Visión por computadora
Computer vision

- Conjunto de algoritmos que permiten obtener una ***representación visual*** del ***mundo***, ***suficiente*** para la realización de una **tarea dada**.
  - Representación visual
  - El mundo: definirlo.
  - Suficiente: necesidad vs. Posibilidad
  - Acotado a una tarea

Visión por computadora

- Representación visual del mundo
  - Inferir las propiedades del mundo a partir de una o más imágenes:
    - Fotografías
    - Video
    - Estudios médicos
  - El mundo en cada caso es algo distinto

Visión por computadora

- Definición de la tarea
  - No existe un sistema general de visón por computadora capaz de resolver cualquier problema.
  - Buena parte del resultado de la aplicación se juega en una correcta discusión y definición inicial del problema, y en una estrecha relación con los utilizadores o clientes del sistema.

Visión por computadora

- Representación “suficiente” para la tarea
  - Compromiso entre la cantidad de información que una máquina puede, en un momento dado de la historia, almacenar y procesar en un tiempo útil y la definición precisa de la tarea que queremos que esta máquina realice.

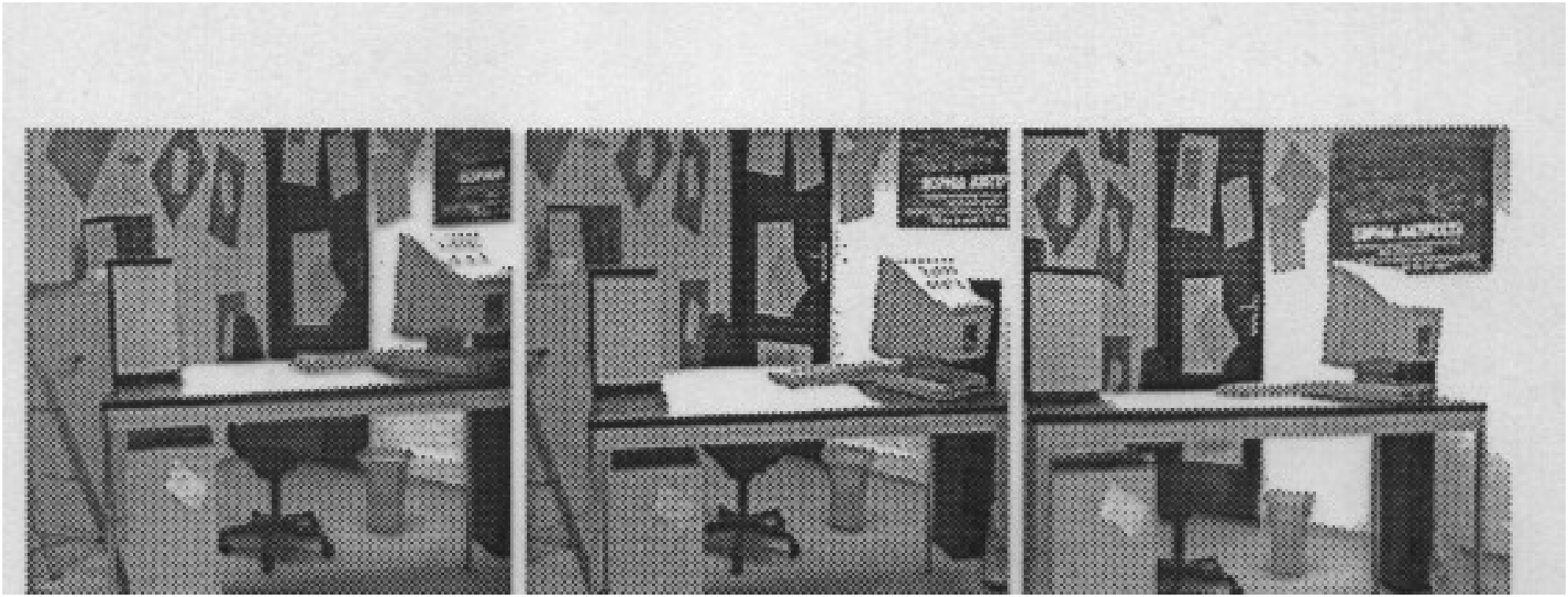
Visión por computadora

- Representación suficiente
  - Utilizar todo el potencial técnico de que dispongamos.
  - No exigir del sistema más que lo necesario para resolver la tarea.
  - En este proceso a menudo es posible interactuar con el medio y es deseable que así sea.

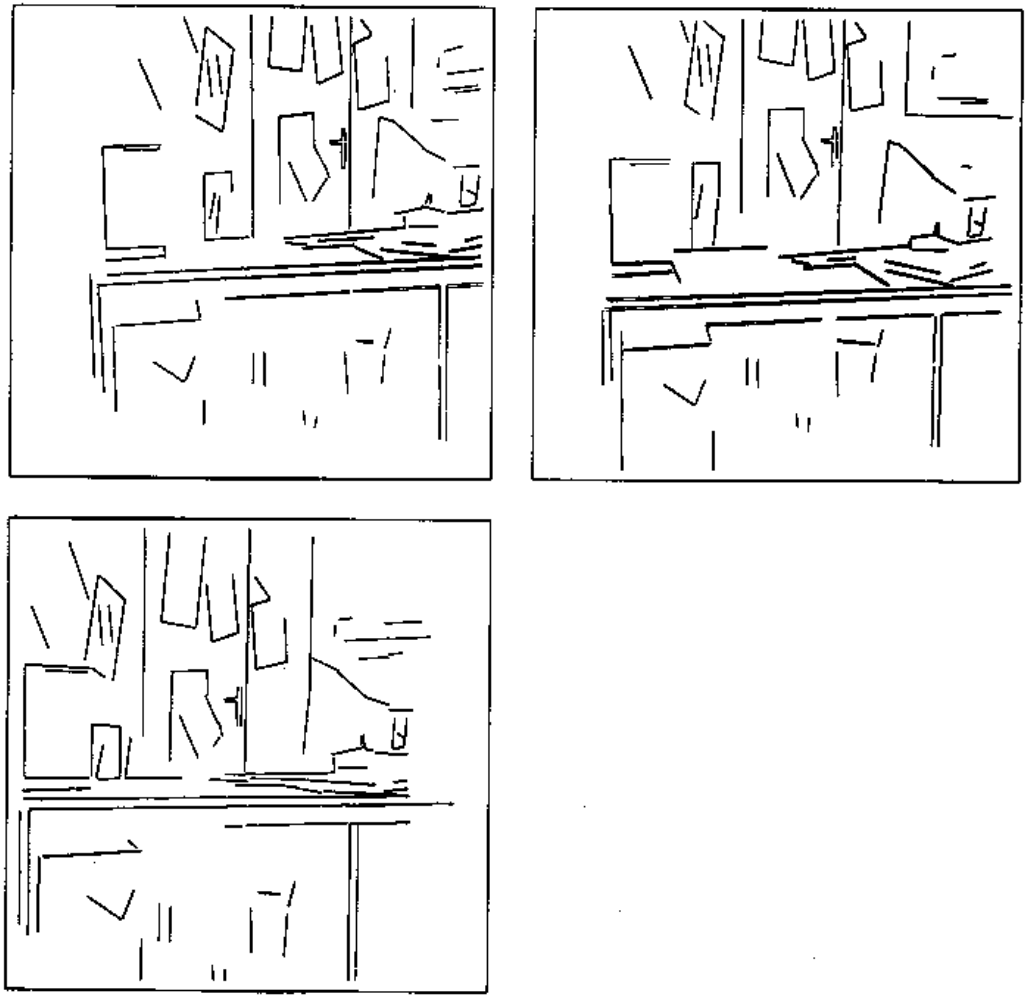
Ej: Robot móvil con 3 cámaras

- Ej: robot móvil con tres cámaras que se desplaza en una pieza (1990)
  - El mundo
  - La tarea
    - Desplazarse sin chocar
  - Representación
    - Conjunto de segmentos detectados en las imágenes de las tres cámaras

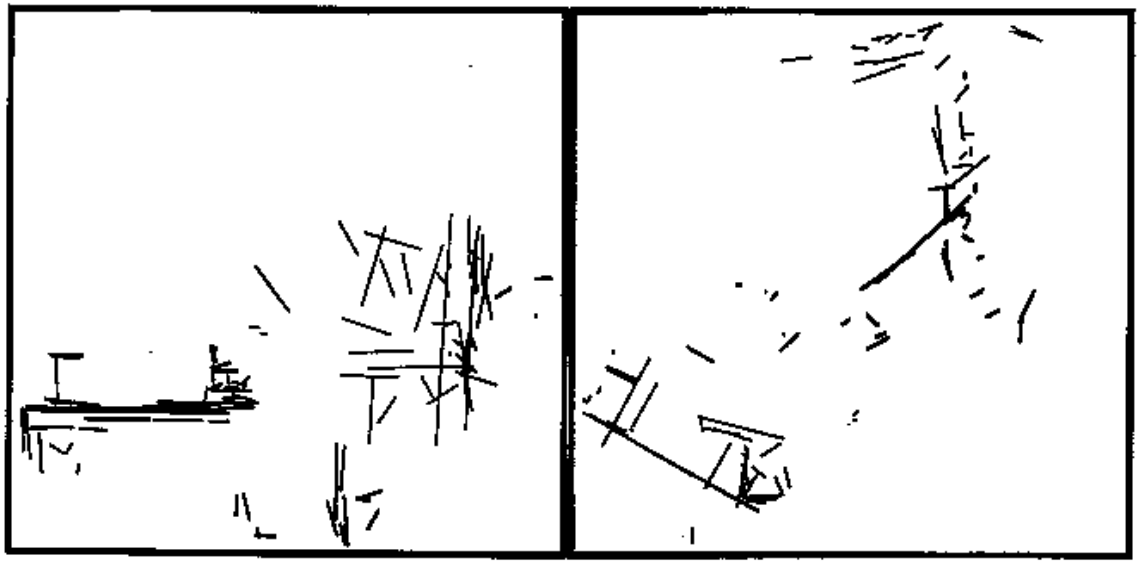
Lo que el robot “ve”



Segmentos registrados



Reconstrucción 3D



Ej: Pieza en cinta trasportadora

- Robot que toma un objeto de la línea transportadora
- “Visual servoing”

Pieza en cinta trasportadora

- El problema así definido no permite enfrentar correctamente la tarea.
- En realidad el problema está mal definido.
- Se podría pensar en varias clases de problemas de complejidad creciente:

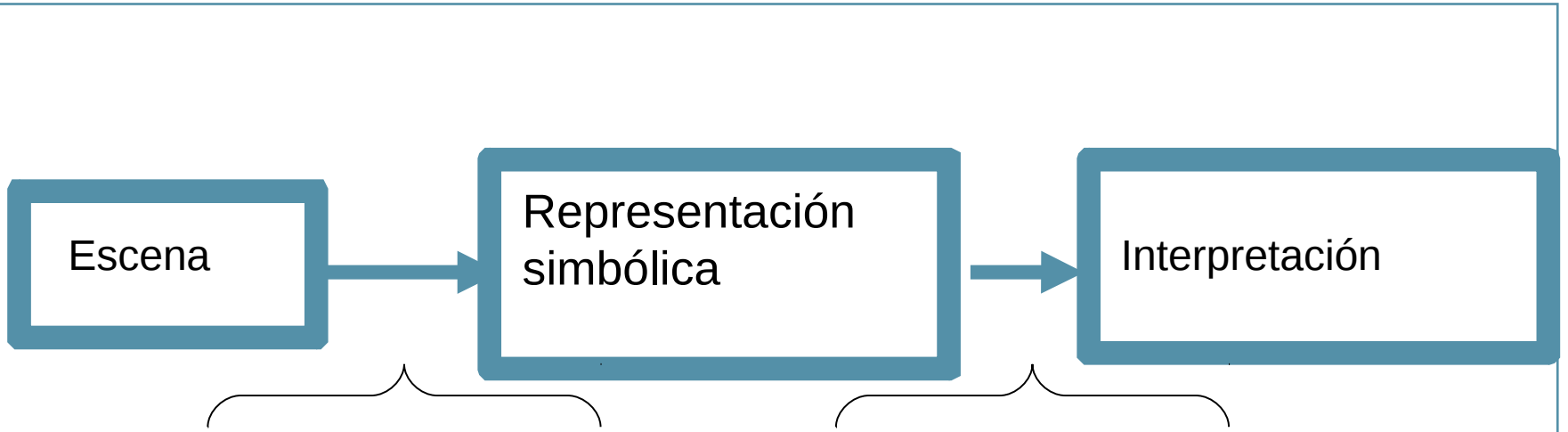
Pieza en cinta transportadora

- Una pieza conocida por vez, sobre un fondo contrastado.
- Una pieza desconocida por vez, sobre un fondo contrastado.
- Una pieza conocida de un montón en un fondo contrastado.
- Una pieza desconocida de un montón en un fondo contrastado.
- Una pieza conocida de un montón en un fondo no contrastante o desconocido.
- Una pieza desconocida de un montón en un fondo desconocido.

