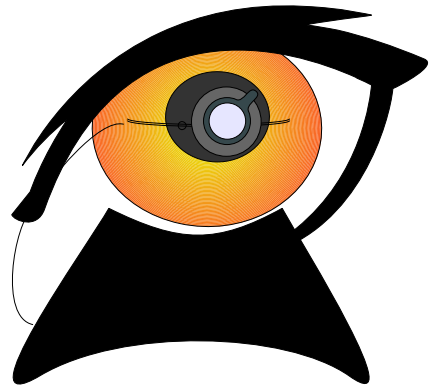
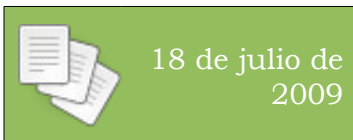


# ***Visión artificial***



## **Introducción a los sistemas de visión artificial**



Copyright © 2008, 2009 Jorge Rodríguez Araújo

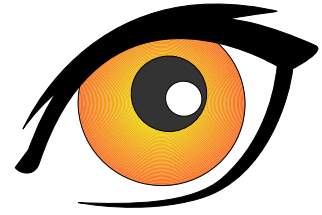
[grodri@gmail.com](mailto:grodri@gmail.com)



# Índice

1. Visión artificial.....	1
1.1 <i>Sistema de iluminación</i> .....	1
1.1.1 Fuentes de luz.....	1
1.1.2 Esquemas de iluminación.....	2
1.2 <i>Captura de imagen</i> .....	3
1.2.1 Sensores digitales de imagen.....	4
Sensor CCD.....	4
Sensor CMOS.....	4
Características.....	4
1.2.2 Objetivos.....	5
Características.....	6
Selección de la lente.....	6
1.3 <i>Procesado de imagen</i> .....	6
1.3.1 Modelos de color.....	7
1.3.2 Tratamiento de imagen.....	7
Transformaciones de histograma.....	7
Eliminación de ruido.....	8
1.3.3 Segmentación.....	8
Operadores morfológicos.....	9
1.3.4 Etiquetado y clasificación.....	10
1.3.5 Visión estéreo.....	10
1.3.6 Template matching.....	11
2. Procesamiento de imagen con Python.....	13
2.1 <i>PIL</i> .....	13
2.1.1 Negativo.....	14
2.1.2 Representación a lo Matlab.....	15
2.2 <i>OpenCV</i> .....	16
2.2.1 Capturando desde una webcam.....	16
3. VirtualSketch.....	18
3.1 <i>Introducción</i> .....	18
3.1.1 Objetivos.....	18
3.1.2 Método.....	18
3.2 <i>Identificación del marcador</i> .....	19
3.2.1 Algoritmo meanshift.....	19
Captura del fondo.....	20
Detección de movimiento.....	20
Segmentación.....	20
Identificación.....	21
Localización.....	21
3.3 <i>Implementación</i> .....	23
Capturas de pantalla.....	23

4. Filtro de Kalman.....	24
4.1 <i>Introducción</i> .....	24
4.2 <i>Filtro de Kalman</i> .....	24
4.3 <i>Aplicación del filtro de Kalman</i> .....	26
4.3.1 <i>Estimación de la posición de un teléfono móvil</i> .....	26
4.3.2 <i>Desarrollo</i> .....	26
5. Identificación de un contorno rectangular.....	28
5.1 <i>Introducción</i> .....	28
5.2 <i>Detección de bordes</i> .....	28
5.3 <i>Detección del contorno rectangular</i> .....	29
Tranformada de Hough para líneas.....	29
5.4 <i>Identificación del contorno rectangular</i> .....	31
6. Cálculo de la homografía.....	32
6.1 <i>Homografía</i> .....	32
6.2 <i>Resolución de sistemas de ecuaciones lineales</i> .....	33



# 1. Visión artificial

Un sistema de visión artificial se compone básicamente por:

Hardware	<ul style="list-style-type: none"> <li>• <b>Sistema de iluminación:</b> Proporciona unas condiciones de iluminación uniformes e independientes del entorno, permitiendo o facilitando la extracción de los rasgos de interés para una determinada aplicación.</li> <li>• <b>Captura de imagen:</b> Conjunto encargado de recoger las características del objeto en estudio y proporcionar los datos para su procesamiento, por medio de una imagen digital.</li> </ul>
	<ul style="list-style-type: none"> <li>• <b>Procesado de imagen:</b> Conjunto de algoritmos y transformaciones necesarias para realizar el análisis de la imagen y de este modo extraer la información de las imágenes capturadas, con el fin de obtener los resultados para los que haya sido diseñado.</li> </ul>
Software	

## 1.1 Sistema de iluminación

Dado que la imagen se forma a partir de la luz que reflejan los objetos, la única posibilidad de obtener una imagen con características constantes en la que se resalten los aspectos de interés y se atenúen los intrascendentes permitiendo o facilitando la resolución del problema es controlando las condiciones de iluminación.

Para ello, se emplea el sistema de iluminación, que se encuentra formado por la fuente de luz según un determinado esquema de iluminación, lo que en conjunto proporciona unas condiciones de iluminación determinadas.

Hay que tener en cuenta que el problema de iluminación no resulta trivial y su coste supone una parte importante del total del proyecto.

Dentro de la reflexión de la luz se distingue entre reflexión especular, donde el rayo incidente es reflejado en una única dirección, y reflexión difusa, donde los rayos incidentes son dispersados en múltiples direcciones.

En la reflexión directa el ángulo de incidencia es igual al ángulo de reflexión:

$$\theta_i = \theta_r$$



### 1.1.1 Fuentes de luz

Entre la multitud de fuentes de luz existentes, las más utilizadas son:

- **Fluorescente** de alta frecuencia<sup>1</sup>: no ofrece demasiada luz y presenta deriva con el tiempo, pero su precio económico y su adaptabilidad tanto en forma como en color los hacen atractivos.
- **Halógena**: presenta gran luminosidad, pero desprende calor, su luz es caliente y su precio es caro. Además, presentan envejecimiento.
- **Xenón**: presenta aún mayor luminosidad, aunque los mismos inconvenientes, y un precio aún mayor.
- **LED**: admite multitud de configuraciones y están disponibles en multitud de colores, son estables, duraderos, funcionan en baja tensión, aunque su precio es elevado.
- **LÁSER**: permiten generar patrones de iluminación.

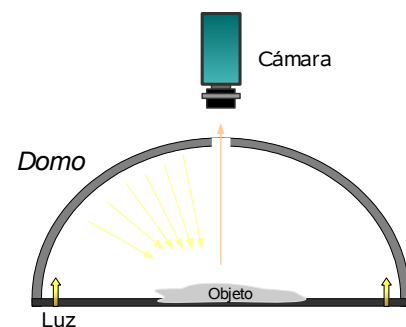
### 1.1.2 Esquemas de iluminación

Los principales esquemas de iluminación son:

- **Iluminación posterior difusa** (Backlight): mediante la iluminación difusa por la parte posterior del objeto se obtiene una imagen de alto contraste, donde la silueta oscura de los objetos se ve resaltada frente al fondo blanco, y por tanto resulta adecuado para medir la forma de los objetos.

- **Iluminación difusa direccional**: emulando a la luz natural pero sin variaciones se emiten los rayos paralelos y en una dirección determinada para lograr una iluminación uniforme.

- **Iluminación omni-direccional difusa**: consigue una imagen libre de sombras y con gran contraste, gracias a que el objeto es iluminado desde todas direcciones con luz difusa. Para ello se emplea una superficie reflectante de forma semiesférica llamada domo, que actúa como fuente de iluminación al reflejar la luz, con lo que se logra eliminar las sombras y reflejos, y se aumenta el contraste, suavizando las texturas y minimizando la influencia de las rayas, el polvo y los relieves, así como de las curvaturas que pueda tener el objeto inspeccionado.



<sup>1</sup> Deben emplearse fluorescentes de alta frecuencia dado que los estándar producen un efecto parpadeo en la imagen que la hace inservible.

Se emplea en la detección de marcas de diferentes colores, caracteres y detección de todo lo que suponga un cambio de color tanto en superficies lisas, esféricas, rugosas o con brillo.

- **Iluminación direccional lateral** (Grazing): resalta la textura de los objetos o aquellas características salientes, dado que al incidir la luz con un ángulo muy pequeño sólo será reflejada hacia la cámara cuando se encuentre algún saliente. Por tanto, resulta útil para resaltar protuberancias y hendiduras de la superficie del objeto.

