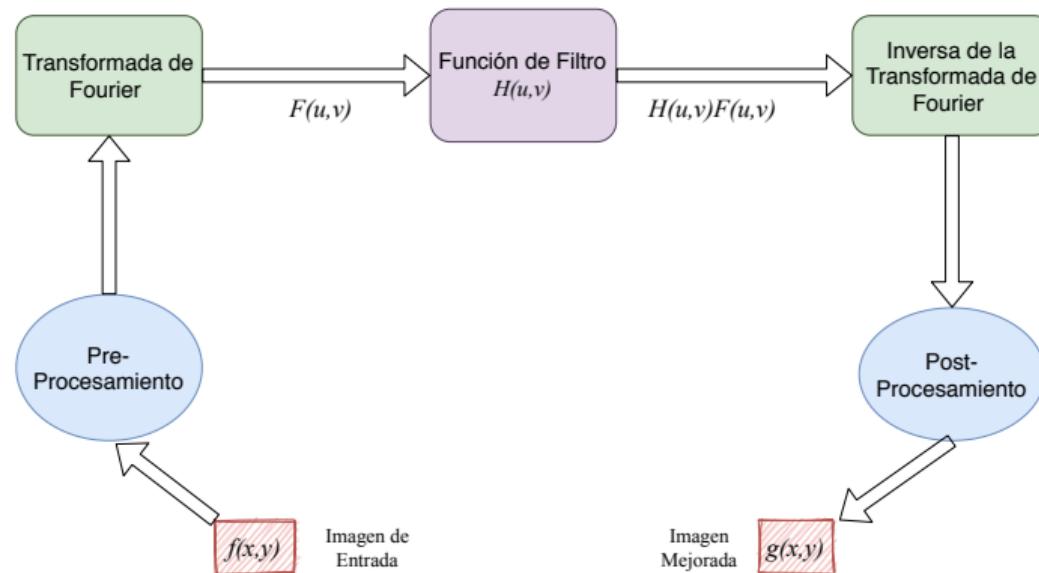
Introducción a la Visión Artificial

- Simplemente tenemos que ver la transformada de Fourier como una transformación de un dominio (el del espacio) a otro dominio (el de frecuencia) sin perder información de la imagen.
- El representar la información de la imagen en el dominio de la frecuencia tiene ventajas a la hora de aplicar algunos algoritmos y de determinar ciertas propiedades de la imagen.
- Cada armónico va a recoger un nivel de detalle, de cómo es la variación de los niveles de gris espacialmente.



Filtros en el Dominio de la Frecuencia III

Introducción a la Visión Artificial



Filtros en el Dominio de la Frecuencia IV

Introducción a la Visión Artificial

Los pasos para la aplicación de filtros en el dominio de la frecuencia se pueden resumir de la siguiente manera:

- ① Se multiplica cada entrada $f(x, y)$ por $(-1)^{x+y}$
- ② Se transforma la imagen en su dominio de la frecuencia mediante la Transformada Discreta de Fourier, $F(u, v)$
- ③ Se multiplica por un filtro de frecuencia $H(u, v)$, para cada (u, v) :
* $G(u, v) = H(u, v)F(u, v)$
- ④ Se calcula la inversa de la TDF de $G(u, v)$ (tomando la parte real), volviendo, así, al dominio del espacio.
- ⑤ Se vuelve a multiplicar por $(-1)^{x+y}$.



Filtros en el Dominio de la Frecuencia I

Introducción a la Visión Artificial

Filtro Paso-Bajo

- * Deja inalterables las bajas frecuencias y atenúa o elimina las altas frecuencias.
- * **Resultado:** elimina pequeños detalles y ruido.

$$H(u, v) = \begin{cases} 1 & \text{si } D(u, v) \leq D_0 \\ 0 & \text{si } D(u, v) > D_0 \end{cases}$$

Donde $D(u, v)$ es la distancia euclídea de (u, v) al origen del plano de frecuencias. y D_0 se le conoce como la frecuencia de corte.

Filtro Paso-Alto

- * Deja inalterables las altas frecuencias y atenúa o elimina las bajas frecuencias.
- * **Resultado:** realza los cambios bruscos en los niveles de grises, se usa para detectar bordes.

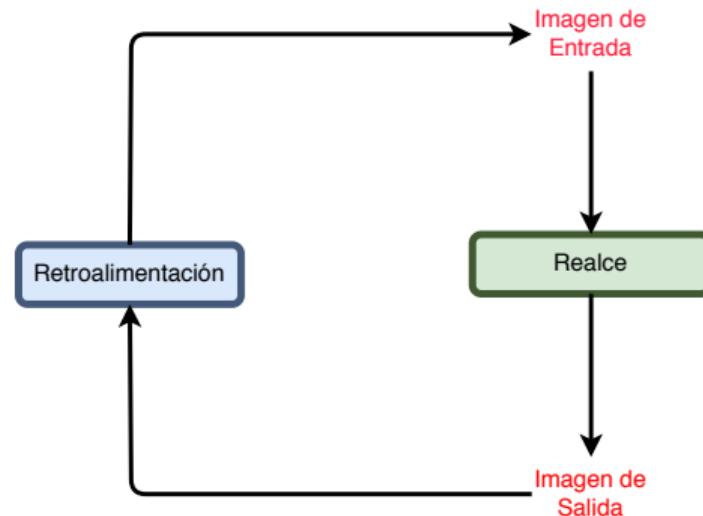
$$H(u, v) = \begin{cases} 0 & \text{si } D(u, v) \leq D_0 \\ 1 & \text{si } D(u, v) > D_0 \end{cases}$$



Filtros en el Dominio de la Frecuencia II

Introducción a la Visión Artificial

Ejemplo de Aplicación de una Filtros de Paso alto y bajo



Filtros No Lineales

2.10 Filtrado de imagen

Filtros No Lineales I

Introducción a la Visión Artificial

- Dentro de esta gama se pueden mencionar los **Filtros Estadísticos de Orden**.
- Este tipo de filtro funciona ordenando los valores en la vecindad de cada punto de menor a mayor.
- Luego de este ordenamiento se obtiene algún valor que representara la salida del filtro.
- Según su criterio, pueden ser:
 - * **Mínimo**: selecciona el valor más pequeño.
 - * **Máximo**: selecciona el valor más alto.
 - * **Mediana**: selecciona el valor en la posición intermedia.