Definición de la tarea

- No se puede aislar el aspecto **imágenes** del **sistema** del que forma parte.
- Es fundamental definir correctamente la tarea a realizar y poder utilizar todas las herramientas del sistema en su conjunto para resolver el problema.
- Entre esas herramientas se encuentra el tratamiento de imágenes pero también una disposición adecuada de la cámara, una correcta iluminación, un fondo estable y controlado (si posible contrastante respecto al objeto), etc.

Visión por computadora

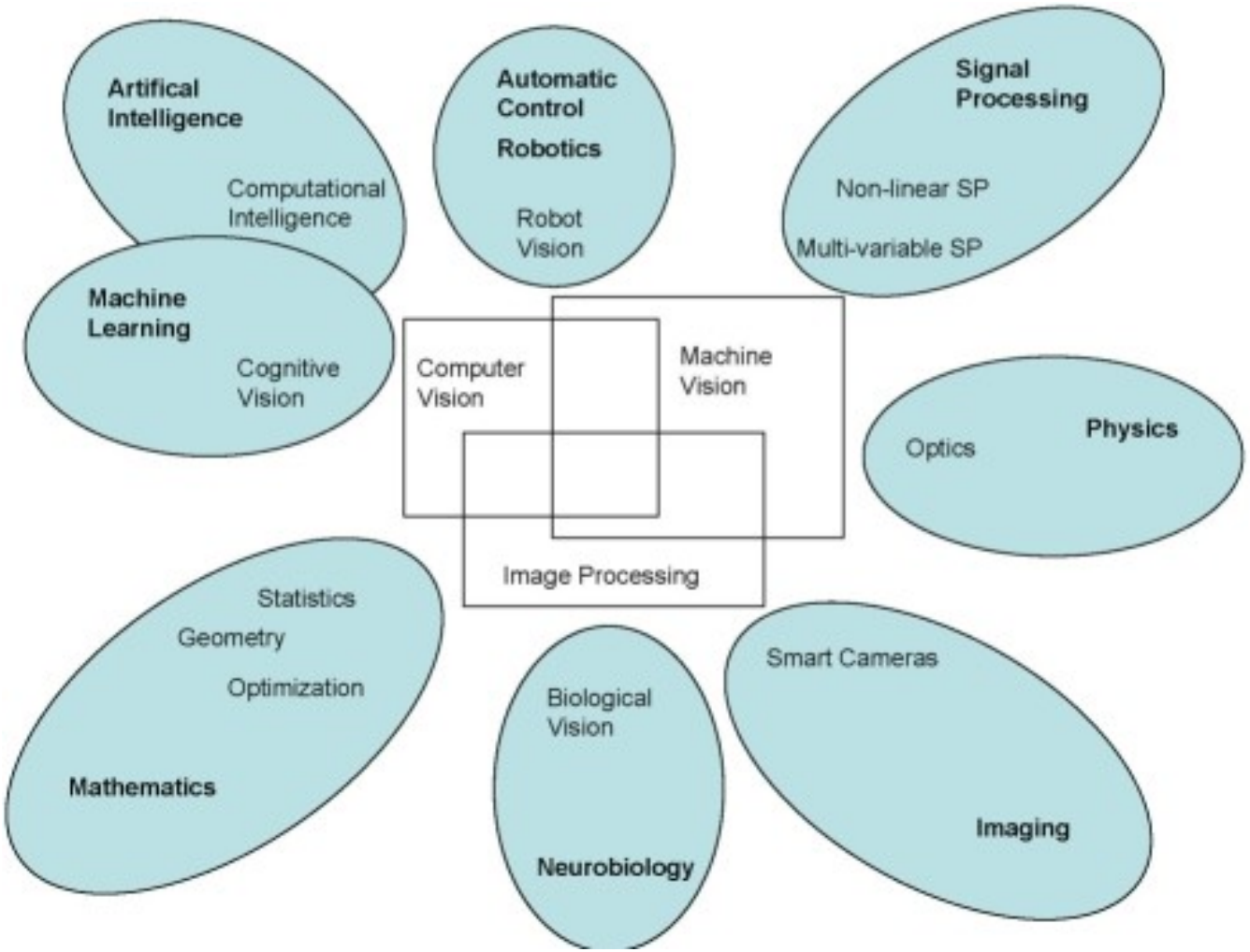


Preprocesamiento.
Segmentación.
Extracción de atributos.

Clasificación.
Apareamiento.
Adapt. a un modelo

Visión por computadora

- Disciplinas próximas



Fuente: Wikipedia

Visión por computadora
Algunas tareas relacionadas con Robótica

- Posibles tareas
  - Detectar la presencia de ciertos objetos en la escena
  - Determinar la pose de un objeto en la escena
  - Determinar qué es fondo y que es frente en la imagen
  - Seguir un objeto o una región de interés
  - Medir distancia del robot a un objeto
  - Evitar obstáculos
  - Ubicarse en el espacio

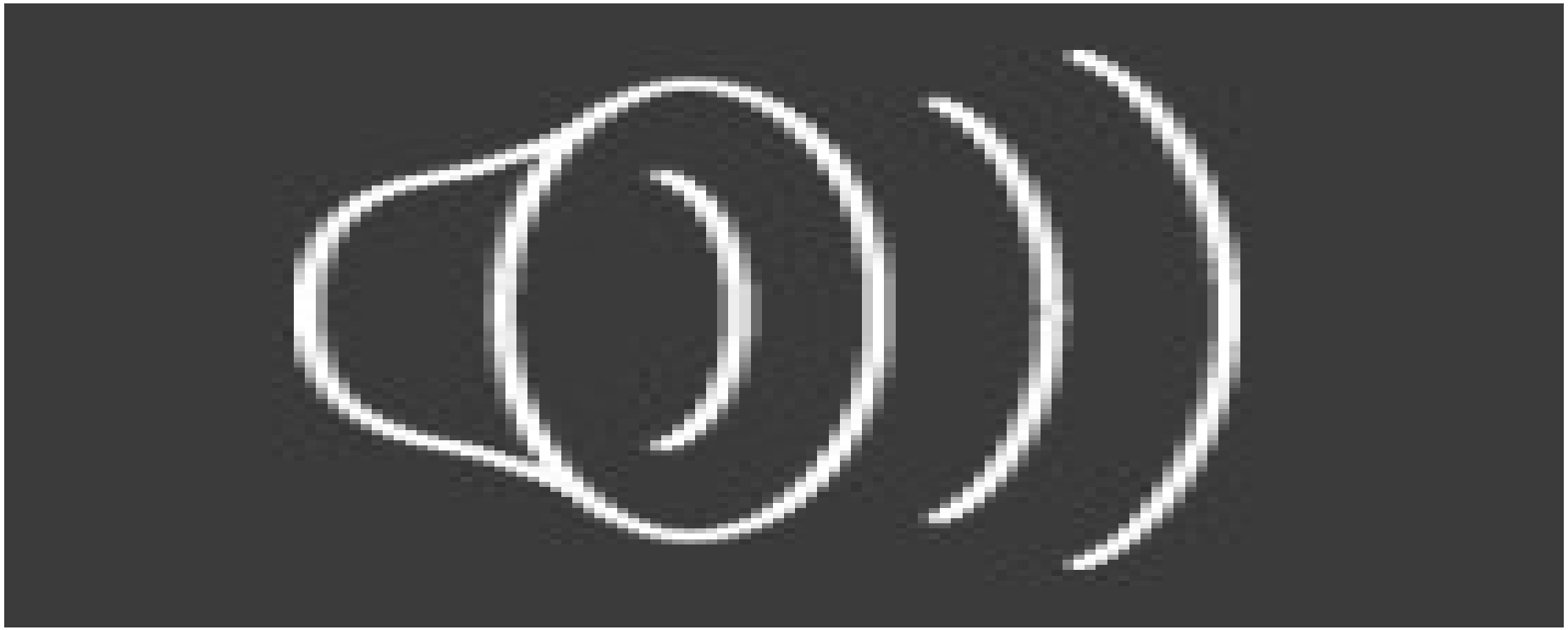
Detectar la presencia de un cierto objeto

- Tener información que describa el objeto
  - Descriptor del objeto dado usualmente por un conjunto de características
- Extraer información de la escena
  - Extraer características.
  - Construir descriptores en base a esas características
- Comparar con el descriptor del objeto
- Es deseable que la deteción sea invariante frente a cambios de escala (el objeto más adelante o más atrás) y a transformaciones rígidas o afines
- Ejemplo: características SIFT o SURF, ASIFT

SIFT o ASIFT

- SIFT: Compara patches (histograma de direcciones locales del gradiente) a distintas escalas. Invariante a cambios de escala y movimientos rígidos.
- ASIFT: agrega invariancia afín.
- SURF variante rápida de SIFT.
- www.ipol.im