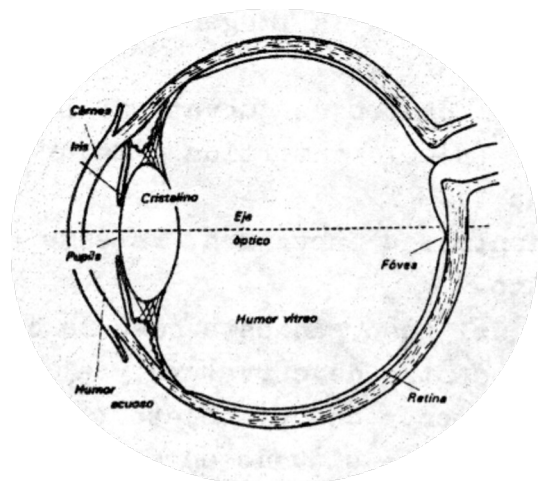
- **Iluminación con luz polarizada:** puede servir para eliminar brillos de objetos brillantes, como en el caso de una bolsa de patatas, dado que los filtros que se instalan en iluminación y cámara impedirán que la luz que no provenga de la fuente de iluminación, y que por tanto presentará distinta polarización, sea capturada por la cámara.

- **Iluminación estructurada:** mediante la proyección de puntos, franjas o rejillas de luz sobre la superficie de trabajo se establece un patrón que permite extraer las características dimensionales del objeto por medio de la medición de la distorsión que sufre dicho patrón de luz ante la presencia de un objeto. Uno de los sistemas más conocidos consiste en la iluminación con un haz láser plano, de modo que al situarse en un ángulo conocido respecto a la cámara, la línea proyectada y distorsionada por la presencia de un objeto presentará un determinado desplazamiento según la profundidad a la que se encuentre el punto, permitiendo de este modo conocer su posición en el espacio, y finalmente reconstruir el objeto 3D mediante técnicas de computación.

- **Montajes con espejos:** los montajes realizados con espejos permiten ver distintas partes de un objeto en una misma imagen.

## 1.2 Captura de imagen

En la captura de imagen intervienen las cámaras y las tarjetas de adquisición de imagen, estando las cámaras, al igual que el ojo humano, formadas por una lente convergente que proyecta la imagen sobre una superficie sensible a la luz denominada sensor de imagen.



### 1.2.1 Sensores digitales de imagen

Los sensores digitales de imagen están formados por una serie de elementos fotosensores (sensibles a la luz) que modifican su señal eléctrica según la intensidad luminosa que reciben, lo que permite la captura de los puntos que conforman la imagen.

Estos sensores suelen estar configurados en forma matricial de modo que proporcionan una imagen bidimensional.

A nivel industrial también se emplean los sensores con configuración en línea, dada sus altas resoluciones, cuando los objetos se sitúan en superficies en movimiento para aplicaciones de medida (metrológica).

Con objeto de reducir costes suelen existir en el mercado multitud de cámaras que emplean lo que se conoce como imagen entrelazada, donde se capturan por un lado las líneas pares y por otro las impares, con lo que al hacerse en instantes de tiempo diferentes no pueden ser empleadas con objetos en movimiento.

Los sensores son realizados en dos tecnologías, principalmente en CCD y en CMOS.

#### *Sensor CCD*

La tecnología CCD (Charge Coupling Device) es la que proporciona una mejor imagen, sobre todo en el caso de las cámaras profesionales en las que se emplea un sensor para cada color, mientras que en las de consumo se emplea un único sensor, de modo que este está constituido como un mosaico de detectores de colores y por tanto captura una menor densidad de puntos. Cuando el sensor ha capturado la imagen envía la información completa serializada fila a fila.

#### *Sensor CMOS*

La tecnología CMOS es la más económica, y ofrece un menor tamaño, gracias a su grado de integración electrónica. Además, puede capturar partes de la imagen, no teniendo que transmitirla completamente.

#### *Características*

A la hora de seleccionar una cámara hay que tener en cuenta una serie de características, y entre las principales se destacan:

- **Resolución:** número de píxeles que conforman la imagen capturada.
- **Sensibilidad:** nivel mínimo de iluminación que puede capturar el sensor.

- **Rango dinámico:** margen de luz (intensidad luminosa) que puede capturar el sensor, o sea, diferencia máxima entre luz y oscuridad que puede distinguir.
- **Señal/ruido** y **Smear:** influencia entre píxeles.
- **Velocidad:** velocidad máxima a la que puede capturar imágenes, siendo su medida en frames por segundo [fps].

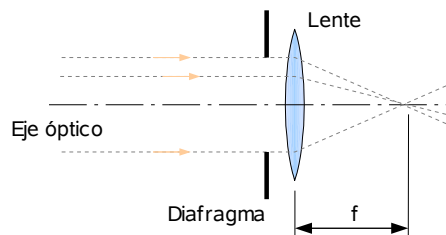
### 1.2.2 Objetivos

El objetivo tiene como función concentrar los rayos de luz, provenientes de los objetos a visualizar, en un plano donde se formará la imagen para ser captada por el sensor.

Está formado por el **diafragma**, que controla la apertura y por tanto limita el haz de rayos entrante, la **lente**, que enfoca el objeto, y el **zoom**, que permite variar la distancia focal (distancia al objeto).

El diafragma permite regular la apertura del objetivo y con ello la luminosidad de la imagen. Además, aunque al cerrar el diafragma los rayos salen enfocados independientemente de la distancia al objeto, entrará menos luz y por tanto la escena será más oscura, con lo que se requerirá un mayor tiempo de exposición.

La lente viene caracterizada por la distancia focal ( $f$ ), que es la distancia desde su centro hasta el punto en el que convergen los rayos que llegan paralelos a ella. Así, el aumento de la distancia focal disminuye el campo visible y con ello la distancia de trabajo de la misma.



De este modo, el desenfoque se produce según el objeto se aleja o acerca de la posición ideal, dado que la imagen dejará de formarse sobre el sensor, y por tanto cada punto de la imagen no se corresponderá con su elemento correspondiente del sensor. Para corregirlo, se cambia la focal de la lente, lo que se traduce en variar la distancia entre lente y sensor.

Existen una clase de objetivos, llamados **telecéntricos**, que permiten ver los objetos en proyección ortográfica; siempre y cuando sean más pequeños que el diámetro de la propia lente.



### Características

La dos características principales que definen un objetivo son:

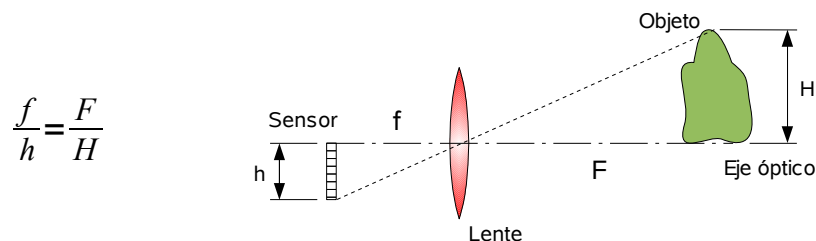
- **Campo visible** (FOV): es el área visible por la lente para una cierta distancia de trabajo en la que los objetivos aparecen enfocados.
- **Profundidad de campo** (DOF): es el rango de distancias a las cuales los objetos se encuentran enfocados.

### Selección de la lente

Como la lente está muy cerca del sensor frente a la distancia al objeto, se aproxima la distancia focal ( $f$ ) por aquella que exista entre la lente y el sensor, de modo que los rayos pasarán por el centro de la lente.

Así, si se desea que un objeto de 150x150 mm sea recogido por una cámara con un sensor CCD 1/3" (3,6x4,8 mm) que se coloca a 400 mm del objeto, la distancia focal máxima que permitiría que el objeto entrase en la imagen capturada por el sensor, vendría dada por:

**Lente de Fresnel:** está compuesta por microcurvaturas que le permiten, en teoría, ser igual de funcional que una lente de alta curvatura. En la práctica se utilizan, principalmente, para conseguir una fuente de luz colimada (rayos paralelos) a bajo coste, dado su ahorro de material.