Filtros No Lineales II

Introducción a la Visión Artificial

- **Filtro de Máximo:** Selecciona el mayor valor dentro de una ventana ordenada de valores de nivel de gris.

Ventaja Elimina el ruido pimienta (píxeles negros).

Inconveniente Funciona solo con el ruido pimienta y tiende a aclarar la imagen

- **Filtro de Mínimo:** Selecciona el menor valor dentro de una ventana ordenada de valores de nivel de gris.

Ventaja Elimina el ruido sal (píxeles blancos).

Inconveniente Funciona solo con el ruido es sal y tiende a oscurecer la imagen



Filtros No Lineales III

Introducción a la Visión Artificial

Ejemplo de Aplicación del Filtro de Máximos

Original



MAX/Dilate



Filtros No Lineales IV

Introducción a la Visión Artificial

Ejemplo de Aplicación del Filtro de Mínimos

Original



MIN/Erode



Filtros No Lineales V

Introducción a la Visión Artificial

- **Filtro de la Mediana:** Se reemplaza el valor por la mediana de los valores vecinos al pixel visitado. de nivel de gris.

Ventaja Da muy buenos resultados para ruido sal y pimienta

Inconveniente Al ser no lineal $\text{med}(A + B) \neq \text{med}(A) + \text{med}(B)$



Filtros No Lineales VI

Introducción a la Visión Artificial

Ejemplo de aplicación de Filtro de Mediana

- * Teniendo la imagen f

$$f = \begin{bmatrix} 66 & 18 & 7 & 43 & 24 \\ 56 & 112 & 107 & 182 & 117 \\ 44 & 171 & 12 & 63 & 170 \\ 68 & 169 & 110 & 4 & 50 \\ 96 & 167 & 56 & 21 & 1 \end{bmatrix}$$

- * Valores Cercanos: 4, 12, 63, 107, 110, 112, 169, 171, 182
- * Valor de Mediana: 110



Filtros No Lineales VII

Introducción a la Visión Artificial

Ejemplo de Aplicación del Filtro de Mediana

Original



Mediana



Introducción a la Visión Artificial con ROS

3

Detección de bordes

- Concepto
- Algoritmos
- Transformada de Hough

Detección de bordes

Sección 3.1 Concepto

Concepto I

Detección de bordes

Definición

La detección de bordes es una técnica de procesamiento de imágenes para encontrar los límites de los objetos dentro de las imágenes. Funciona mediante la detección de discontinuidades en el brillo.

La detección de bordes se utiliza para la segmentación de imágenes y la extracción de datos en áreas como el procesamiento de imágenes y la visión artificial.

Mathworks



Concepto II

Detección de bordes

- Incluye una variedad de métodos matemáticos que apuntan a identificar puntos en una imagen digital en los que el brillo de la imagen cambia bruscamente o, más formalmente, tiene discontinuidades.
- Los puntos en los que el brillo de la imagen cambia bruscamente se organizan típicamente en un conjunto de segmentos de línea curva denominados bordes.

Concepto III

Detección de bordes

- El mismo problema de encontrar discontinuidades en las señales unidimensionales se conoce como detección de paso y el problema de encontrar discontinuidades de la señal a lo largo del tiempo se conoce como detección de cambio.
- Es una herramienta fundamental en el procesamiento de imágenes, particularmente en las áreas de detección y extracción de características.

Detección de bordes

Sección 3.2 | Algoritmos

Algoritmos I

Detección de bordes

Los algoritmos de detección de bordes comunes incluyen:

- Sobel
- Canny
- Prewitt
- Roberts
- Métodos de lógica difusa.

Operador Sobel I

Algoritmos

- El operador Sobel, a veces llamado operador Sobel-Feldman o filtro Sobel, se utiliza en el procesamiento de imágenes y en la visión por ordenador, especialmente en los algoritmos de detección de bordes, donde crea una imagen que enfatiza los bordes.
- Se llama así por Irwin Sobel y Gary Feldman, colegas del Laboratorio de Inteligencia Artificial de Stanford (SAIL).
- Sobel y Feldman presentaron la idea de un “Operador de Gradiente de Imagen Isótropo 3x3” en una charla en el SAIL en 1968.



Operador Sobel II

Algoritmos

- Es un operador de diferenciación discreta que calcula una aproximación del gradiente de la intensidad de una imagen.
- Combina el suavizado Gaussiano con la diferenciación.
- Su resultado para cada punto es la magnitud de la dirección del cambio de intensidad.
- Según este valor se puede determinar como de abrupto o suave es el cambio en cada punto analizado de la imagen.



Operador Sobel III

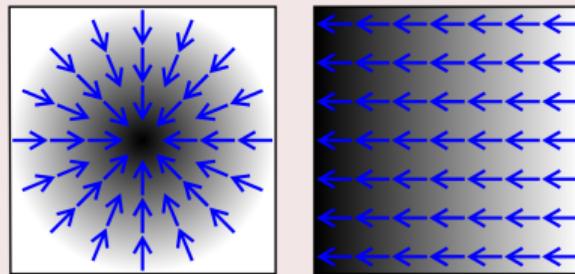
Algoritmos

Gradiente

Formulación Matemática

$$\nabla f = \frac{\partial f}{\partial x}x + \frac{\partial f}{\partial y}y \quad (1)$$

Representación Gráfica



Operador Sobel IV

Algoritmos

Operadores del Gradiente para Imágenes

Cambios Horizontales

$$G_x = \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \times I$$

Cambios Verticales

$$G_y = \begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \times I$$

Valor Final

$$G = \sqrt{G_x^2 + G_y^2}$$

$$\phi(xy) = \tan^{-1} \frac{G_y}{G_x}$$

Aproximación

$$|G| \approx |G_x| + |G_y|$$



Operador Sobel V

Algoritmos



Detector Canny I

Algoritmos

- El detector de bordes de Canny es un operador de detección de bordes que utiliza un algoritmo de varias etapas para detectar una amplia gama de bordes en las imágenes.
- Fue desarrollado por John F. Canny en 1986.
- Canny también produjo una teoría computacional de detección de bordes que explica por qué la técnica funciona.