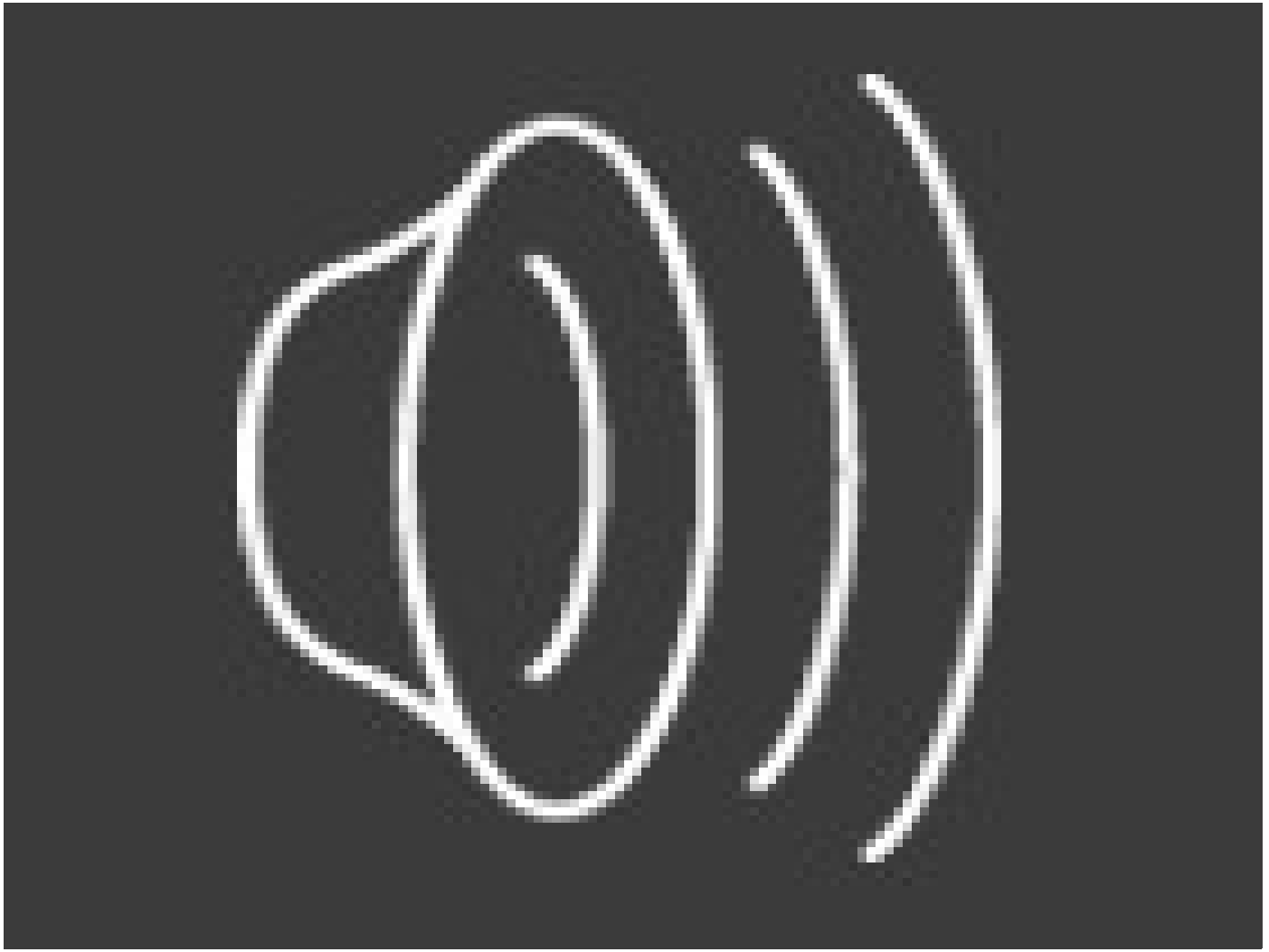
SURF



Detectar la presencia de objetos de un cierto tipo

- Entrenar un clasificador con múltiples ejemplos verdaderos y falsos del tipo de objeto
- Para una nueva imagen, clasificar las regiones de la imagen.
- Ejemplo: reconocimiento de caras
  - En este ejemplo se utilizan características de Haar de patches dispuestos en cierta relación geométrica.

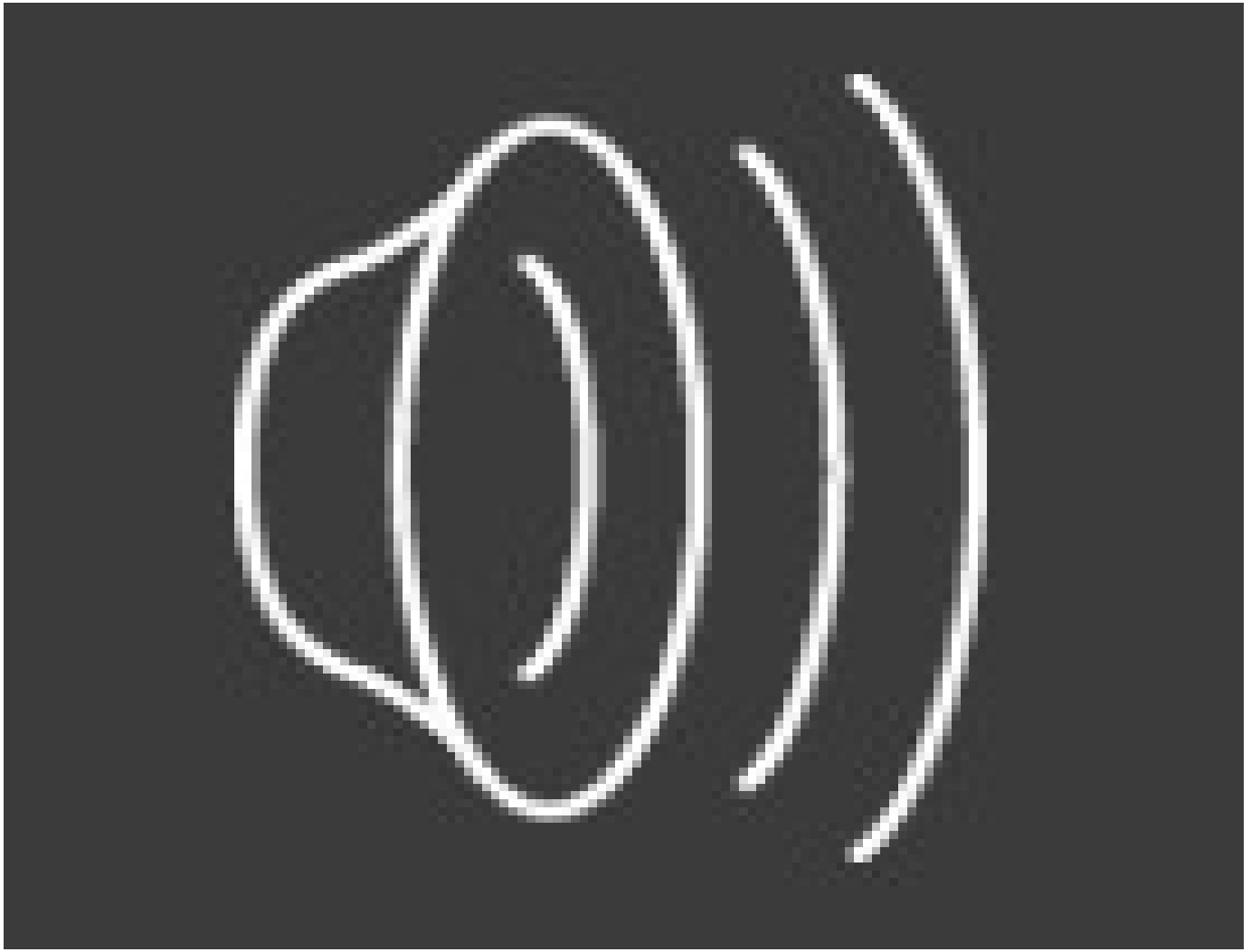
Detección de caras



Determinar frente y fondo

- Aprender el fondo a medida que transcurre el tiempo
- Detectar las zonas que son distintas del fondo
- Ejemplo: sustracción de fondo

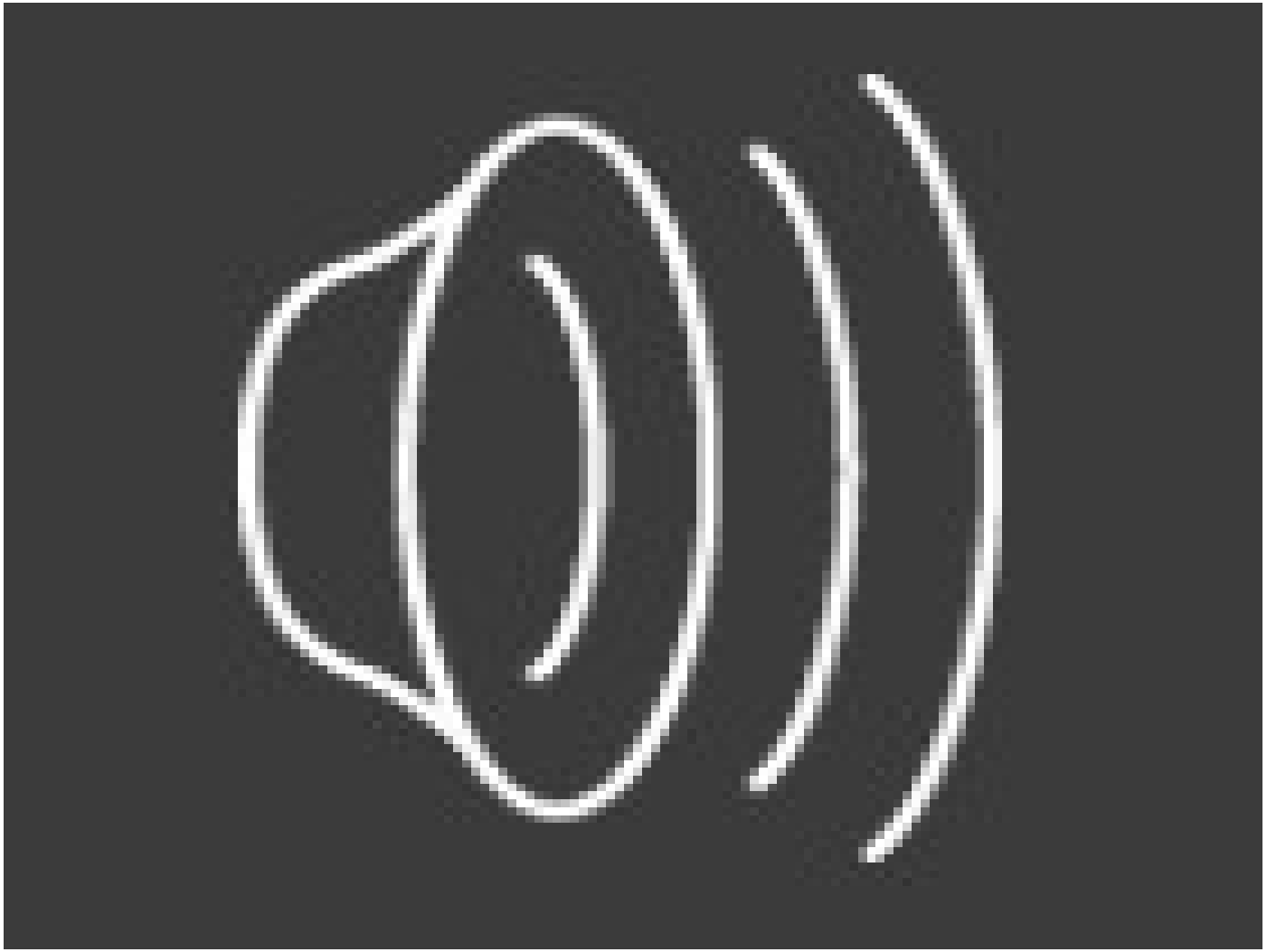
Frente y fondo



Seguir una región por color

- Determinar las características del color de una región (espacio RGB, HSI, etc.)
- Segmentar la región de interés en el primer cuadro
- Buscar en el siguiente cuadro la región cercana que tiene las mismas características.
- Ejemplo: Mean-shift (buscar una región cercana en forma y posición que mantenga el valor medio).
  - http://isa.umh.es/pfc/rmvision/opencvdocs/papers/camshift.pdf
  - demo:
  - http://isa.umh.es/pfc/rmvision/opencvdocs/appPage/CamShift/CamShift.htm
-
-