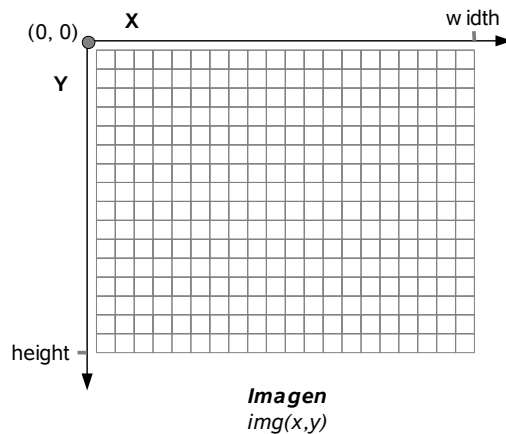
De tal modo que para que el objeto quepa en la imagen, la distancia focal máxima vendrá dada por la altura menor del sensor ( $h=3,6$  mm), siendo  $H=150$  mm la altura del objeto y  $F=400$  mm la distancia.

Distancia focal:  $f = (3,6 \cdot 400) / 150 = 9,6$  mm

## 1.3 Procesado de imagen

Una imagen digital no es más que un arreglo matricial donde el valor de cada elemento ( $img(x,y)$ ), cuyas coordenadas en el plano son  $(x,y)$ , representa el nivel de intensidad de iluminación de cada píxel (punto) que la compone.

## Visión artificial



Aunque lo dicho se refiere a una imagen en escala de grises, donde cada píxel presenta un valor asignado de entre 0 y 255 (un byte), en las imágenes en color se tiene un conjunto valores, dependiendo del modelo de color, por cada píxel.

### 1.3.1 Modelos de color

Aunque existen multitud de modelos de color según su finalidad, los modelos típicos empleados en procesamiento digital son el RGB y el HSI.

El modelo RGB representa el color de cada píxel por medio de un conjunto de tres valores, donde cada valor representa el nivel de intensidad de cada uno de los colores básicos: rojo, verde y azul.

El modelo HSI representa los valores de tono, saturación e intensidad. El tono representa la fuerza del color, la saturación representa el grado con el que el color se diluye en luz blanca, y la intensidad presenta la característica de ser prácticamente independiente de la iluminación, y se obtiene como la suma de los valores RGB (  $I = R + G + B$  ).

Así, mientras que una imagen RGB presenta los colores que deben ser directamente representados en el monitor, una imagen HSI resulta apropiada para obtener información independientemente de las condiciones de iluminación.

### 1.3.2 Tratamiento de imagen

En el tratamiento de imagen se realiza un procesado a bajo nivel para mejorar las características de la imagen, ya que aunque esto no aumentará la información contenida en ella si puede mejorar su visualización o tratamiento posterior.

#### *Transformaciones de histograma*

El histograma es el gráfico de las frecuencias de aparición de cada nivel de gris, o

de los niveles de cada canal de un modelo de color de la imagen, y facilita información para realizar multitud de operaciones de reasignación de los valores de cada píxel de la imagen.

- **Negativo o vídeo inverso:** cambia valores oscuros por claros y viceversa, de tal modo que  $g(x,y) = 255 - f(x,y)$ .
- **Imagen gamma:** resalta los oscuros y satura los claros, mejorando la percepción en imágenes oscuras, o a la inversa,  $(g(x,y) = f(x,y)^v)$  según el valor del exponente.
- **Pseudocolor:** se asigna un mapa de colores a una imagen en escala de grises, o se reasigna el mapa de colores de una imagen a color (color falso), con lo que se logra apreciar detalles de la imagen que debido a la limitación del ojo humano (sólo detecta unos 20 niveles de gris) pasarían desapercibidos.
- **Estiramiento del histograma:** se emplea para mejorar el contraste de las imágenes, dado que distribuye un intervalo del histograma de la imagen de modo que ocupe todo el rango de valores, de tal modo que  $g(x,y) = 255 \cdot (f(x,y) - a)/(b-a)$ , donde  $[a,b]$  es el intervalo de la imagen que será extendido.
- **Ecualización del histograma:** intenta igualar la frecuencia de aparición de todos los niveles de gris, mejorando el contraste de la imagen, aunque a veces puede provocar la aparición de ruido donde hay una rampa en el histograma acumulado.

#### *Eliminación de ruido*

El ruido tipo sal y pimienta (Salt and Pepper) es un tipo especial de ruido impulsivo, que presenta un punteado característico que lo hace fácilmente identificable y que se elimina mediante el empleo del filtro de mediana. Este filtro presenta la ventaja de no eliminar los bordes como lo haría un filtro de media, aunque resulta computacionalmente costoso debido a que requiere la ordenación de los valores de los puntos de la vecindad para calcular la mediana y así sustituir el valor de cada píxel.

El ruido coherente es aquel que presenta un patrón de comportamiento frecuencial, y se elimina por medio del filtrado en frecuencia. Para esto, se realiza la transformación en frecuencia de la imagen por medio de la aplicación de la transformada discreta de Fourier (DFT), y se le aplica un filtro para eliminar el ruido.

#### 1.3.3 Segmentación

En la segmentación se busca la separación entre objetos y fondo.

El método de segmentación más simple es la umbralización, donde se transforma

la imagen a binaria, asignando a cada píxel uno de dos valores según su valor original supere o no un conocido como threshold (umbral).

La identificación de bordes (edges) o contornos, representa otro método de segmentación, y consiste en la aplicación de alguno de los núcleos de convolución (Soble, Roberts, Laplaciano) diseñado a tal efecto. Con estas operaciones se identifican las regiones en las que los píxeles experimentan fuertes variaciones en sus niveles de intensidad.

También existe la posibilidad de aplicar un algoritmo conocido como filtro de Canny para detectar un contorno óptimo de un píxel de ancho. Este filtro consiste en la determinación del borde más probable bajo las condiciones de que: debe detectar todos los contornos y solamente los contornos, la distancia entre el píxel señalado como contorno y el contorno real debe ser tan pequeña como se pueda, y un único contorno debe ser identificado por un único píxel. De este modo la aplicación del filtro de Canny produce un máximo local donde existe un contorno, eliminando así los efectos del ruido.

#### *Operadores morfológicos*

Suelen emplearse en la mejora de los resultados de la segmentación, los conocidos como operadores morfológicos, dado que se basan en la matemática morfológica, ya que descartan o unen puntos que componen los objetos presentes en una imagen según un criterio de forma.

Para esto, las distintas operaciones morfológicas emplean una estructura (núcleo) que define la forma y tamaño de la superficie mínima que compone el objeto. Y se definen dos operaciones básicas, erosión y dilatación, que darán lugar a otras dos, apertura y cierre.

Con la **erosión** se seleccionan los puntos para los cuales el núcleo se encuentra totalmente incluido en la imagen.

Con la **dilatación** se seleccionan los puntos para los cuales el núcleo toca a algún punto de la imagen.

Con la **apertura** se eliminan objetos pequeños y se suavizan los contornos generados por la mala segmentación, dado que está compuesto por una operación de erosión seguida de una de dilatación.

Con el **cierre** se cierran los huecos y aperturas del tamaño del núcleo que presente el objeto, dado que está compuesto por una dilatación seguida de una erosión.

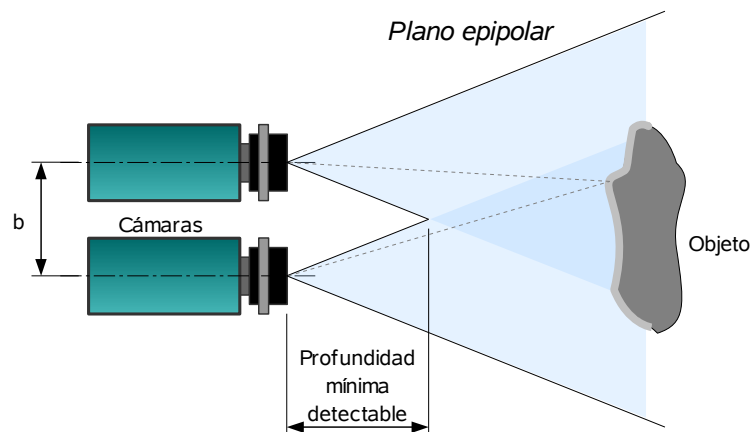
#### 1.3.4 Etiquetado y clasificación

En el etiquetado se busca distinguir entre los distintos objetos encontrados en la etapa anterior. Así, se agrupan los puntos con características similares para formar objetos a los que se les asigna un identificador único (etiquetado).

Posteriormente, se va más allá, y se clasifican en clases, categorías o modelos según sus propiedades.

#### 1.3.5 Visión estéreo

Se conoce como visión estéreo al empleo de dos cámaras para recuperar la información de profundidad de un objeto empleando técnicas de visión artificial.