Detector Canny II

Algoritmos

El proceso del algoritmo de detección de bordes de Canny se puede descomponer en 5 pasos diferentes [más información](#):

- ① Reducción de Ruido
- ② Calculo del Gradiente
- ③ *Non-maximum Suppression*
- ④ Doble Umbralizado
- ⑤ Seguimiento de Bordes



Detector Canny III

Algoritmos

① Reducción de Ruido

- Se aplica un filtro Gaussiano para suavizar la imagen y para eliminar el ruido, el cual puede afectar el gradiente.
- Los resultados de los detectores de bordes son muy sensibles al ruido

② Calculo del Gradiente

- Encuentra los gradientes de intensidad de la imagen
- Aplica un filtro de Sobel para calcular el gradiente y dirección en cada punto.



Detector Canny IV

Algoritmos

③ *Non-maximum Suppression*

- Luego de la detección del Gradiente se pueden observar bordes con diversos grosores.
- Aplicar un *Non-maximum Suppression* permite deshacerse de las respuestas extrañas(falsas) a la detección de bordes.
- Esto permite afinar/adelgazar los bordes detectados ya que la imagen debe tener bordes finos.
- Se hace analizando el gradiente y encontrando los máximos valores en las direcciones de los bordes detectados.



Detector Canny V

Algoritmos

④ Doble Umbralizado (Histéresis)

- Aplicar el doble umbral para determinar los bordes potenciales
- Esto detecta 3 tipos de píxeles: **fuertes, débiles y no-relevantes**.
- Se usan para ello 2 umbrales para el máximo y mínimo.

⑤ Seguimiento de Bordes

- En este paso se analizan los bordes clasificados como débiles y se suprimen si estos no tienen ninguna conexión con los bordes clasificados como fuertes.

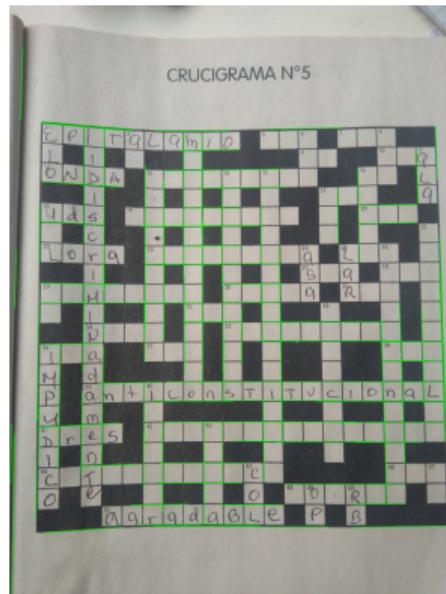
Detector Canny VI

Algoritmos



Algoritmos I

Detección de bordes



Algoritmos II

Detección de bordes

- Existen multitud de métodos de detección de características, pero no tantos de descriptores.
- Tanto los detectores como los descriptores deben ser invariantes a escala, posición, luminosidad y rotación.
- Se suelen usar para encontrar correspondencias entre imágenes (estéreo, flujo óptico, reconocimiento de objetos).



Sección 3.3

Detección de bordes

Transformada de Hough

Transformada de Hough I

Detección de bordes

- La transformación Hough es una técnica de extracción de características utilizada en el análisis de imágenes, la visión por ordenador y el procesamiento de imágenes digitales.
- Fue presentada y patentada en 1962 por Paul Hough
- En sus inicio era aplicada solo para detectar rectas posteriormente se extendió su uso con otras figuras que se pudieran parametrizar con pocos datos.
- La versión actual fue inventada por Richard Duda y Peter Hart en 1972 y popularizada por Dana H. Ballard en su artículo de 1981.



Transformada de Hough II

Detección de bordes

- La transformada de Hough permite encontrar primitivas geométricas dentro de una imagen.
- Recibe una ecuación donde se parametrizan las primitivas que deseamos encontrar.
- La imagen donde se realiza la búsqueda se debe pre-procesar para extraer sus bordes utilizando algún operador.



Transformada de Hough III

Detección de bordes

- En el análisis automatizado de imágenes digitales, a menudo surge un subproblema de detección de formas simples, como líneas rectas, círculos o elipses.
- Debido a imperfecciones de los datos de la imagen o del detector de bordes, pueden faltar puntos o píxeles en las curvas deseadas, así como desviaciones espaciales entre la línea/círculo/elipse ideal y los puntos de borde ruidosos, tal como se obtienen del detector de bordes.



Transformada de Hough IV

Detección de bordes

- Por estas razones, a menudo no es trivial agrupar las características de los bordes extraídos en un conjunto apropiado de líneas, círculos o elipses.
- El propósito de la transformada de Hough es resolver este problema haciendo posible la agrupación de puntos de borde en objetos candidatos mediante la realización de un procedimiento de votación explícito sobre un conjunto de objetos de imagen parametrizados.



Transformada de Hough V

Detección de bordes

- El caso más simple para la transformada de Hough es la transformación lineal para detectar líneas rectas.
- En el espacio de la imagen, la recta se puede representar con la ecuación $y = mx + n$ y se puede graficar para cada par (x, y) de la imagen.
- La idea principal es considerar las características de una recta en término de sus parámetros (m, n) y no como puntos de la imagen $(x_1, y_1), \dots, (x_n, y_n)$.



Transformada de Hough VI

Detección de bordes

- Basándose en lo anterior, la recta $y = mx + n$ se puede representar como un punto (m, n) en el espacio de parámetros.
- Sin embargo, cuando se tienen rectas verticales, los parámetros de la recta (m, n) son indefinidos.
- Por esta razón es mejor usar los parámetros que describen una recta en coordenada polares, denotados (θ, ρ) .