Mean-shift



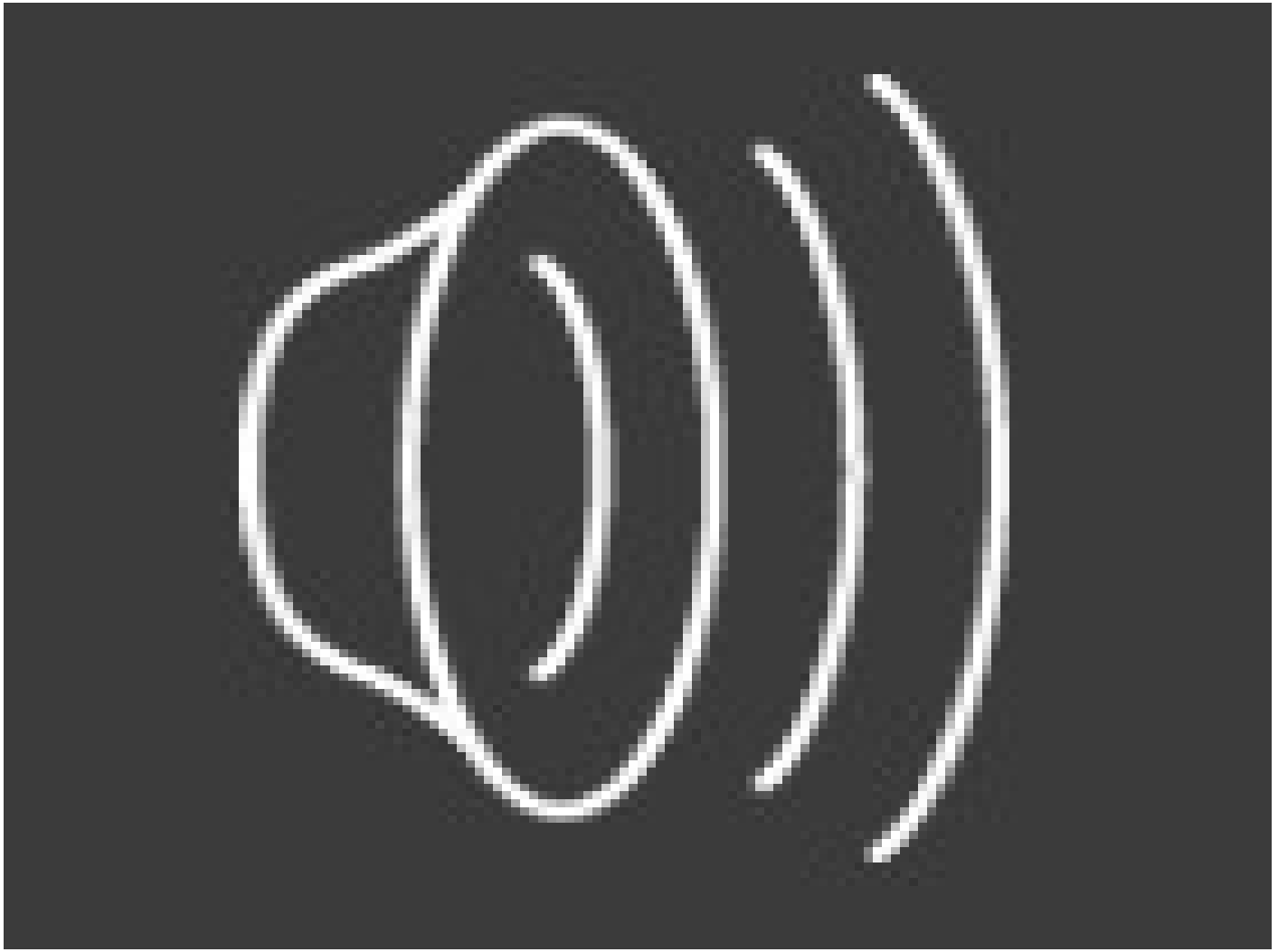
Detección de movimiento

- Detectar zonas que se mueven en la imagen
- Ejemplo: Flujo óptico
  - http://www.cs.umd.edu/~djacobs/CMSC426/OpticalFlow.pdf

Flujo óptico – Lukas/Kanade

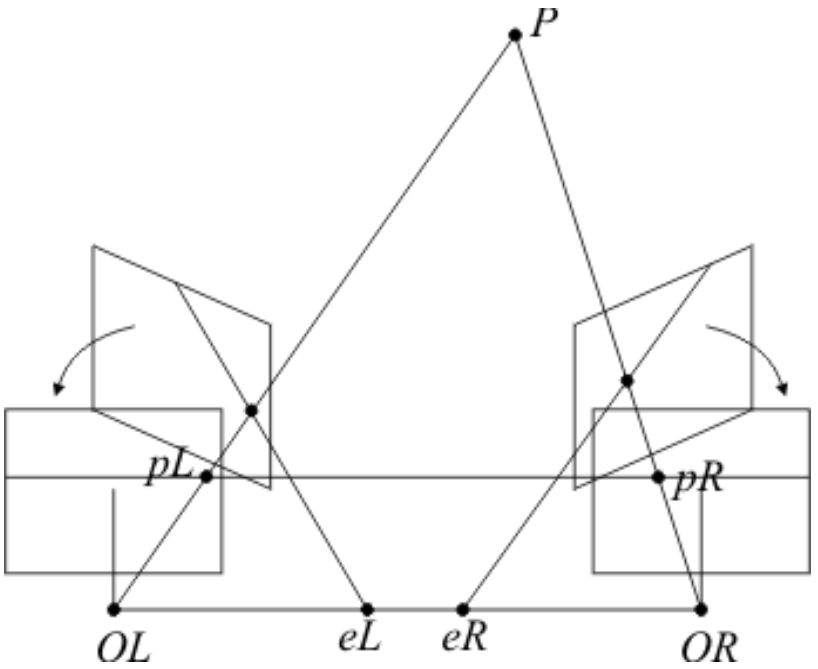
- I,J imágenes “consecutivas” (secuencia)
- Modelos de movimiento
- $I(x, y, t+T) = I(x - \Delta_x(x, y, t, T), y - \Delta_y(x, y, t, T))$
- Movimiento afín $\vec{\delta} = (\Delta_x, \Delta_y) = D\vec{x} + \vec{d}$
- $J((1+D)\vec{x} + \vec{d}) = I(\vec{x})$
- Selección de “buenos puntos” a seguir
- Búsqueda de los parámetros en una ventana centrada en los puntos