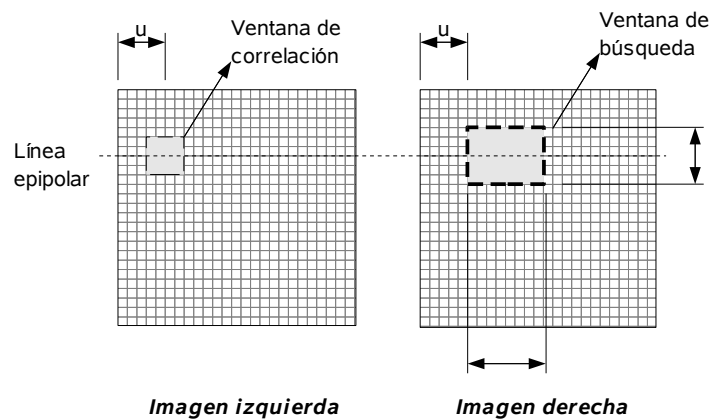
Para simplificar notablemente el problema se emplean dos cámaras iguales, tanto óptica como geoméricamente, que se alinean situándose sus ejes paralelos y coplanarios.

Al cumplirse lo anterior se tiene que se puede aplicar la restricción epipolar, donde los pares conjugados (puntos correspondientes entre las dos imágenes) aparecerán sobre la línea epipolar, resultado de la intersección del plano epipolar con el objeto.

Para obtener la profundidad de cada punto, y así realizar la reconstrucción 3D, habrá que recorrer los puntos de una de las imágenes buscando los correspondientes en la otra (encontrar pares conjugados), para encontrar las disparidades, y con ellas, las profundidades.

Para realizar esta búsqueda-correlación de los puntos de la imagen se emplea el método de máxima correlación (máxima correspondencia), donde se recorta un patrón en torno al punto a buscar de una de las imágenes, al que llamaremos ventana de correlación, y se recorre la otra imagen calculando su correlación, de tal modo que aquel

cuyo valor sea máximo será el más semejante.



Empleando la restricción epipolar, se sabe que los puntos correspondientes se encontrarán en su entorno, y teniendo en cuenta el horopter, se restringe el rango de búsqueda, a la llamada ventana de búsqueda, en torno a la localización del punto (u) en la otra imagen por medio de un intervalo definido por el máximo desplazamiento posible.

Mediante esta comparación se obtiene para cada punto de la imagen izquierda un vector de candidatos, donde se considerará que el par conjugado es aquel que presenta la máxima correlación.

El horopter representa el rango de interés sobre la línea epipolar, dado que para encontrar el punto correspondiente en la otra imagen no será necesario recorrer toda la línea epipolar, ya que el objeto debe presentarse en un determinado rango de disparidad, ya que tanto si se encontrase demasiado cerca como demasiado lejos no se produciría la convergencia de los rayos.

Así, tenemos que la profundidad de un punto vendrá dada por:  $z = \frac{b \cdot f}{d}$

Donde b es la distancia entre cámaras, f la distancia focal, y d la disparidad entre el punto y su correspondiente en la otra imagen.

Aún, con todas estas hipótesis, el coste computacional es elevado y la precisión disminuye con la distancia.

### 1.3.6 Template matching

Se emplea para identificar una parte determinada de la imagen por medio del empleo de una plantilla (template).

Para ello se establece un núcleo formado por el trozo de imagen que contiene la

plantilla ( $g(x, y)$ ) y se busca la región de mayor semejanza dentro de la imagen ( $f(x, y)$ ), lo que vendrá dado por la máxima correlación. Así:

$$R(i, j) = \sum_m \sum_n f(i+m, j+n) \cdot g(m, n) \Rightarrow \text{Mejor correlación} = \max(R(i, j))$$

NOTA: Para que los resultados sean independientes de la imagen, se emplea la correlación normalizada.

## 2. Procesamiento de imagen con Python

El procesamiento de imágenes digitales es el uso de algoritmos de computadora para el tratamiento de imágenes digitales.

- Adquisición de la imagen
- Preprocesamiento
- Segmentación
- Representación y descripción
- Reconocimiento e interpretación

### 2.1 PIL

**PIL** (Python Image Library) es una librería para el manejo de imágenes con Python.

Para manejar las imágenes se utiliza el módulo **Image**, cuyas funciones **open** y **save** permiten cargar y guardar, respectivamente, una imagen en o desde un archivo. La librería, automáticamente, determina a partir de la extensión del nombre de archivo el formato de la imagen.

Cuando se carga o se crea una imagen, lo que realmente se hace es crear un objeto **Image**, a través del cual se da acceso a una serie de métodos y atributos que facilitarán su manipulación y tratamiento.

Así, por ejemplo, el tamaño de la imagen es devuelto por el atributo **size** como un par cuyos valores son el ancho (width) y el alto (height) de la imagen en píxeles. Otro atributo, **mode**, define el modelo de color de la imagen, como “L” (luminance) para escala de grises o “RGB” para color verdadero.

```
#!/usr/bin/env python
# Carga una imagen .jpg, la visualiza y la guarda como .png
import Image

im = Image.open("fruits.jpg")

print im.size, im.mode

im.show()

im.save("fruits.png")
```





### 2.1.1 Negativo

El método **point** proporciona un camino para cambiar los valores de los píxeles de la imagen de una forma directa mediante el empleo de expresiones lambda (funciones definibles en una sola línea), al ser procesado cada píxel de la imagen de acuerdo a la función indicada.

Esta técnica permite aplicar, rápidamente y en una única expresión, todo tipo de transformaciones de punto a una imagen.

```
#!/usr/bin/env python
# Genera el negativo de la imagen
import Image

image = Image.open("fruits.png")
image = image.point(lambda i: 255-i)
image.show()
```



### 2.1.2 Representación a lo Matlab

Mediante la librería `matplotlib` se pueden realizar representaciones gráficas 2D con una interfaz similar a Matlab.

```
#!/usr/bin/env python
# Representa los histogramas de la imagen a color

from pylab import *
import Image

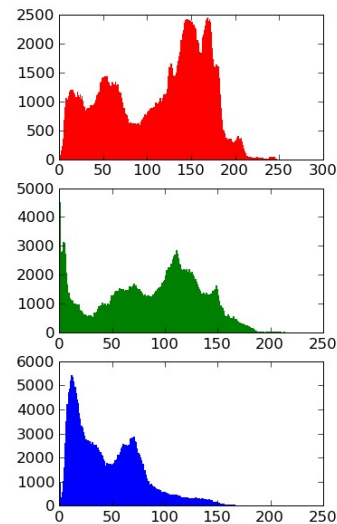
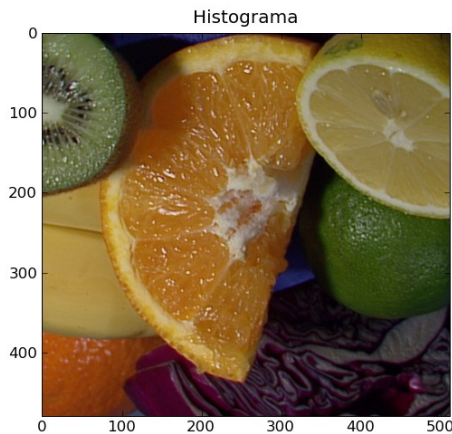
# Carga la imagen
image = Image.open("fruits.png")

# Representa la imagen
subplot(121)
im = imshow(image)

# Representa los histogramas
title("Histograma")
h = image.histogram()
subplot(333)
bar(arange(256), h[0:256], ec="r", fc="r")
subplot(336)
```

```
bar(arange(256), h[256:512], ec="g", fc="g")
subplot(339)
bar(arange(256), h[512:768], ec="b", fc="b")

# Visualiza el resultado
show()
```



## 2.2 OpenCV

**OpenCV** (Open Source Computer Vision Library) es una librería de funciones orientadas al procesamiento de imágenes en tiempo real, desarrollada por Intel y liberada como open source, que permite su uso desde Python tanto en Windows como en Linux.