Transformada de Hough VII

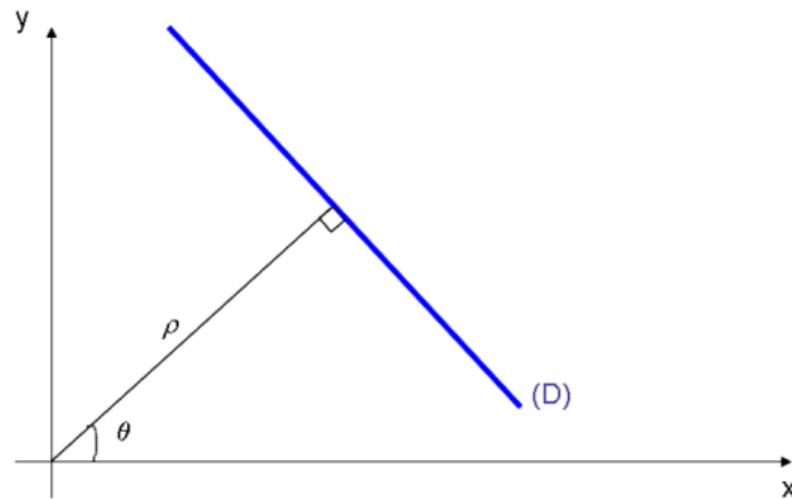
Detección de bordes

- El parámetro ρ representa la distancia entre el origen de coordenadas y el punto (x, y) mientras que θ es el ángulo del vector director de la recta perpendicular a la recta original y que pasa por el origen de coordenadas.



Transformada de Hough VIII

Detección de bordes



Ecuación de una línea (D): $\rho = x \cos \theta + y \sin \theta$



Transformada de Hough IX

Detección de bordes

La transformación Hough tiene como objetivo estimar las líneas a partir de un conjunto de punto $S = (x_i, y_i)$ Para $i = 1, \dots, n$.

- Crear una matriz bidimensional para la variable (θ, ρ) con ceros de entrada.
- Para cada observación (x_i, y_i) , incrementan todas las celdas de la matriz de tal manera que:

$$\rho = x_i \cos \theta + y_i \sin \theta$$

Esta matriz es entonces una estimación bidimensional del histograma de $p(\theta, \rho)$.

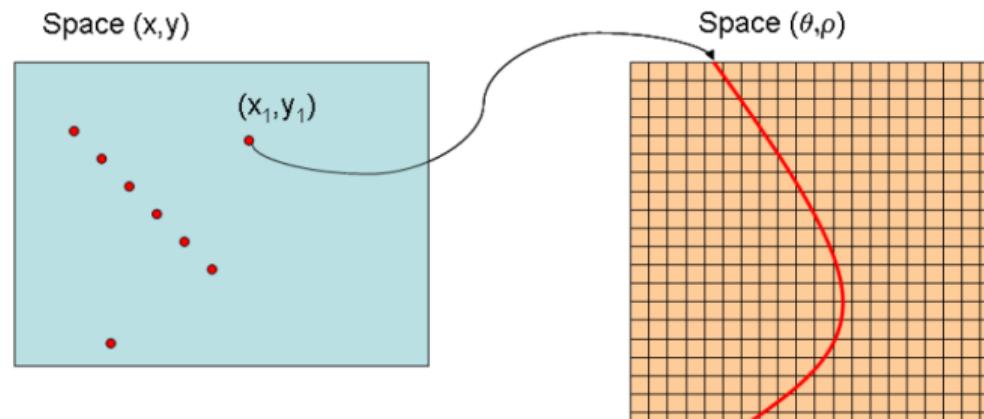
- Detectar los máximos en el $p(\theta, \rho)$.