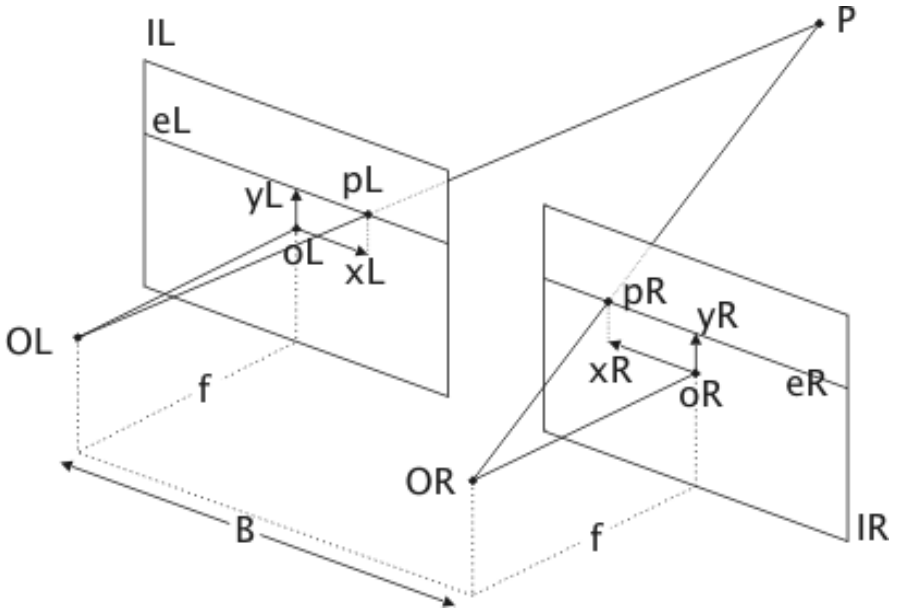
Flujo óptico – Lukas/Kanade



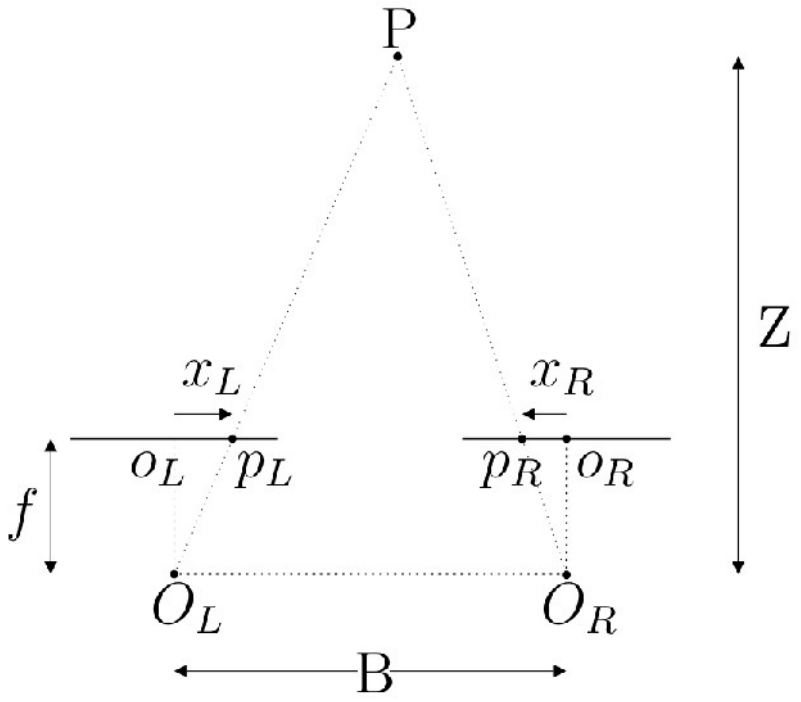
Determinar distancia a un objeto

Determinar distancia a un objeto



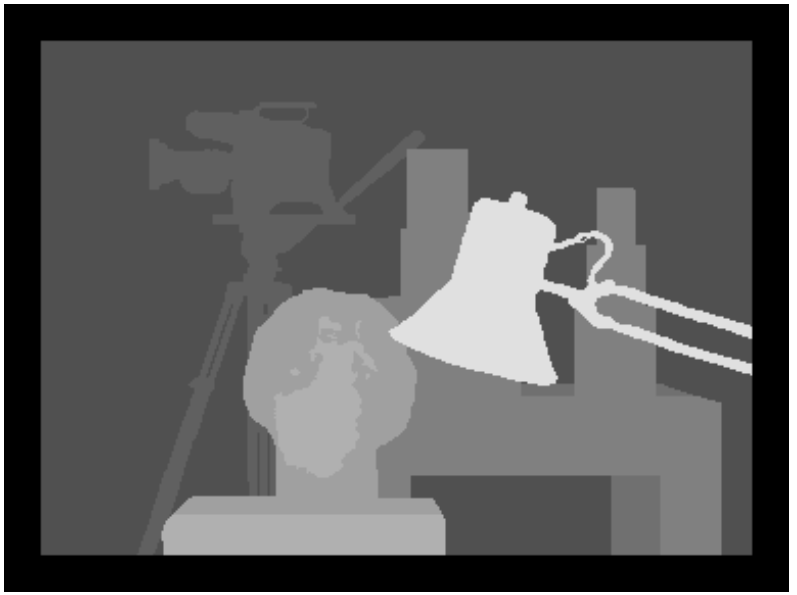


Determinar distancia a un objeto



$$d = (x_L - x_R) = \frac{f}{Z} B$$

Determinar distancia a un objeto

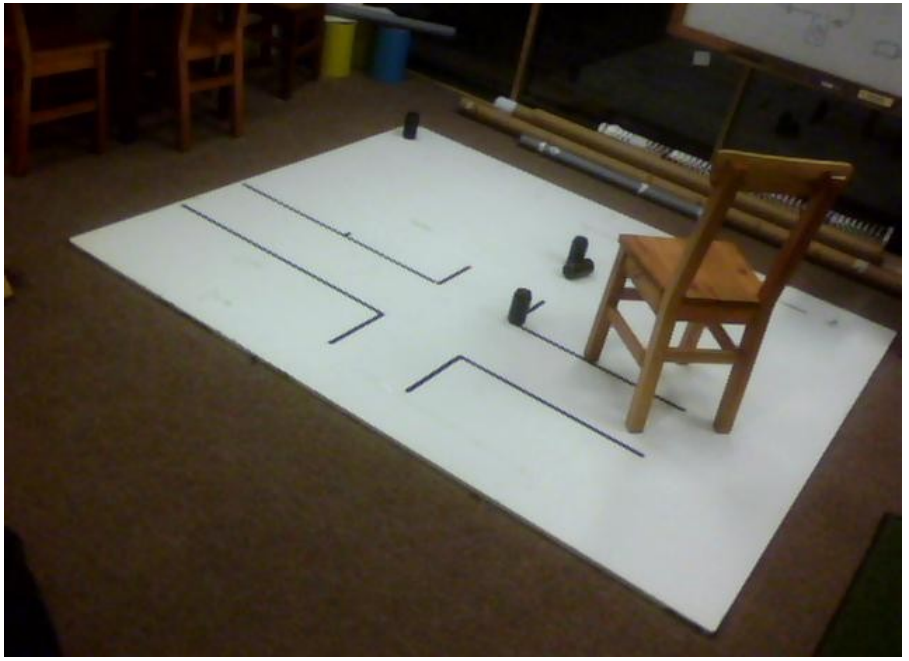


Determinar distancia a un objeto

Computer Vision, Robotics and Arduino

- http://robocv.blogspot.com/2012/04/robot-control-using-opencv-and-arduino.html

Problema del curso



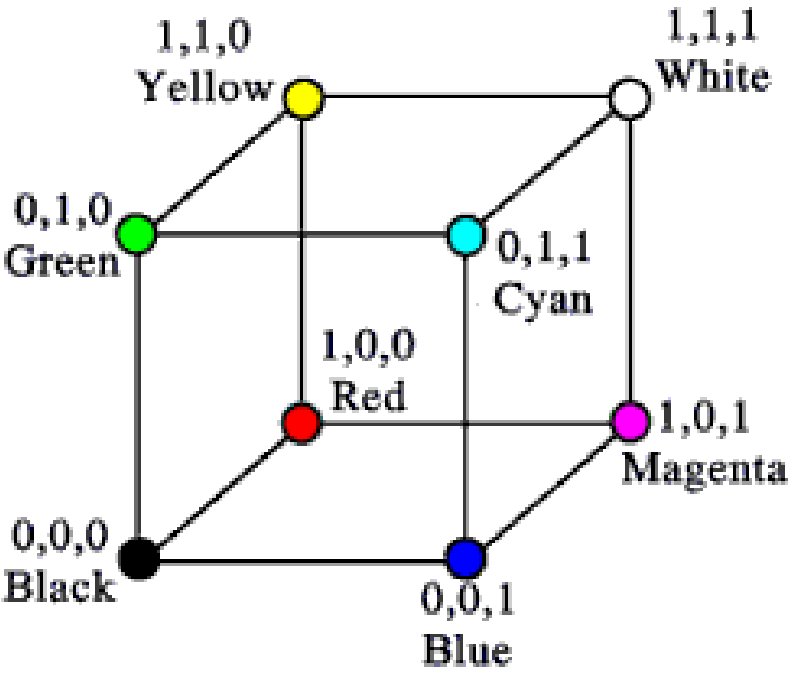
Una pista: el color

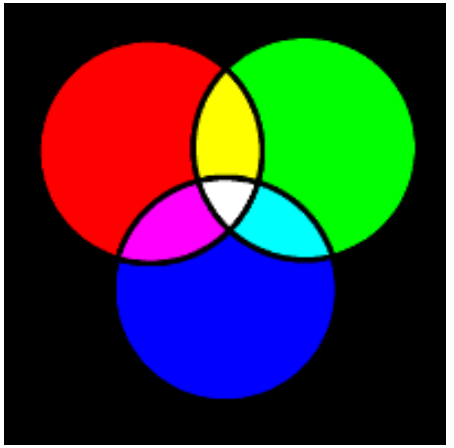
- Piso sin color (blanco)
- Límite del piso con color
- Latas sin color
  - Negras con reflejos del gris al blanco
- Obstáculos con color

Espacios de color

- El color es una característica tridimensional
- Diferentes formas de representarlo – espacios de color