### 2.2.1 Capturando desde una webcam

A continuación se muestra como capturar y visualizar las imágenes adquiridas a través de una webcam.

```
#!/usr/bin/env python
# Captura las imagenes de una webcam
import sys
from opencv import cv,highgui

if __name__ == '__main__':
    # Inicia el dispositivo de captura
    capture = highgui.cvCreateCameraCapture(0)

    # Crea la ventana de visualizacion
    highgui.cvNamedWindow("Webcam", highgui.CV_WINDOW_AUTOSIZE)

    # Bucle de captura infinito
    while 1:
```

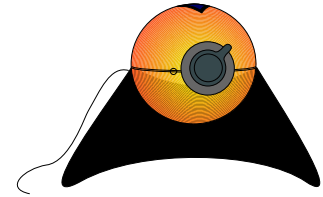
```
# Captura de la imagen actual
frame = highgui.cvQueryFrame(capture)

# Muestra la imagen capturada
highgui.cvShowImage("Webcam", frame)

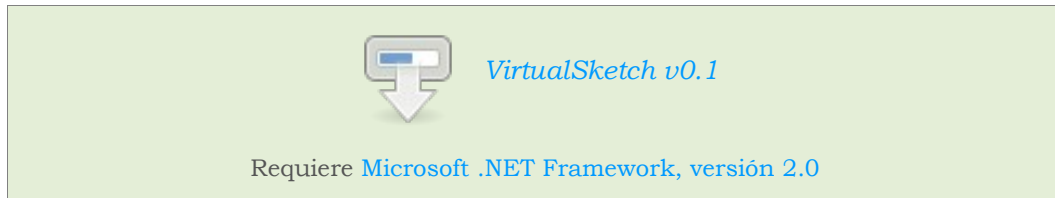
# Captura los eventos de teclado mediante una espera
c = highgui.cvWaitKey(10)

# Detiene la ejecucion al pulsar Esc
if c == "\x1b":
    break

cvDestroyWindow("Webcam")
```



### 3. VirtualSketch



#### 3.1 Introducción

VirtualSketch es una aplicación de escritorio que por medio de visión artificial permite una interacción directa entre hombre y computador a través de una webcam.

De forma natural, y con la única ayuda de una webcam para capturar la imagen, se puede dibujar directamente en el ordenador.

La idea es utilizar cualquier clase de marcador, como el dedo o un bolígrafo para dibujar de forma virtual sobre cualquier tipo de superficie, trazándose el dibujo en el ordenador.

Así, en posteriores versiones se pretende utilizar una simple hoja de papel como panel y mediante el marcador controlar el movimiento del puntero, lo que permitiría simular desde una pantalla táctil hasta una tableta digitalizadora, por supuesto, con las limitaciones de precisión que implica la resolución de la webcam.

##### 3.1.1 Objetivos

Cuando se desarrolla cualquier proyecto resulta fundamental mantener en mente una serie de objetivos que se establecen en su inicio.

En este caso:

- La identificación del marcador debe ser robusta y totalmente automática.
- El seguimiento del marcador debe realizarse con la máxima precisión y estabilidad que permitan los recursos disponibles.
- Los resultados deben ser independientes de la posición y orientación de la cámara.
- El funcionamiento ha de ser simple y transparente al usuario.

##### 3.1.2 Método

Para capturar la posición de un objeto por medio de visión artificial hay que

resolver dos problemas. El primero consiste en identificar y localizar el objeto, y el segundo es establecer la correspondencia entre la posición medida (localización en la imagen) y la real (localización en el mundo real).

Para identificar el objeto se utiliza detección de movimiento, comparando la imagen capturada con una que se toma como referencia (background) y en la que no se puede encontrar el objeto. Con esto se puede identificar el marcador sin su conocimiento a priori e independientemente de las condiciones de iluminación.

Para definir la correspondencia entre la posición medida y la buscada, se empleará en próximas versiones la identificación del recinto rectangular que representa la hoja de papel, y que dado que presenta dimensiones conocidas permite la determinación de la homografía o matriz de proyección, que establece la correspondencia entre las coordenadas de la imagen y las reales.

### 3.2 Identificación del marcador

Para identificar el objeto que servirá como marcador hay que capturar el fondo de tal modo que sirva como imagen de referencia, y por medio de la diferencia con la imagen capturada detectar los cambios.

Los cambios entre los píxeles de la imagen de referencia y la actual posibilitan que mediante la aplicación de un determinado umbral se pueda binarizar la imagen para identificar el marcador.

Una vez identificado el objeto, se establece el patrón de color que lo identifica, y se utiliza el algoritmo meanshift, para realizar y el seguimiento y proporcionar la localización del marcador.

Para que los resultados sean independientes de las condiciones de iluminación, habrá que utilizar un modelo de color como el HSI (Tono, Saturación, Intensidad), en vez del RGB.

#### 3.2.1 Algoritmo meanshift

El algoritmo meanshift permite el seguimiento de objetos basado en una segmentación en color, por medio de la búsqueda del mejor candidato en las posiciones vecinas de dos frames consecutivos.

Así:

- Se establece un modelo de referencia en el frame actual.
- Se calcula el modelo.

- Se busca el modelo en el siguiente frame.

Para encontrar el modelo se empieza en la posición actual y se busca en las posiciones vecinas del siguiente frame, así, cuando se encuentra el mejor candidato según una función de máxima similitud, se detiene el algoritmo y se repite con el siguiente par de frames.



#### Captura del fondo

Cuando se inicia la cámara la imagen estará variando hasta que el sensor se acomode. En ese momento se capturará la imagen de referencia dado que se tendrá una imagen estable, donde las únicas variaciones serán debidas a los errores del propio sistema de captura y a las pequeñas variaciones ambientales.



#### Detección de movimiento

Para detectar el movimiento, una forma sencilla, consiste en realizar la sustracción entre la imagen actual y la de referencia. De esta forma, la imagen resultante contiene los cambios entre imágenes, con lo que mediante la aplicación de un determinado umbral permite identificar los cambios (presencia del marcador).

Así, se restan los píxeles de las imágenes en RGB y se convierte la diferencia a nivel de gris.

Para convertir entre RGB y escala de grises se aplica:

$$GRAY = 0,299 \cdot R + 0,587 \cdot G + 0,114 \cdot B$$



#### Segmentación

Para obtener el umbral que permita segmentar la imagen de forma automática se utiliza el método Otsu.

Este método proporciona el umbral óptimo (threshold) para la segmentación de la imagen bajo el criterio de máxima varianza entre fondo (background) y objeto (foreground).

Así, se calcula la varianza entre todas las posibles divisiones y se toma el umbral que presenta la máxima varianza entre clases, de tal modo que:

$$T = \max(\sigma^2) \quad [\text{umbral óptimo}]$$

siendo



$$\sigma^2 = w_B (\mu_B - \mu)^2 + w_F (\mu_F - \mu)^2 \quad [\text{varianza}]$$

$$w_k = \sum_{i=0}^k p_i \quad [\text{probabilidad acumulada}]$$

$$\mu_k = \sum_{i=0}^k i \cdot p_i \quad [\text{media acumulada}]$$

$$\mu_c = \frac{\mu_k}{\omega_k} \quad [\text{media de la clase (B o F)}]$$

Donde  $p_i$  es la probabilidad de aparición de un determinado nivel  $i$ .

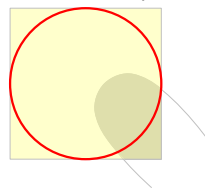


### Identificación

Una vez obtenida la imagen binaria podría parecer que el resultado será nuestro marcador, pero lo cierto es que probablemente el resultado presente agrupaciones de píxeles que deben ser descartadas. De modo que sería recomendable aplicar un algoritmo de conectividad para identificar el objeto, discerniendo entre las distintas agrupaciones de píxeles (blobs<sup>2</sup>) para poder seleccionar el correcto según un criterio de tamaño, pero por simplicidad, aún no se ha implementado.

Para restringir la búsqueda y optimizar el procesado, se define una ventana de búsqueda, que para simplificar será cuadrada, de tal modo que su tamaño queda definido por el radio del círculo inscrito.

Ventana de búsqueda



Al definir una ventana de búsqueda, se tiene un comportamiento similar al que presentaría una ficha de parchís, donde permanece inmóvil mientras no se desliza con el movimiento del dedo que se sitúa encima.



### Localización

Para establecer la localización del marcador se utilizan sus características geométricas.

<sup>2</sup> Los blobs son agrupaciones de píxeles, en imágenes binarias, entre los cuales existe conectividad.



El centroide o centro de masas (  $X_{CM}$  ,  $Y_{CM}$  ) viene dado por:

$$( X_{CM} = \frac{\sum X_i}{A} , Y_{CM} = \frac{\sum Y_i}{A} )$$

Donde el tamaño queda definido por medio del área (  $A$  ), que no es más que el número de píxeles que lo componen, y que también se utiliza para identificar el marcador, dado que será el mayor.