Transformada de Hough X

Detección de bordes

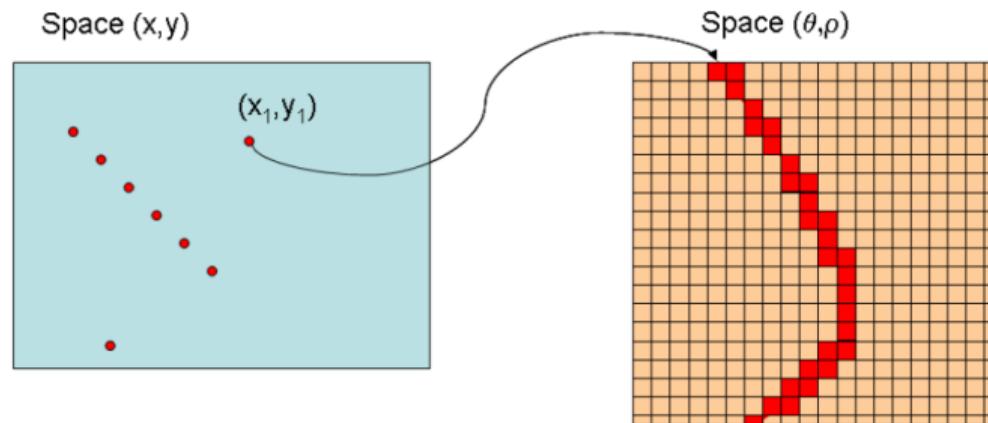
Ejemplo



Transformada de Hough XI

Detección de bordes

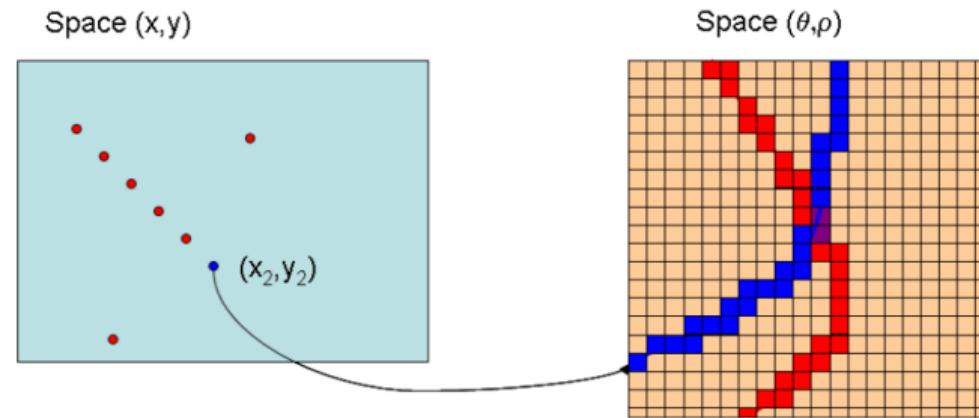
Ejemplo



Transformada de Hough XII

Detección de bordes

Ejemplo



Transformada de Hough XIII

Detección de bordes

Algorithm 1: Algoritmo de detección de Lineas Rectas con Transformada de Hough

input : Imagen *Img*

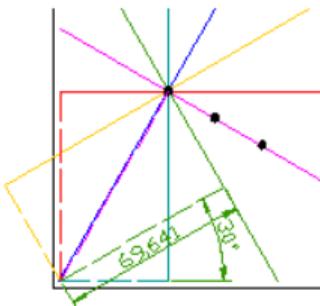
output: Parámetros de Recta (ρ, θ)

```
1 Cargar imagen Img
2 Crear una matriz 2D acum[ $\rho, \theta$ ] llena de 0
3 // Detectar Bordes en la Imagen
4 ImgBor = detectarBordes(I)
5 forall (x, y) en Img do
6   if (x, y) en ImgBor then
7     foreach  $\theta$  Posible do
8       Calcular  $\rho$  de (x, y) usando  $\theta$ 
9       // Incrementar la posición en el acumulador
10      acum[ $\rho, \theta$ ] += 1
11    end foreach
12  end if
13 end forall
14 Buscar las posiciones en acum con mayores valores(más votos)
15 Devolver los parámetros de recta( $\rho, \theta$ ) de los mayores valores
```

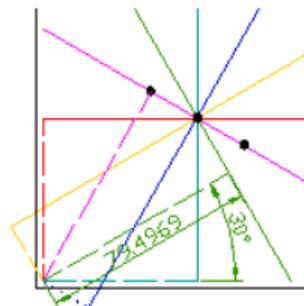
Transformada de Hough XIV

Detección de bordes

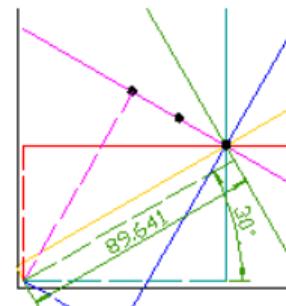
Consideremos tres puntos en una imagen, mostrados aquí como puntos negros:



Angle	Dist.
0	40
30	69.6
60	81.2
90	70
120	40.6
150	0.4



Angle	Dist.
0	57.1
30	79.5
60	80.5
90	60
120	23.4
150	-19.5



Angle	Dist.
0	74.6
30	89.6
60	80.6
90	50
120	6.0
150	-39.6



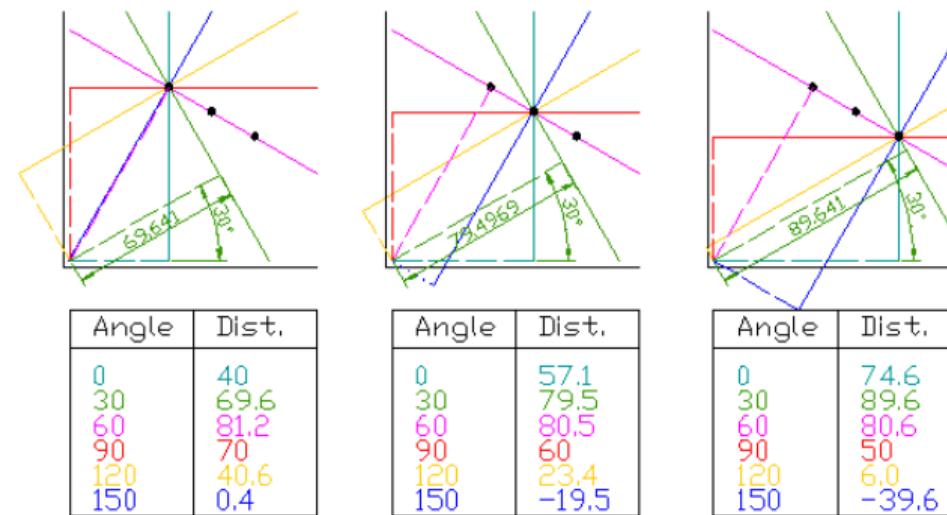
Transformada de Hough XV

Detección de bordes

Por cada **punto** se dibujan un número de líneas que pasan por los mismos, con distintos ángulos. Son las líneas continuas en la figura.

Transformada de Hough XVI

Detección de bordes

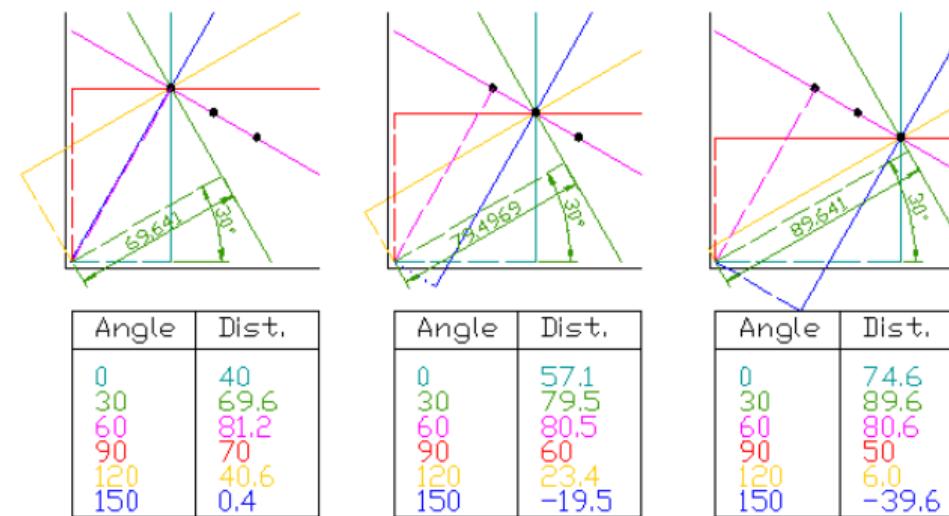


Por cada **línea** se dibuja una recta perpendicular a esta que pasa por el origen de coordenadas. Son las líneas discontinuas de la figura.