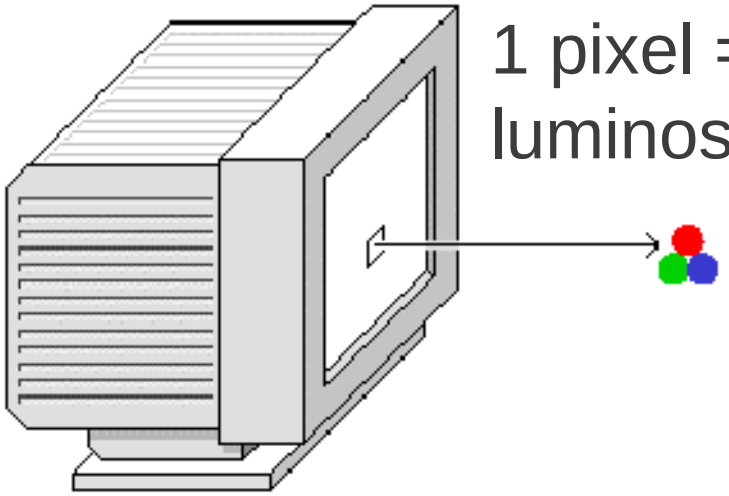
RGB



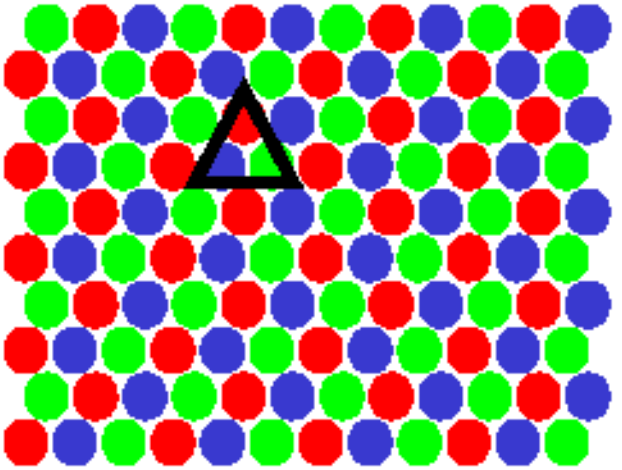


- Espacio aditivo
- Colores primario de luz

Monitor CRT

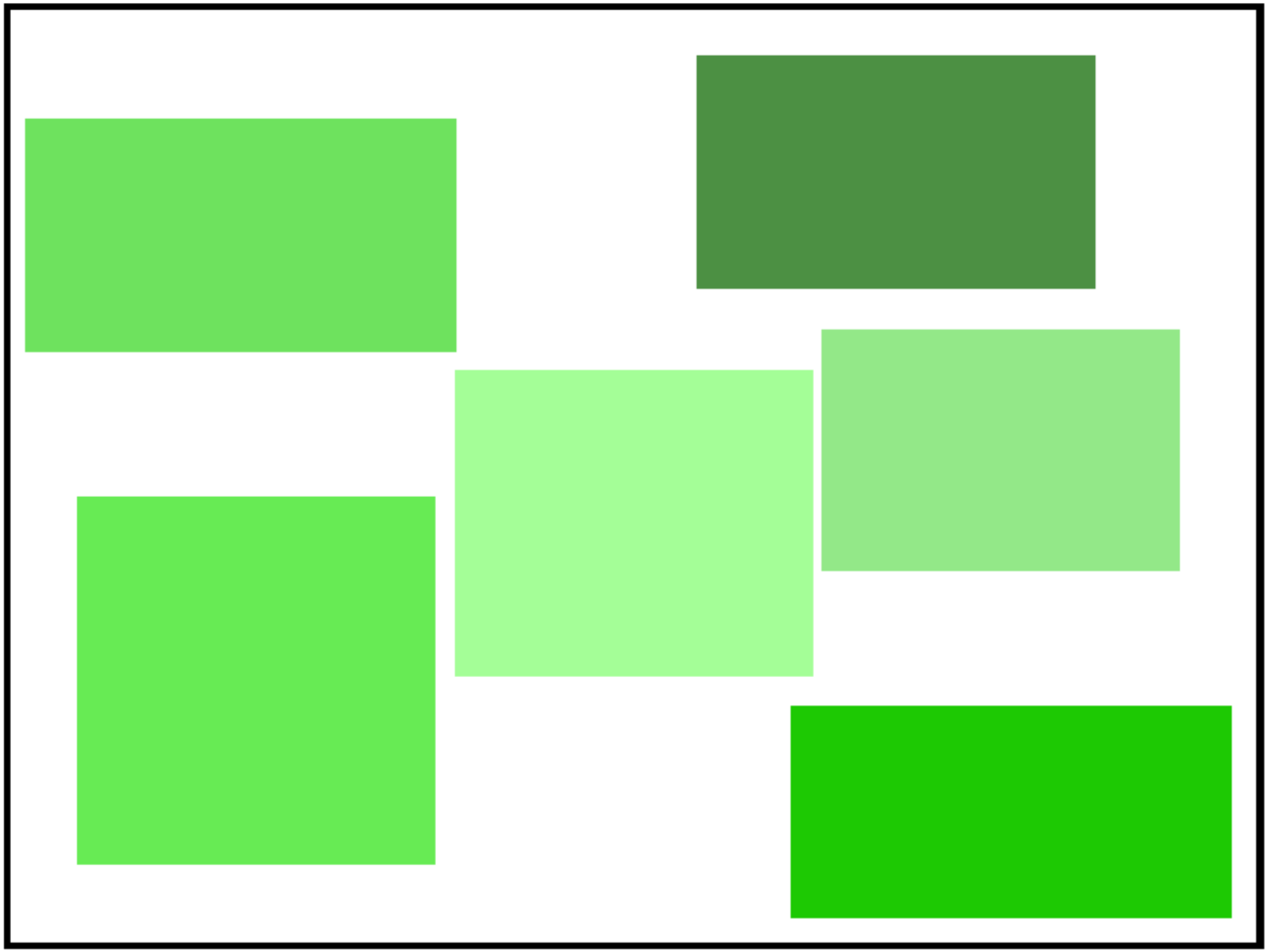


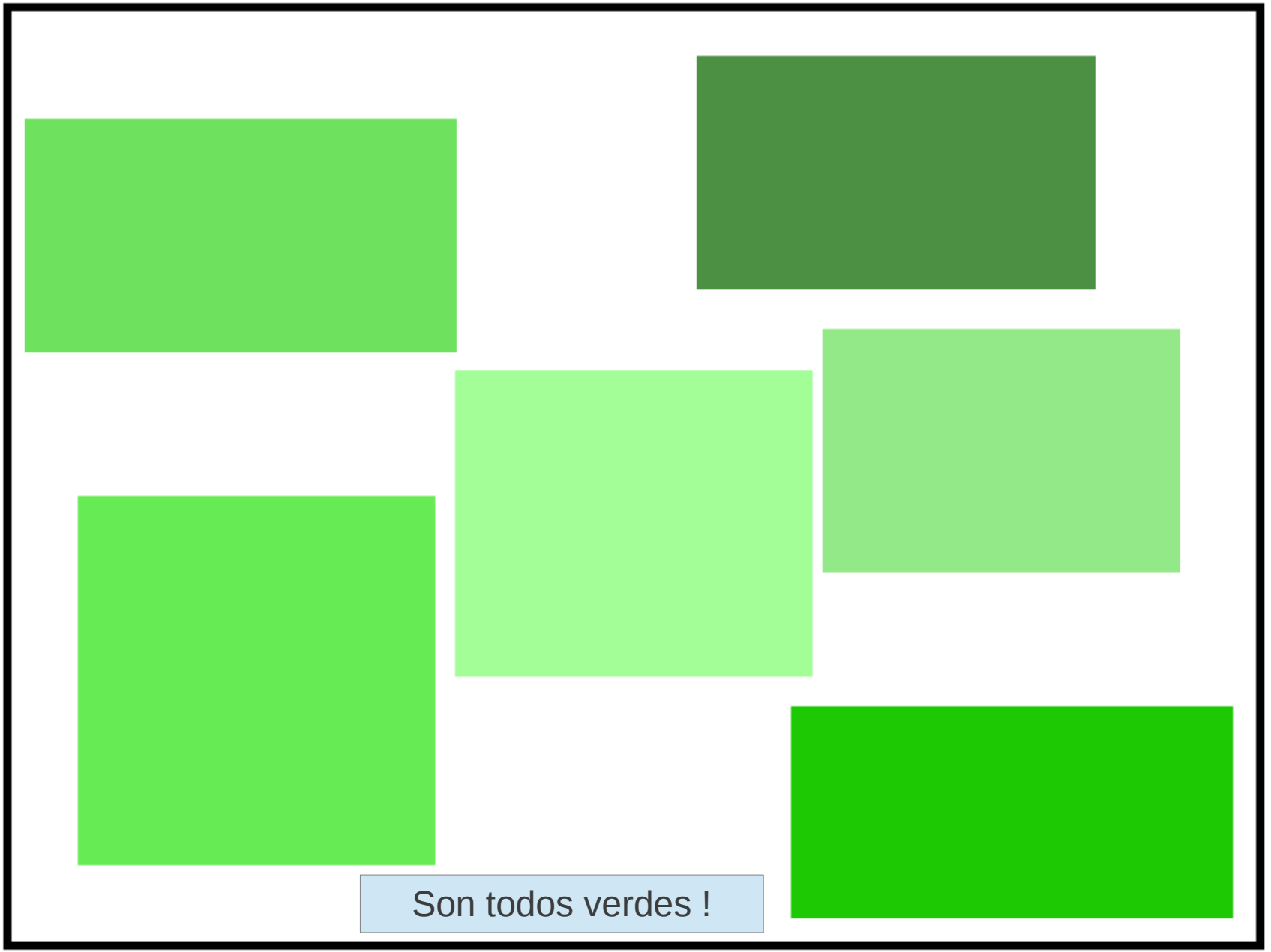
1 pixel = 3 fósforos (R,G,B)
luminosidad con intensidad variable



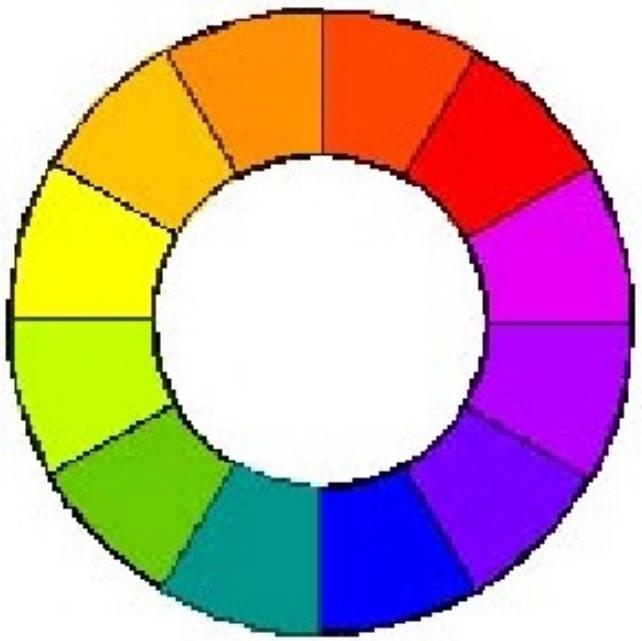
Pero...

- Nosotros no pensamos en términos de RGB
- ¿Qué tienen en común los siguientes colores?




Son todos verdes !

Hue (Tono)