Se puede obtener la orientación del marcador por medio de la determinación de los ejes principales de inercia, cuya dirección queda definida por medio del ángulo (  $\theta$  ), siendo aquel que forman los ejes principales con respecto al sistema de referencia (  $x$  ,  $y$  ) situado en el centro de masas.

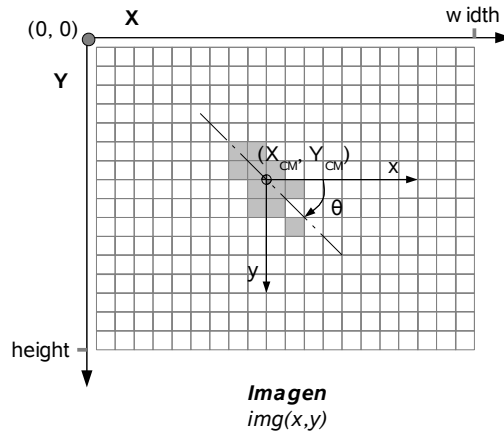
$$\tan(2\theta) = \frac{-2 \cdot P_{xy}}{I_x - I_y}$$

Donde los momentos de inercia del blob respecto a los ejes  $x$  e  $y$  vienen dados por:

$$I_x = \sum y_i^2 , \quad I_y = \sum x_i^2$$

Y, el producto de inercia por:

$$P_{xy} = \sum x_i \cdot y_i$$



Dado que la posición de los píxeles viene dada para el sistema de referencia absoluto (  $X$  ,  $Y$  ) de la imagen, habrá que aplicar el teorema de Steiner para determinar los momentos y el producto de inercia que hay que aplicar.

$$I_p = I_{CM} + A \cdot d_p^2$$

De tal modo, que los momentos y productos de inercia quedan definidos a partir de aquellos respecto al origen de la imagen, como:

$$I_X = \sum Y_i^2 - A \cdot Y_{CM}^2$$

$$I_Y = \sum X_i^2 - A \cdot X_{CM}^2$$

$$P_{XY} = \sum X_i \cdot Y_i - A \cdot X_{CM} \cdot Y_{CM}$$

### 3.3 Implementación

La implementación se ha realizado en C#, utilizando:

- SharpDevelop IDE
- .NET 2.0



Implementado

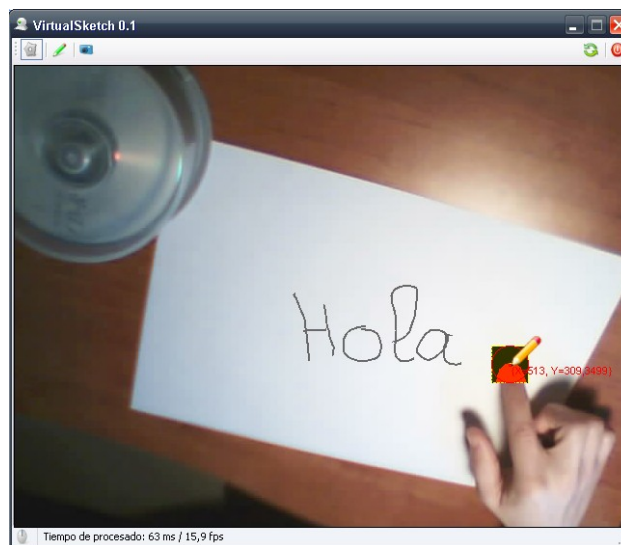


En desarrollo



Capturas de pantalla

Marker	
Public Class	
tracking:	bool
Properties	
Name:	string
Threshold:	byte
Size:	int
Location:	PointF
EstimateLocation:	PointF
Methods	
Process():	void
Diference():	void
OtsuThreshold():	byte
Binarize():	void
Locate():	PointF
KalmanFilter():	PointF



## 4. Filtro de Kalman

### 4.1 Introducción

Todo sistema dinámico<sup>3</sup> puede ser representado en su forma general por un conjunto de ecuaciones de estado (  $x$  ), que en su forma discreta se expresan como:

$$x(k+1) = Ax(k) + Bu(k)$$

$$y(k) = Cx(k) + Du(k)$$

Mientras que el primer conjunto de ecuaciones modela el comportamiento del sistema, el segundo expresa las ecuaciones de medida, o sea, relaciona los datos recogidos por los diferentes sensores y el estado del sistema.

Dado que las medidas (  $y$  ) presentarán un determinado error<sup>4</sup> (  $v$  ), para obtener la medida que más se aproxime a la real se debe encontrar la mejor estimación (  $\hat{x}$  ) para el vector de estado.

Esta mejor estimación se contempla como aquel vector de estado que minimiza la suma de los cuadrados del error<sup>5</sup>, de tal modo que:

$$y = C \cdot x + v \quad \rightarrow \quad \hat{x} = (C^T C)^{-1} C^T \cdot y$$

Dado que algunas medidas serán más fiables que otras, porque no todos los sensores serán igual de fiables, y considerando que las medidas son independientes, se tiene que la matriz de covarianzas (R) será diagonal, con lo que se introduce una matriz de ponderación (  $V$  ) formada por las inversas de las varianzas de cada sensor, de tal modo que:

$$\hat{x}_{opt} = (C^T V C)^{-1} C^T V \cdot y$$

### 4.2 Filtro de Kalman

El filtro de Kalman es una mejora empleada en la estimación del estado de un sistema dinámico, dado que además de las medidas, emplea el modelo del sistema.

3 Un sistema dinámico es aquel que varía con el tiempo.

4 Por el principio de incertidumbre de Heisenberg se sabe que la observación de cualquier sistema implica una alteración del estado del mismo.

5 La pseudoinversa de una matriz [  $(C^T \cdot C)^{-1} \cdot C^T$  ] proporciona una solución de mínimos cuadrados (error mínimo) cuando la matriz del sistema no es cuadrada.

Así, con el modelo del sistema se realiza una predicción del estado que es corregida por el sistema de medida.

Dado que el modelo discreto del sistema podrá ser expresado como:

$$[\text{Predicción}] \quad x(k+1) = A x(k) + B u(k)$$

$$[\text{Medida}] \quad y(k) = C x(k)$$

Y teniendo en cuenta que cada medida y cada estimación tendrán un determinado grado de confianza, y unos determinados errores de medida ( $v(k)$ ) y predicción ( $w(k)$ )<sup>6</sup>, se tiene que:

$$[\text{Ecuación del modelo}] \quad x(k+1) = A(k)x(k) + B(k)u(k) + E(k)w(k)$$

$$[\text{Ecuación de las medidas}] \quad y(k) = C(k)x(k) + v(k)$$

Si se asume que estos errores son gaussianos de media nula e incorrelados, se tiene que sus matrices de covarianza son diagonales, siendo  $Q$  la de la ecuación de propagación, y  $R$  la de la ecuación de medida.

Además, si el sistema es no lineal, ya sea en su ecuación de estado, o en su ecuación de medidas, o en ambas, hay que linealizar dichas ecuaciones mediante el cálculo de los jacobianos entorno al punto de trabajo.

Finalmente, la solución por el filtro de Kalman viene dada por:

- Establecimiento de las condiciones iniciales: estado inicial ( $x(0)$ ) e incertidumbre inicial ( $P(0)$ ).
- Propagación: estimación del estado según el modelo del sistema y la propagación de su covarianza.

$$\hat{x} = A \hat{x} + B u$$

$$P = A P A^T + E Q E^T$$

- Actualización: corrección de la la estimación del estado con la información procedente de las medidas.

$$L = P C^T [C P C^T + R]^{-1}$$

$$P = P - L C P$$

---

<sup>6</sup> Ruido del sistema y error del modelo.

$$\hat{x} = \hat{x} + \underbrace{L}_{\text{corrección}} \left( \underbrace{y}_{\text{medida}} - \underbrace{C \hat{x}}_{\text{estimación}} \right)$$

### 4.3 Aplicación del filtro de Kalman

#### 4.3.1 Estimación de la posición de un teléfono móvil

Se va a aplicar el filtro de Kalman para estimar la posición de un usuario de teléfono móvil que se encuentra en un área en la que se conocen las distancias a tres estaciones base, localizadas en posiciones conocidas.

Se asume que el teléfono se mueve en un espacio plano delimitado por sólo tres estaciones base situadas en las posiciones: (0, 0), (500, 1000) y (1000, 500).

La intensidad de la señal recibida desde cada estación base permite estimar la distancia entre teléfono y repetidor, y se asume media nula y error típico de 10 m más el 10% de la medida correspondiente.