Transformada de Hough XVII

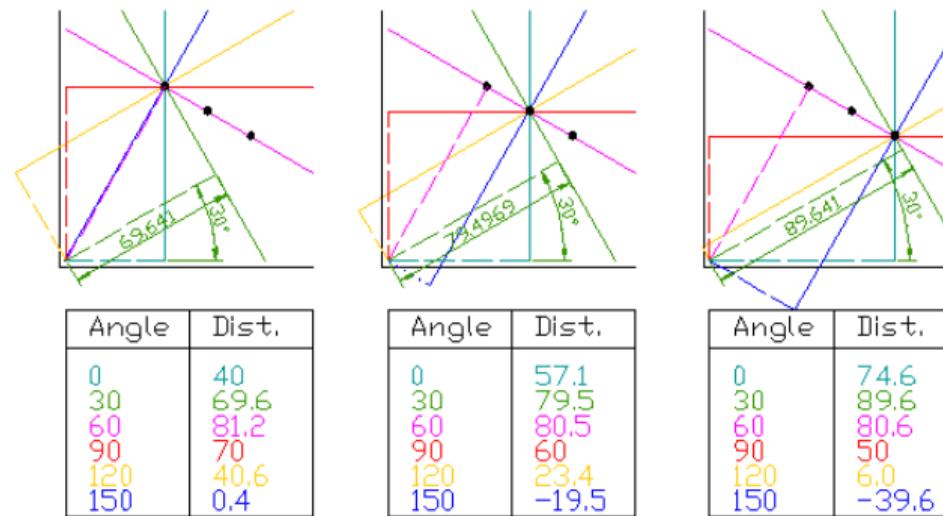
Detección de bordes



La **longitud** (ρ) y el **ángulo** (θ) de cada línea discontinua se calcula. Los resultados se muestran en las tablas.

Transformada de Hough XVIII

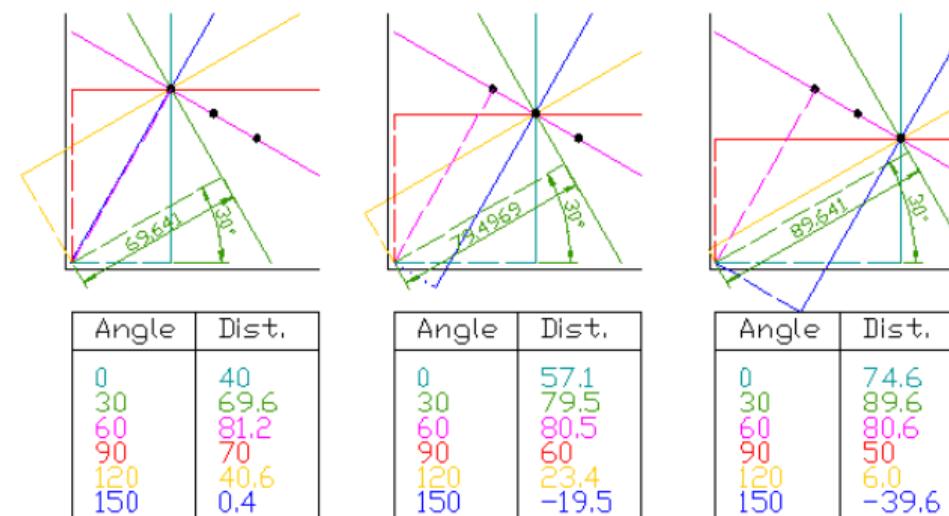
Detección de bordes



Esta operación se repite para cada punto dentro de la imagen.

Transformada de Hough XIX

Detección de bordes

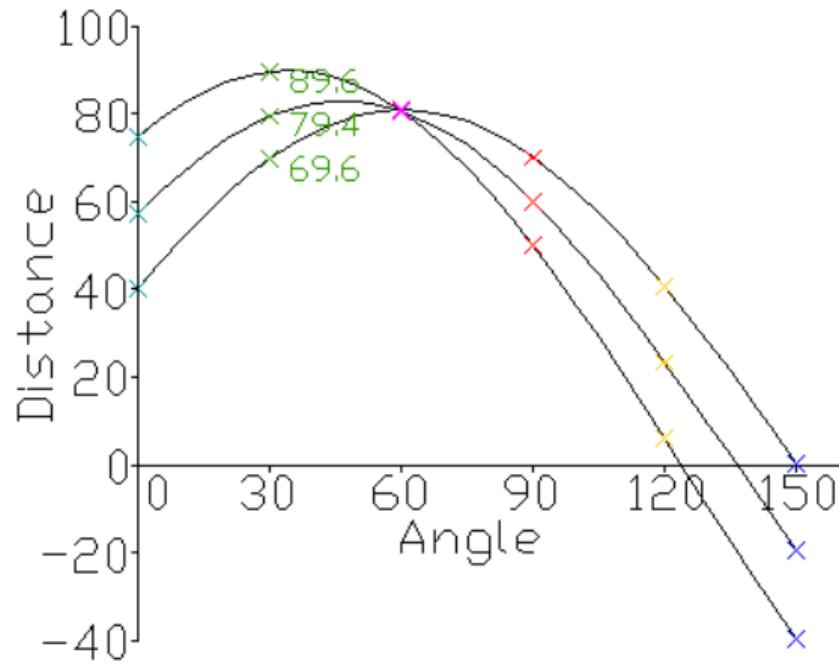


Se crea una gráfica con las longitudes de las líneas por cada ángulo, conocido como **grafo del espacio de Hough**.



Transformada de Hough XX

Detección de bordes



Transformada de Hough XXI

Detección de bordes

El punto donde las curvas se **intersecan** da la **distancia**(ρ) y el **ángulo**(θ). Este par (ρ, θ) indica la recta que contiene los puntos anteriores.

Limitaciones I

Detección de bordes

- La transformada de Hough sólo es eficiente si un alto número de votos caen en la celda correcta, de modo que la celda pueda ser fácilmente detectada en medio del ruido de fondo.
- Esto significa que la celda no debe ser demasiado pequeña, o de lo contrario algunos votos caerán en las celdas vecinas, reduciendo así la visibilidad de la celda principal.

Limitaciones II

Detección de bordes

- Además, cuando el número de parámetros es grande, es decir, cuando usamos la transformada Hough con típicamente más de tres parámetros, el número promedio de votos emitidos en una sola casilla es muy bajo.
- Esas casillas que corresponden a una figura real en la imagen no parecen tener necesariamente un número mucho mayor de votos que sus vecinos.



Limitaciones III

Detección de bordes

- Por último, gran parte de la eficiencia de la transformación Hough depende de la calidad de los datos de entrada.
- Una buena detección de los bordes en la imagen es esencial para asegurar que la transformación Hough sea eficiente.