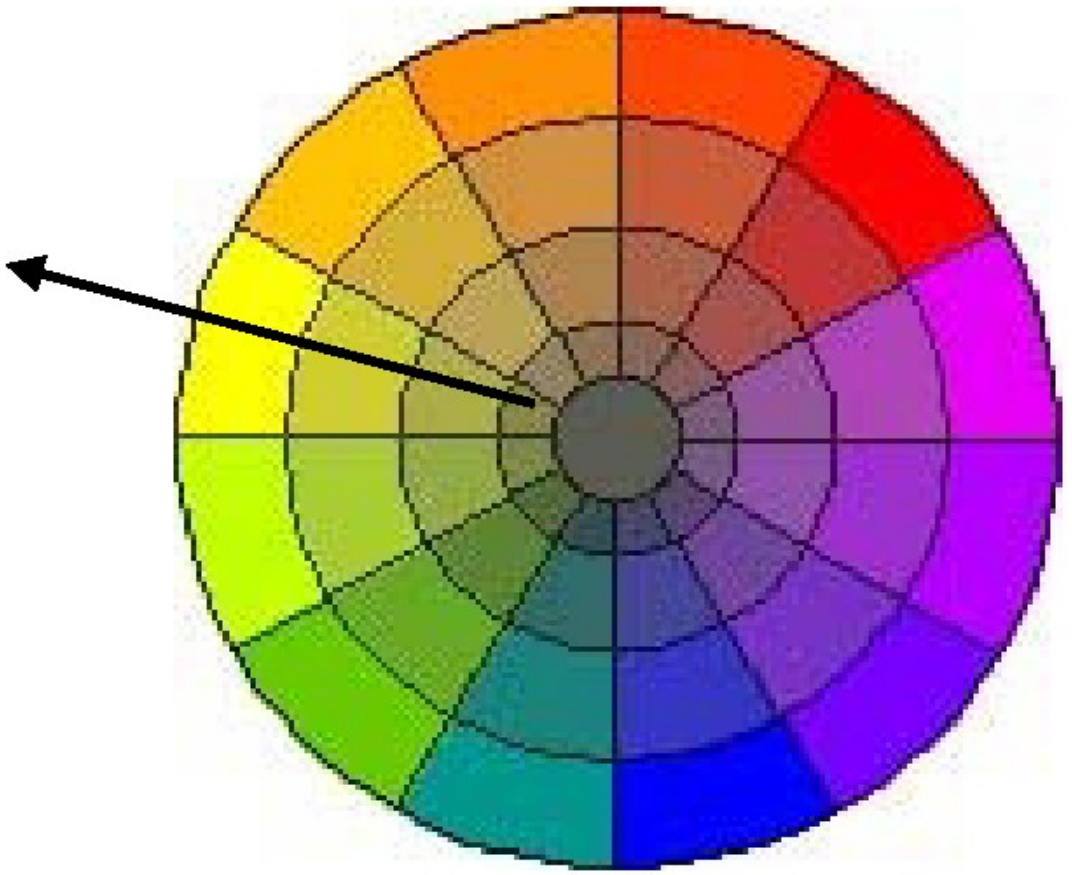


Itten color wheel

This is “less” green

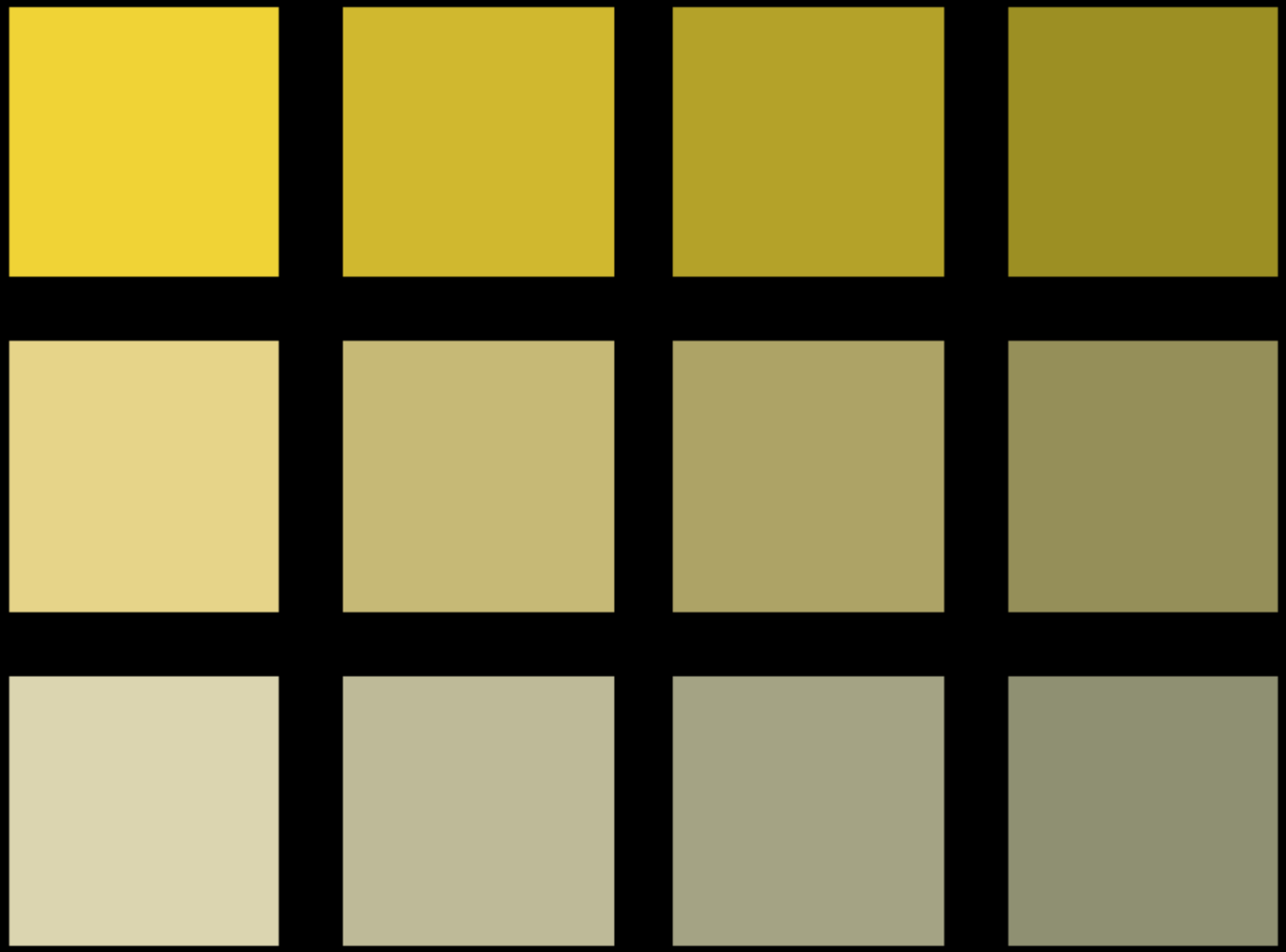
This is “more” green

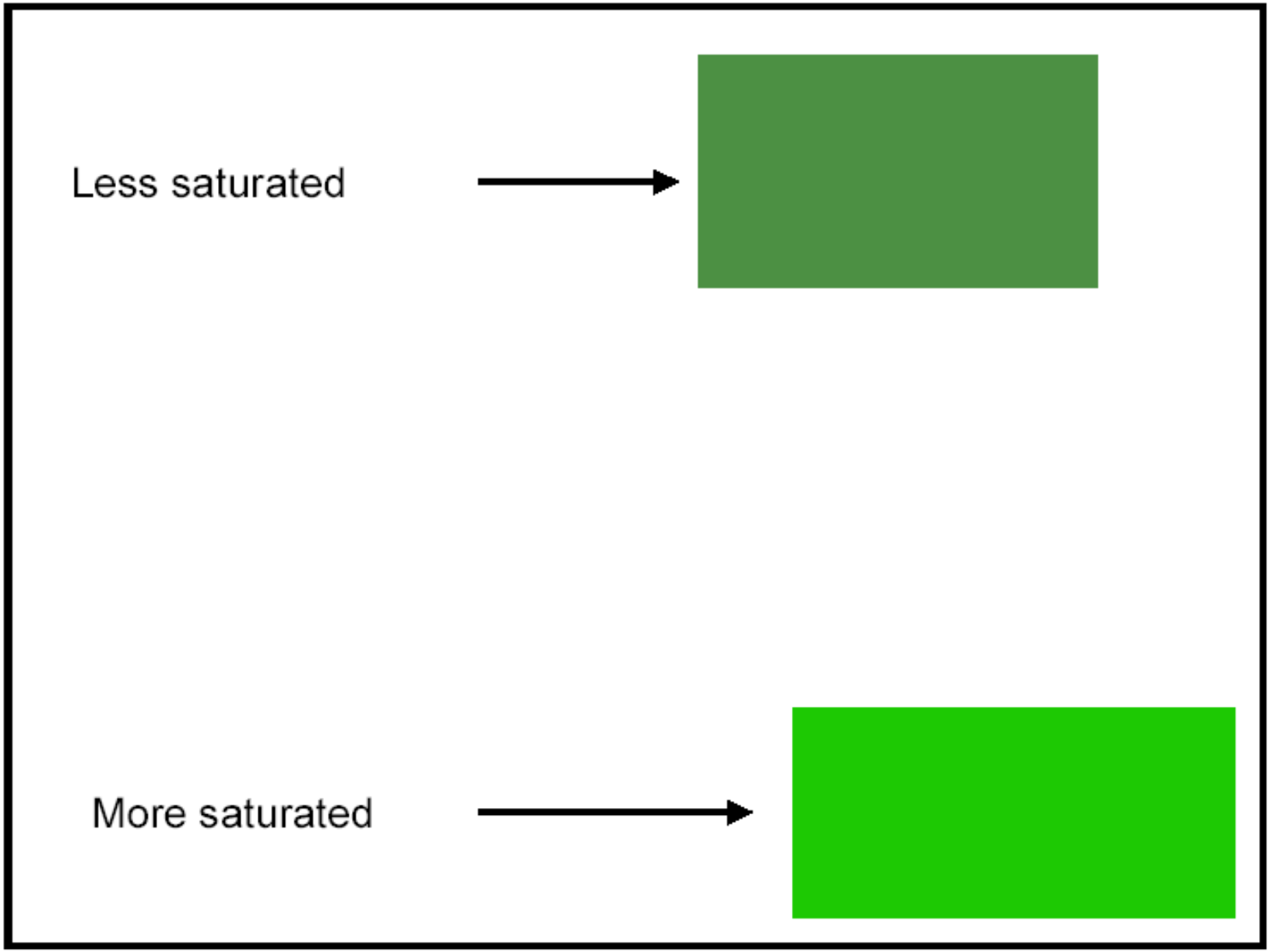
Saturación



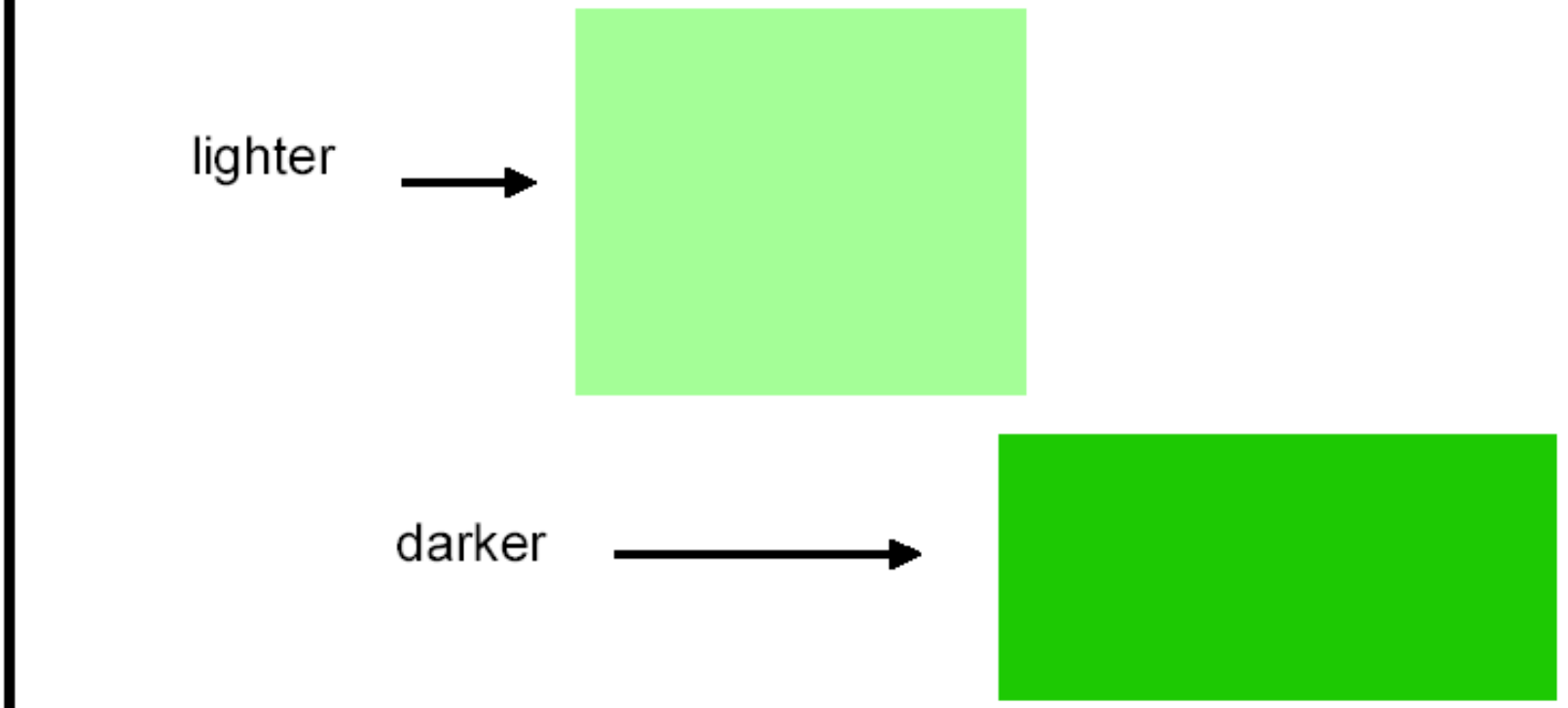
“Pureza”