Como el usuario podrá estar en movimiento se asume un desplazamiento aleatorio, de modo que en las condiciones normales de funcionamiento el desplazamiento típico entre dos instantes de muestreo es de 50 m.

#### 4.3.2 Desarrollo

##### 1. Definición del vector de estado.

Dado que se trata de resolver la posición del teléfono móvil respecto al mismo sistema de referencia absoluto que da la situación de las estaciones base, el vector de estado vendrá definido por las coordenadas cartesianas que indican su posición.

$$x = \begin{bmatrix} X \\ Y \end{bmatrix}$$

##### 2. Definición del modelo dinámico.

Como se desconoce totalmente la evolución de la persona que porta el teléfono, se supondrá que está parada, y por tanto, toda variación de posición será asociada al error de la ecuación de propagación del estado (error de predicción).

$$\begin{bmatrix} X \\ Y \end{bmatrix}(k+1) = \begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} \begin{bmatrix} X \\ Y \end{bmatrix}(k) + w(k)$$

##### 3. Modelado del error de predicción.

Dado que se asume un desplazamiento aleatorio de unos 50 m, entre dos muestreos consecutivos, y dado que se asume que este error es de media nula y no

correlado, la matriz de covarianzas de la estimación (  $Q$  ) vendrá definida por los valores de varianza ocupando las posiciones de la diagonal.

$$w(k) \simeq N(0, Q) \quad / \quad Q = \begin{bmatrix} 50^2 & 0 \\ 0 & 50^2 \end{bmatrix}$$

#### 4. Definición de las ecuaciones de medida.

Dado que la medida que se tiene es la distancia a cada una de las estaciones base, vendrá dada en función de las coordenadas cartesianas por la distancia Euclídea, de tal modo que:

$$d_i = \sqrt{(X - X_i)^2 + (Y - Y_i)^2} + v_i \quad \text{con } i=1,2,3$$

#### 5. Linealización de las ecuaciones de medida.

Como resulta que la ecuación de medida no es lineal, para poder aplicar el filtro de Kalman, hay que linealizarla por medio de la aplicación del jacobiano, de tal modo que:

$$\mathbf{H} = \begin{bmatrix} h_{11} & h_{12} \\ h_{21} & h_{22} \\ h_{31} & h_{32} \end{bmatrix} \quad \text{siendo } h_{ij} = \frac{\partial d_i}{\partial x_j} = \frac{(x_j - x_j^i)}{\sqrt{(x_1 - x_1^i)^2 + (x_2 - x_2^i)^2}}$$

#### 6. Modelado del ruido.

Dado que el error de las medidas es de 10 m más un 10% de la distancia, se tiene que:

$$v(k) \simeq N(0, R) \quad / \quad R = \begin{bmatrix} (10+0,1 \cdot d_1)^2 & 0 & 0 \\ 0 & (10+0,1 \cdot d_2)^2 & 0 \\ 0 & 0 & (10+0,1 \cdot d_3)^2 \end{bmatrix}$$

#### 7. Establecimiento de valores iniciales.

Definido el problema, para establecer los valores iniciales que permitan ejecutar el filtro se asumirá que el móvil está cerca del origen, y dado que el estado inicial es desconocido se asignará una covarianza muy grande, una desviación típica de 500 m en cada coordenada, de tal modo que se tienda hacia una estimación real de la posición del usuario.

$$x(0) = \begin{bmatrix} X \\ Y \end{bmatrix} (0) = \begin{bmatrix} 10 \\ 10 \end{bmatrix} \quad P(0) = \begin{bmatrix} 500^2 & 0 \\ 0 & 500^2 \end{bmatrix}$$

## 5. Identificación de un contorno rectangular

### 5.1 Introducción

Cuando se quiere obtener, independientemente de la posición de la cámara, la imagen de un plano delimitado por un contorno rectangular, como una hoja de papel o la pantalla de un ordenador, se puede establecer la proyección que define la verdadera proporción de la imagen conocidas las dimensiones reales del rectángulo e identificándolo en la imagen.

Para identificar ese contorno rectangular se realiza un proceso de tres etapas:

1. Se aplica un operador tipo laplaciano para detectar los bordes existentes en la imagen.
2. Sobre la imagen de bordes, se identifican los bordes rectos por medio de la aplicación de la transformada de Hough para líneas.
3. Una vez obtenidas las líneas, se buscan aquellas que verifican las condiciones impuestas por un contorno rectangular.

### 5.2 Detección de bordes

Un borde es toda región de la imagen donde aparece una fuerte variación del nivel de intensidad entre píxeles adyacentes.

A la hora de identificar un borde en una imagen, existen dos posibilidades matemáticas, aplicar un filtro basado en un operador gradiente o en un operador laplaciano.

La diferencia estriba en que mientras que con el operador gradiente se obtienen los niveles de variación entre píxeles, con el laplaciano lo que se obtiene son los puntos de inflexión de las variaciones de intensidad, o sea los cambios de tendencia.

Dado que todos los operadores de gradiente proporcionan resultados similares, se suele emplear el operador de Sobel, cuyo kernel de convolución varía según la dirección de detección de los bordes, así:

$$\begin{bmatrix} -1 & -2 & -1 \\ 0 & 0 & 0 \\ 1 & 2 & 1 \end{bmatrix} \quad \text{o} \quad \begin{bmatrix} -1 & 0 & 1 \\ -2 & 0 & 2 \\ -1 & 0 & 1 \end{bmatrix} \quad [\text{Sobel}]$$

Dado que se desconoce la orientación del contorno, se opta por el operador

laplaciano, que además es más exacto que el gradiente en la precisión del borde , siendo su kernel de convolución típico:

$$\begin{bmatrix} 0 & -1 & 0 \\ -1 & 4 & -1 \\ 0 & -1 & 0 \end{bmatrix} \quad [\text{Laplaciano}]$$

### 5.3 Detección del contorno rectangular

Para la identificación de contorno rectangular se aplica, sobre la imagen de bordes binarizada, la transformada de Hough para líneas, para de este modo identificar las líneas rectas de la imagen que permitirán encontrar las que conforman el contorno rectangular.

#### Transformada de Hough para líneas

La transformada de Hough para líneas es una técnica muy robusta que permite encontrar puntos alineados en una imagen binarizada, o sea, detectar líneas rectas.

Una serie de puntos  $(x_i, y_i)$  alineados definen una línea recta cuya ecuación en la forma explícita es  $y = m \cdot x + b$ . Con lo que cada recta del espacio de la imagen  $(X, Y)$  vendrá dada por un único par  $(m, b)$  en el espacio paramétrico.

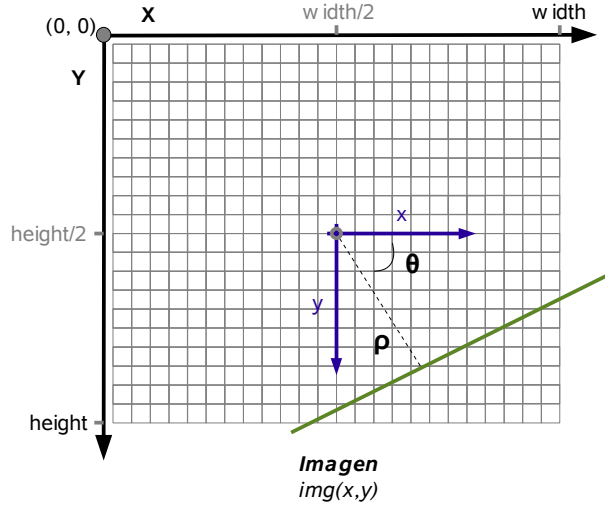
Debido a que tanto la pendiente de la recta  $(m)$  como el término independiente  $(b)$  tienden a infinito cuando la recta tiene una posición cercana a la vertical, en lugar de utilizar la ecuación  $y = m \cdot x + b$  se emplea la parametrización  $(\rho, \theta)$ , que define la recta en su forma polar:

$$\rho = x \cdot \cos \theta + y \cdot \sin \theta$$

Por tanto hay que realizar una transformación entre el plano de imagen  $(x, y)$  y el espacio de parámetros  $(\rho, \theta)$ .



### Identificación de un contorno rectangular



Como por cada punto de la imagen podrán pasar un número infinito de rectas, y por tanto podrá presentar un conjunto infinito de parámetros, se toma un espacio de parámetros discretizado.