Introducción a la Visión Artificial con ROS

4

Reconocimiento Facial

- Concepto
- Enfoques

Reconocimiento Facial

Sección 4.1 | Concepto

Concepto I

Reconocimiento Facial

- Esta sección se enfoca en presentar diversos métodos existentes que se utilizan para realizar operaciones sobre el rostro humano.
- Estos métodos y/o librerías son de amplio uso en la actualidad y se emplean en diversos tipos de situaciones que involucran desde el ocio hasta la seguridad.

Concepto II

Reconocimiento Facial

- Cuando se habla de reconocimiento facial, se debe ver a manera general como el área de la visión artificial que cuyas operaciones están optimizadas para el procesamiento de rostros con diferentes enfoques, bajo diversos objetivos y que actúan a varios niveles.
- Los enfoques de reconocimiento facial se pueden considerar especializaciones de varios de los métodos de detección y reconocimiento existentes en la visión artificial.



Reconocimiento Facial

Sección 4.2 | Enfoques

Enfoques I

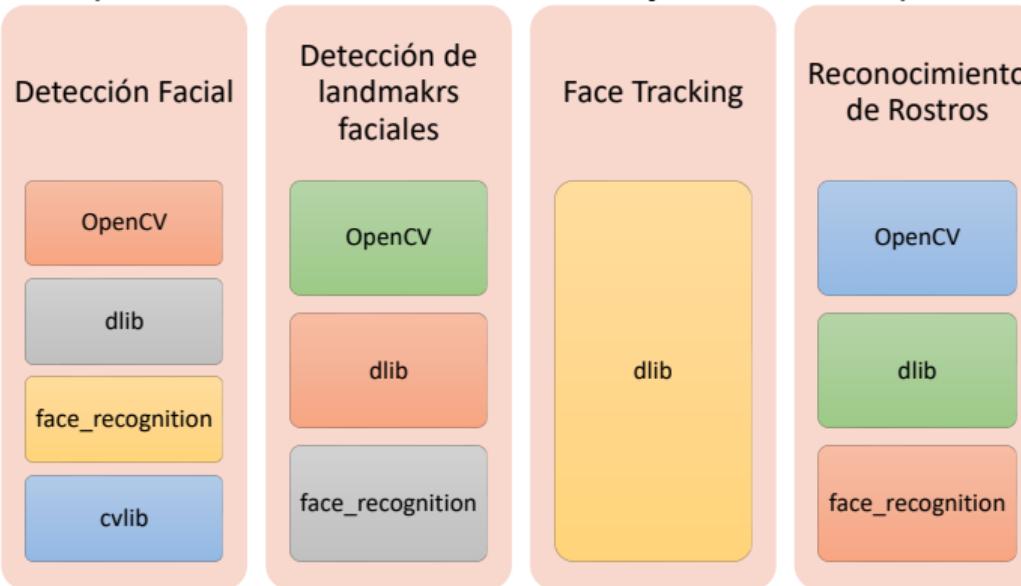
Reconocimiento Facial

- El reconocimiento facial se puede subdividir en algunas áreas o enfoques, las cuales presentan cierto grado de similitud en sus procedimiento o métodos.
- Estos enfoques persiguen cada uno sus objetivos y aplican técnicas de visión artificial para cumplirlos.
- Dentro de estos se pueden mencionar los más conocidos que son
 - ① *Face detection*
 - ② *Facial landmarks detection*
 - ③ *Face Tracking*
 - ④ *Face Recognition*

Enfoques II

Reconocimiento Facial

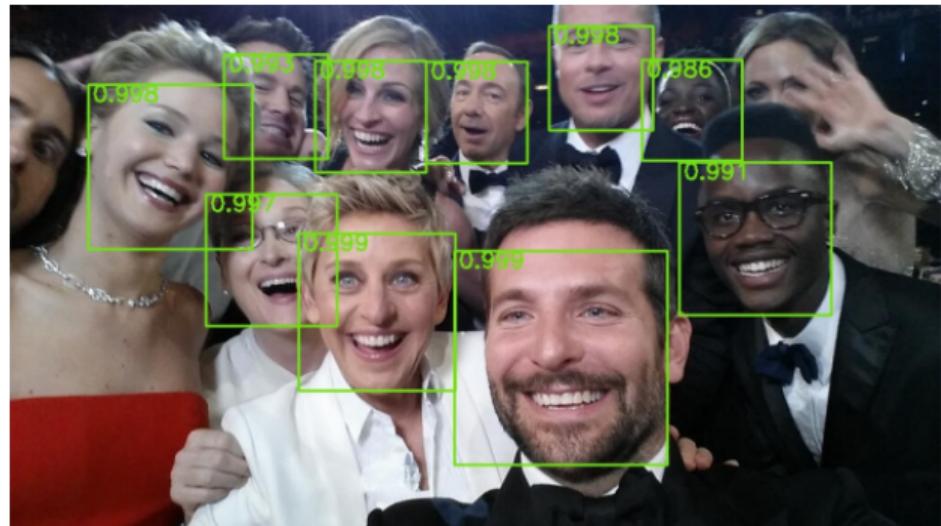
Enfoques de Reconocimiento facial y módulos disponibles



Face detection

Enfoques

- Detección facial se puede considerar un caso específico de detección de objetos.
- En este caso su objetivo es encontrar la ubicación y tamaño de todas las caras en una imagen.



Face detection - Algoritmos

Enfoques

- OpenCV
 - Haar Cascade Clasifiers
 - Single Shot Detector SSD (DNN)
- DLIB
 - HOG Descriptor usando Support Vectors Machine (SVM)
- FaceRecognition
 - HOG + Convolutional Neural Network (CNN)
- CVLIB
 - Caffe Models (DNN)



Facial landmarks detection

Enfoques

- Consiste en un caso especial de detección de puntos de referencia
- En este caso su objetivo es encontrar la mayor cantidad de puntos dentro de un rostro.