Mismo tono, diferente saturación

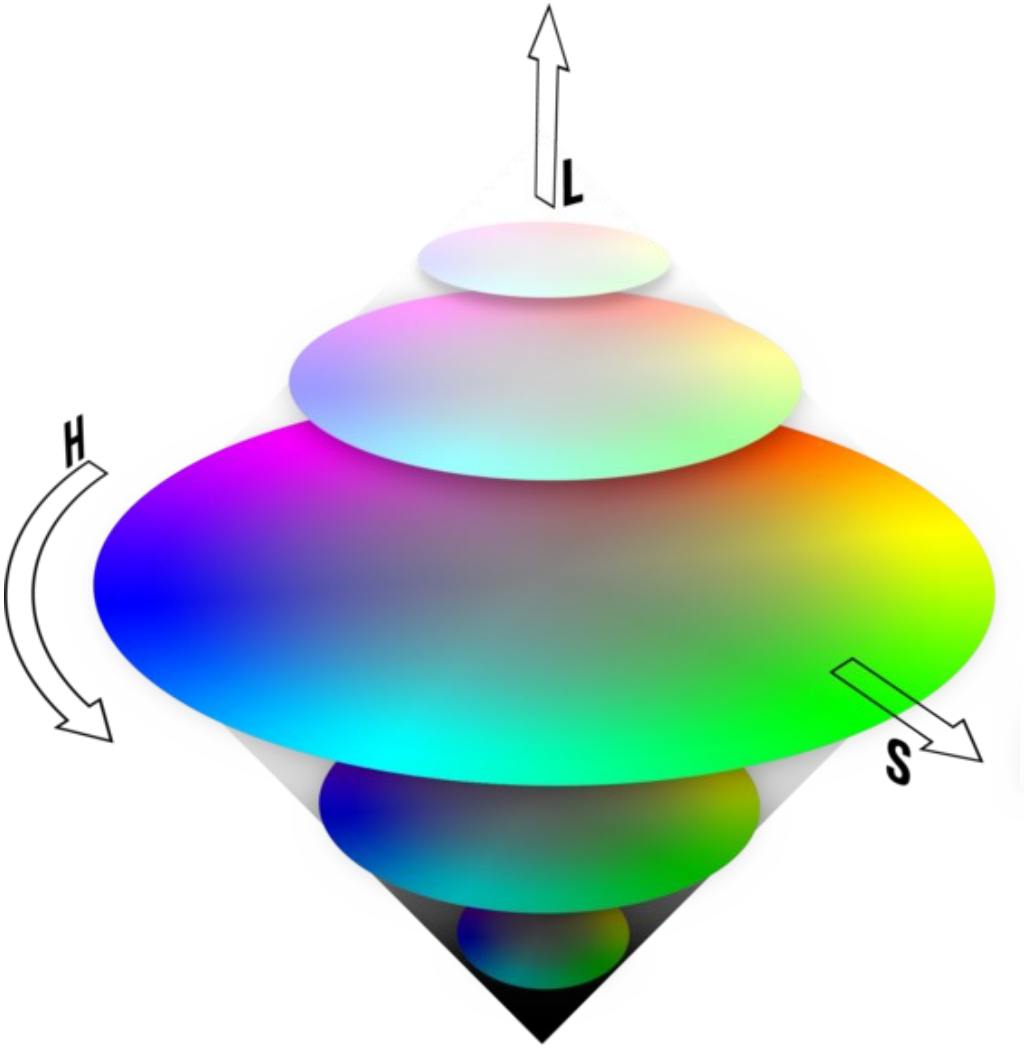



Less saturated
More saturated

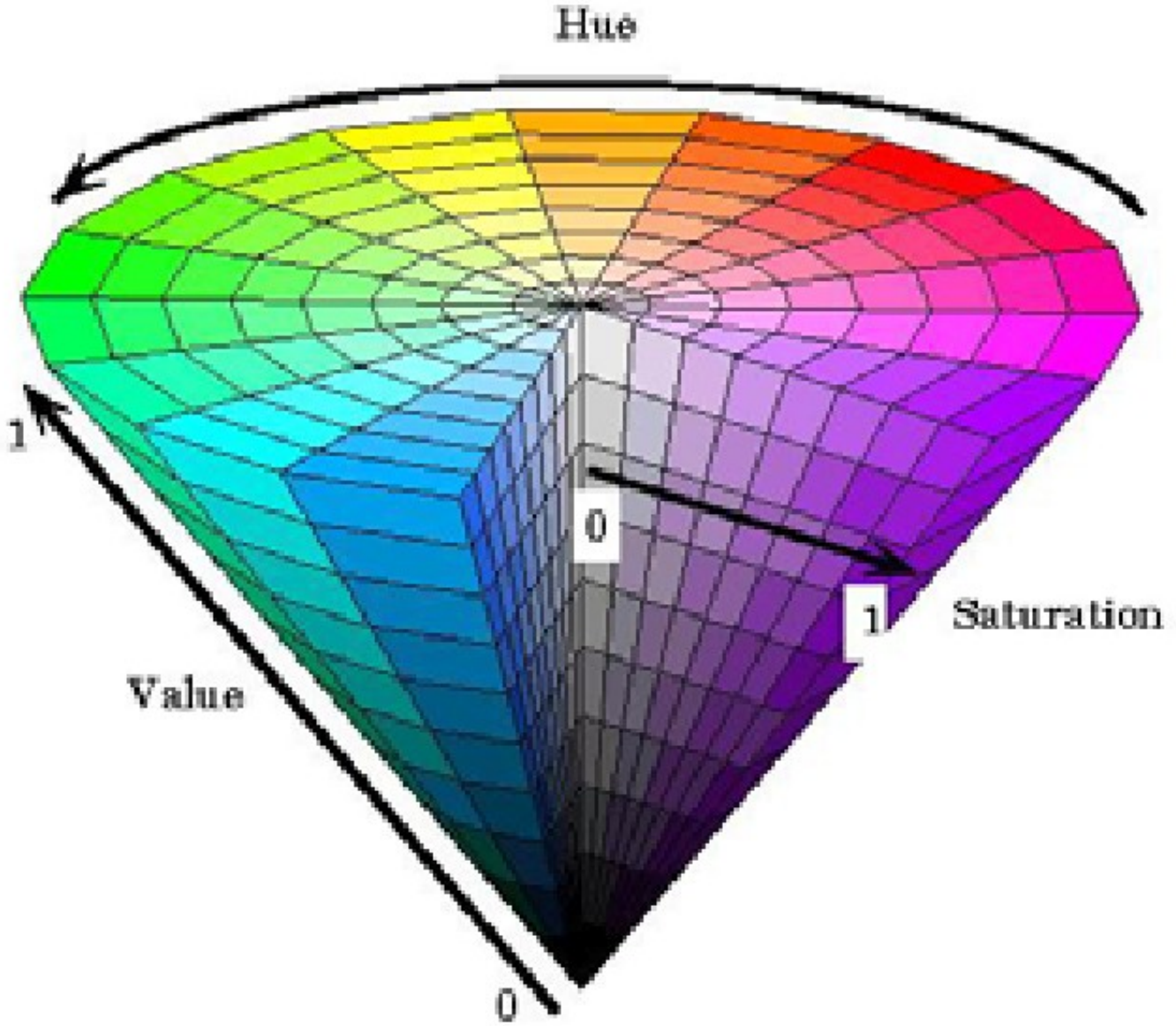
Igual saturación

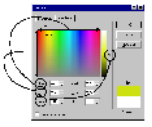


Luminosidad



HSV





Ventajas de espacios HSL o HSV

- En este espacio es más fácil
  - agrupar colores similares
  - separar los que es luminosidad (intensidad) de lo que es color (croma).
  - Ej. (ver eliminación de brillos):

Imagen original

Luminosidad

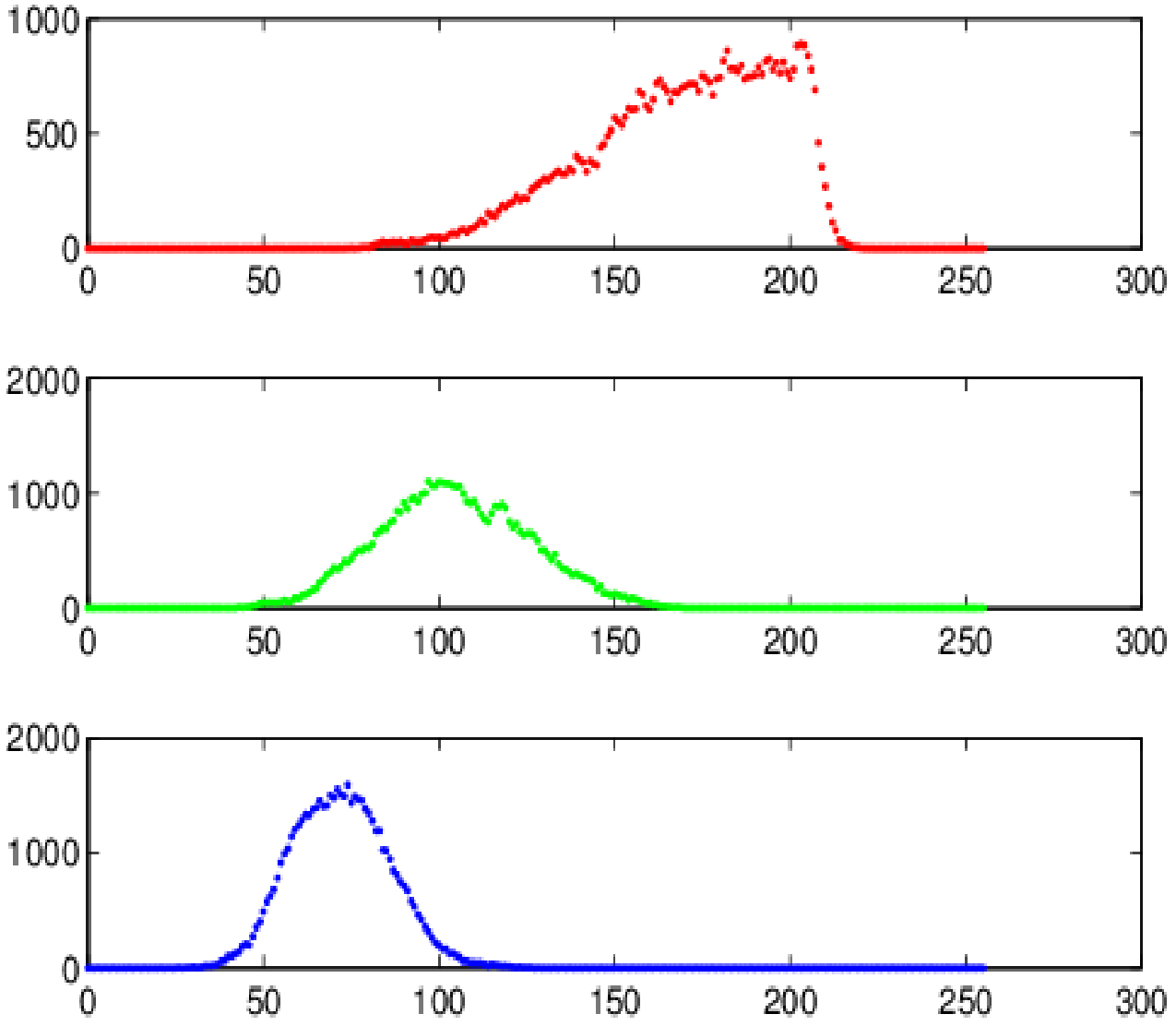
Color y saturación

Detección de colores

- Aprender las variaciones de color de los objetos que se quieren detectar
  - Esto define una región en el espacio de color elegido
- Definir criterios que permitan separar la región del espacio (ej. umbrales)
- Para cada pixel de una imagen nueva, clasificar a qué región del espacio de color pertenece
  - Es del color de los objetos buscados o no

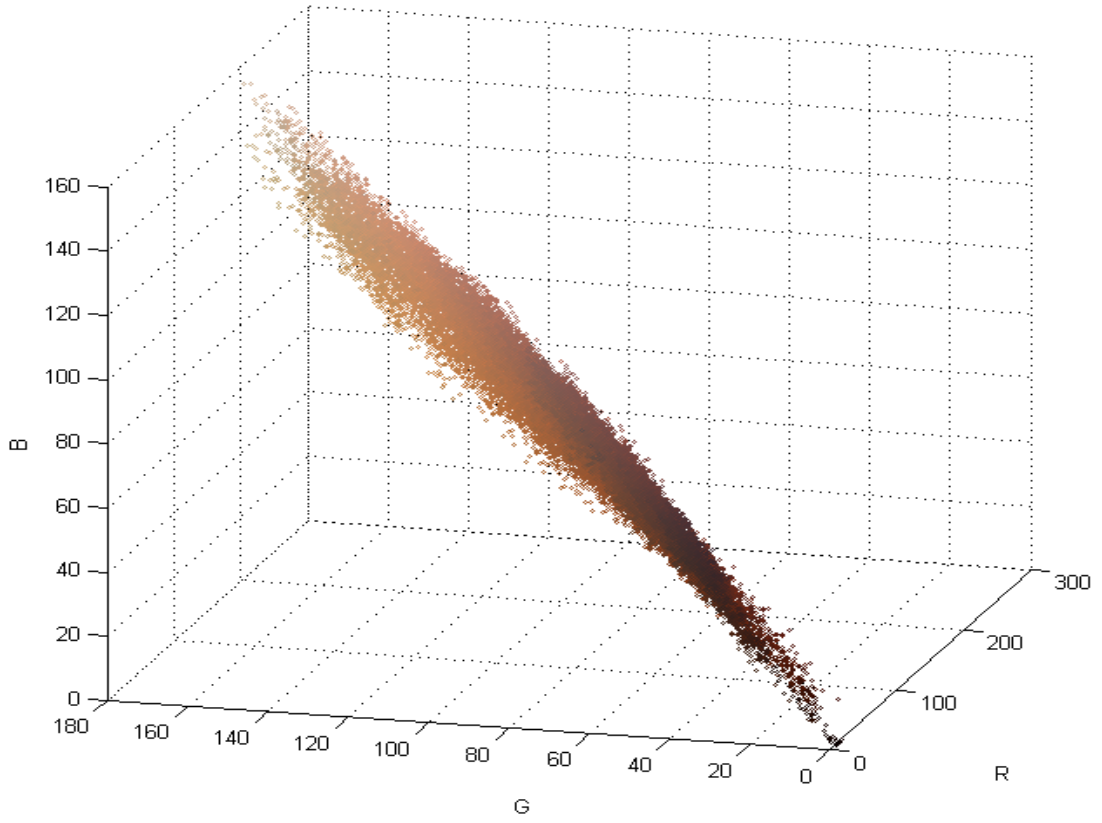
Ej. Detección de piel en RGB

Histogramas para una cara



Muchas caras

- Variabilidad de los tonos de piel
- Cluster de todos los pixeles de todas las caras de una base en el espacio RGB



Opciones para la clasificación

- Definir umbrales fijos. Ej.

  $(R > 95) \& (G > 40) \&(B > 20) \&$
  $(\max\{R, G, B\} - \min\{R, G, B\} > 15) \&$
  $(|R - G| > 15) \& (R > G) \&(R > B)$

- Aprender de la estadística del cluster
  - Ej. matriz de covarianza va a indicar las relaciones entre los valores R, G y B del cluster

Detección