Así, dado que dos puntos que estén sobre la misma recta presentarán un par de parámetros comunes, si por cada punto perteneciente a la recta se genera un voto sobre un acumulador, se tiene que el número de votos para un par de parámetros es el número de puntos perteneciente a esa recta.

Para ello, en la transformada de Hough se crea una matriz de acumulación  $A(i,j)$  que subdivide el espacio paramétrico en una serie de intervalos que irán desde  $\rho_{min}$  a  $\rho_{max}$ , y desde  $\theta_{min}$  a  $\theta_{max}$ , de tal modo cada celda  $(i,j)$  de la matriz contendrá la probabilidad de que  $(\rho_i, \theta_j)$  sea una recta de la imagen.

Dado que la parametrización  $(\rho, \theta)$  se define desde el centro de la imagen, los intervalos de valores posible para una imagen de tamaño  $W \times H$  vendrán dados por:

$$\rho \in \left[ \underbrace{\frac{-\sqrt{W^2 + H^2}}{2}}_{-d}, \underbrace{\frac{\sqrt{W^2 + H^2}}{2}}_d \right]$$

$$\theta \in [0, \pi)$$

De este modo se toma un conjunto de ángulos  $\theta$  de prueba, resultado de dividir 90 veces el intervalo  $[0^\circ, 180^\circ)$ , y con cada uno de ellos se calculan los valores de  $\rho$  para cada punto de la imagen y se vota la celda de acumulación  $A(i,j)$  correspondiente a  $(\theta, \rho)$ , estando el intervalo  $[-d, d]$  dividido en 100 partes, donde  $d$  es la mitad de la diagonal de la imagen.

Finalmente se tiene que las celdas de la matriz de acumulación  $A(i,j)$ , más

votadas, serán las rectas más probables, y siempre según la precisión de las subdivisiones tomadas.

#### Algoritmo

```
// Descripción del algoritmo

1. Inicializar el acumulador a cero.
2. Para cada punto de la imagen calcular los posibles
   valores de  $\rho$  partiendo de los valores de  $\theta$ .
    $\rho = x \cdot \cos \theta + y \cdot \sin \theta$ 
3. Incrementar la celda del acumulador correspondiente.
    $A(\theta, \rho) ++$ 
4. Identificar las celdas con mayor valor.
```

### 5.4 Identificación del contorno rectangular

Finalmente, para localizar los vértices que definen el contorno rectangular, se buscarán entre todas las líneas más probables proporcionadas por la transformada de Hough aquellas primeras que cumplan las siguientes condiciones:

- La orientación de dos lado vecinos debe diferir en al menos  $20^\circ$  (lados perpendiculares).
- La orientación de dos lados opuestos no debe diferir en más de  $20^\circ$  (lados paralelos).
- Los lados opuestos deben presentar un  $\rho$  diferente en al menos 40 píxeles.

## 6. Cálculo de la homografía

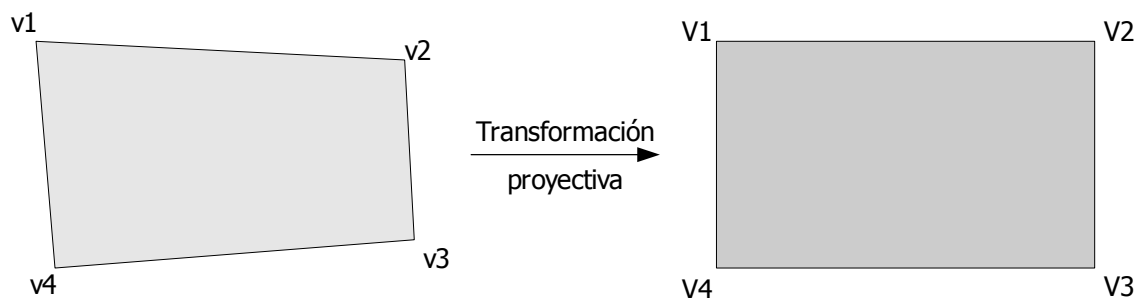
Para establecer la correspondencia entre las coordenadas de la imagen y las reales se asume que la cámara cumple con una perspectiva proyectiva (modelo pin-hole), de modo que la relación entre la proyección de un plano y su vista ortogonal viene dada por la homografía.

### 6.1 Homografía

Dado un punto  $(x, y)$  que se sitúa en la proyección del plano identificado en la imagen, el punto correspondiente en la vista ortogonal de ese plano buscado  $(X, Y)$  viene dado por la relación que establece la matriz de homografía ( $H$ ), de tal modo que:

$$\lambda \begin{pmatrix} X \\ Y \\ 1 \end{pmatrix} = \underbrace{\begin{pmatrix} h_{11} & h_{12} & h_{13} \\ h_{21} & h_{22} & h_{23} \\ h_{31} & h_{32} & 1 \end{pmatrix}}_H \begin{pmatrix} x \\ y \\ 1 \end{pmatrix}$$

donde  $\lambda$  es un factor de escala distinto de cero que queda fijado al hacer  $h_{33}=1$ .



Así, se tiene que para determinar  $H$  y con ella realizar la corrección de orientación y posición que permita obtener la posición real del marcador, basta con establecer la correspondencia entre los cuatro pares de puntos, los determinados en la etapa anterior sobre la imagen, y los que representan a los vértices del contorno rectangular de dimensiones reales, quedando un sistema de ecuaciones lineales con solución única, tal que:

$$\begin{pmatrix} x_1 & y_1 & 1 & 0 & 0 & 0 & -(x_1 X_1) & -(y_1 X_1) \\ x_2 & y_2 & 1 & 0 & 0 & 0 & -(x_2 X_2) & -(y_2 X_2) \\ x_3 & y_3 & 1 & 0 & 0 & 0 & -(x_3 X_3) & -(y_3 X_3) \\ x_4 & y_4 & 1 & 0 & 0 & 0 & -(x_4 X_4) & -(y_4 X_4) \\ 0 & 0 & 0 & x_1 & y_1 & 1 & -(x_1 Y_1) & -(y_1 Y_1) \\ 0 & 0 & 0 & x_2 & y_2 & 1 & -(x_2 Y_2) & -(y_2 Y_2) \\ 0 & 0 & 0 & x_3 & y_3 & 1 & -(x_3 Y_3) & -(y_3 Y_3) \\ 0 & 0 & 0 & x_4 & y_4 & 1 & -(x_4 Y_4) & -(y_4 Y_4) \end{pmatrix} \cdot \begin{pmatrix} h_{11} \\ h_{12} \\ h_{13} \\ h_{21} \\ h_{22} \\ h_{23} \\ h_{31} \\ h_{32} \end{pmatrix} = \begin{pmatrix} X_1 \\ X_2 \\ X_3 \\ X_4 \\ Y_1 \\ Y_2 \\ Y_3 \\ Y_4 \end{pmatrix}$$

Cada punto proporciona dos ecuaciones lineales para la solución de los elementos de la matriz H, con lo que conocidos 4 puntos se tiene una solución exacta, ya que  $h_{33} = 1$ .



Si se tuviesen más de cuatro puntos, la matriz sería sobredeterminada y H tendría que ser estimado en términos de minimización, donde la mejor opción pasaría por aplicar el método de descomposición en valores singulares (SVD), dado que facilita la obtención de una solución óptima y robusta de un sistema de ecuaciones lineales, a pesar de que se encuentre mal condicionado.

## 6.2 Resolución de sistemas de ecuaciones lineales

A la hora de resolver un sistema de ecuaciones lineales, la solución pasa por aplicar alguno de los métodos existentes, pudiendo ser directo o iterativo según las características del problema en cuestión.

$$A_{n \times n} \cdot x_n = b_n$$

En este caso se opta por aplicar un método directo, concretamente el método de eliminación de Gauss con pivotación parcial, con lo que se logra minimizar los problemas debidos a los errores de redondeo. Esto se debe a que con este tipo de pivotación se toma como pivote el elemento de mayor valor absoluto de la columna.

Este es el método más sencillo de resolución de sistemas de ecuaciones lineales.

1. Eliminación: consiste en reducir el sistema a uno triangular superior<sup>7</sup>.
2. Sustitución: consiste en la obtención de la solución por medio del método conocido como sustitución regresiva.

Otra opción podría ser aplicar el método de Gauss-Jordan, con el que se elimina la etapa de sustitución regresiva al eliminar los elementos a ambos lados de la diagonal.

<sup>7</sup> Matriz que tiene todos sus coeficientes debajo de la diagonal iguales a cero.