Facial landmarks detection - Algoritmos

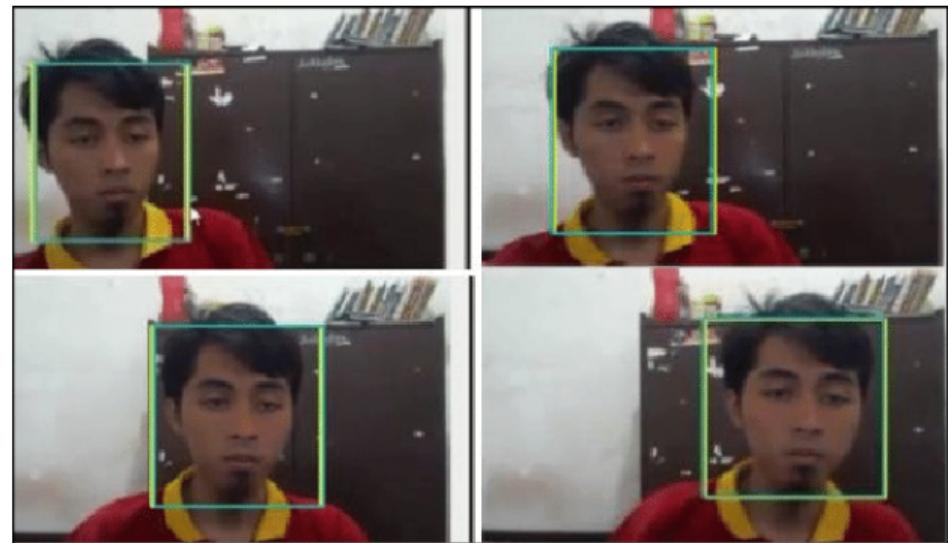
Enfoques

- OpenCV
 - LBF
 - Kamezi
 - AAM
 - Se requiere un detector de rostros antes (Haar)
- DLIB
 - ShapePredictor
- FaceRecognition
 - Detector de 5 y 68 landmarks

Face Tracking

Enfoques

- Consiste en un caso especial de seguimiento de objetos
- En este caso su objetivo es detectar una cara y seguir su movimiento en *frames* de vídeo sucesivos.



Face Tracking - Algoritmos

Enfoques

- DLIB

- *Correlation Tracker*
- Puede ser aplicado también al seguimiento de objetos



Face Recognition

Enfoques

- Consiste en un caso especial de reconocimiento de objetos
- Su enfoque busca identificar y verificar a una persona según su información facial.
- Se puede separar en dos fases:
 - Identificación
 - Verificación



Face Recognition - Algoritmos

Enfoques

- OpenCV
 - Local Binary Patterns (LBP)
 - EigenFaces
 - FisherFaces
 - Estos se basan en aprendizaje automático
- DLIB
 - ResNet-34 entrenada con 3 millones de caras
- FaceRecognition
 - Enfoque basado en DLIB, cambiando esquemas de comparación al final



Introducción a la Visión Artificial con ROS

5 Conexión ROS con OpenCV

Importaciones de Librería

- Para trabajar con imágenes en ROS se utiliza el CV Bridge, el cual es una enlace que permite obtener imágenes mediante tópicos de ROS y utilizarlos para diferentes procesos.
- Para su uso se debe importar lo siguiente en nuestro nodo.

```
from sensor_msgs.msg import Image
from cv_bridge import CvBridge, CvBridgeError
```

Tópico de la Cámara

- Para la conversión de la información del tópico a un tipo de dato de OpenCV se emplean las siguientes líneas de código.
- En estas nuestro subscriber se configura para recibir mensajes del tipo *Image*.
- Posteriormente el mensaje recibido es convertido mediante la instancia de *CvBridge()*, lo cual permite su procesamiento usando OpenCV.

```
self.image_pub = rospy.Publisher("image_topic_2",Image, queue_size=1)

self.bridge = CvBridge()
self.image_sub = rospy.Subscriber("/cv_camera/image_raw",Image,self.callback)
cv_image = self.bridge.imgmsg_to_cv2(data, "bgr8")
```

Recursos Bibliográficos I

RAS-IEEE

-  [Pattern recognition.](https://www.journals.elsevier.com/pattern-recognition)
URL <https://www.journals.elsevier.com/pattern-recognition>.
-  [OpenCV: Feature Matching with FLANN, a.](https://docs.opencv.org/master/d5/d6f/tutorial_feature_flann_matcher.html)
URL https://docs.opencv.org/master/d5/d6f/tutorial_feature_flann_matcher.html.
-  [OpenCV: 2D Features framework \(feature2d module\), b.](https://docs.opencv.org/master/d9/d97/tutorial_table_of_content_features2d.html)
URL https://docs.opencv.org/master/d9/d97/tutorial_table_of_content_features2d.html.
-  [OpenCV: Sobel Derivatives, c.](https://docs.opencv.org/3.4/d2/d2c/tutorial_sobel_derivatives.html)
URL https://docs.opencv.org/3.4/d2/d2c/tutorial_sobel_derivatives.html.
-  [Christopher M Bishop.](#)
Pattern recognition and machine learning.
springer, 2006.



Recursos Bibliográficos II

RAS-IEEE

 Izhar Haq, Shahzad Anwar, Kamran Shah, Muhammad Tahir Khan, and Shaukat Ali Shah.

Fuzzy logic based edge detection in smooth and noisy clinical images.

PLOS ONE, 10(9):1–17, 09 2015.

doi: 10.1371/journal.pone.0138712.

URL <https://doi.org/10.1371/journal.pone.0138712>.

 MATLAB.

Edge detection.

URL <https://www.mathworks.com/discovery/edge-detection.html>.

 P P Pankaj.

Applications of pattern recognition, Dec 2019.

URL <https://www.geeksforgeeks.org/applications-of-pattern-recognition/>.

Recursos Bibliográficos III

RAS-IEEE



Sofiane Sahir.

Canny Edge Detection Step by Step in Python — Computer Vision, January 2019.

URL <https://towardsdatascience.com/canny-edge-detection-step-by-step-in-python-computer-vision-b49c3a2d8123>.



Scott E Umbaugh.

Digital image processing and analysis: human and computer vision applications with CVIPtools.

CRC press, 2010.



Junqing Wang and Weichuan Zhang.

A survey of corner detection methods.

In *2018 2nd International Conference on Electrical Engineering and Automation (ICEEA 2018)*. Atlantis Press, 2018.

