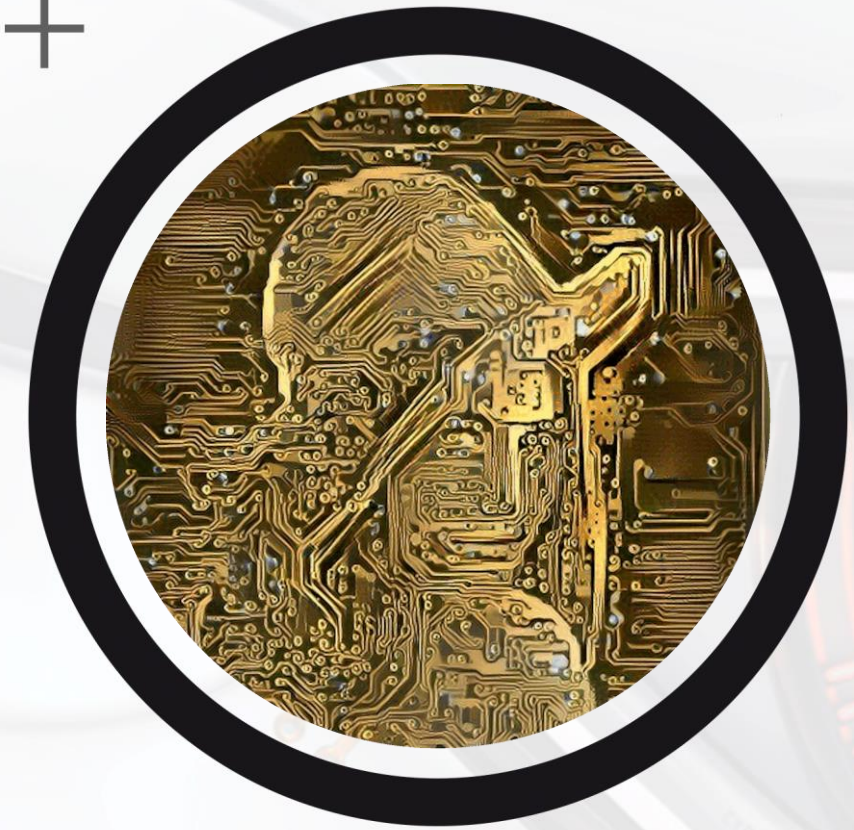


P3

Imagen y Vídeo por Computador

. . .

Práctica 3: Segmentación y descriptores

Vicente Morell
vicente.morell@ua.es

Ingeniería Multimedia

Universitat d'Alacant
Universidad de Alicante

HUMAN ROBOTICS

ÍNDICE

1. **Sesión 1: Segmentación por umbralización**
   - Segmentación
   - Umbralización
   - Umbralización global
   - Método de Otsu
   - Umbralización local por zonas
2. Sesión 2: Detección de formas: descriptores

Vicente Morell
vicente.morell@ua.es

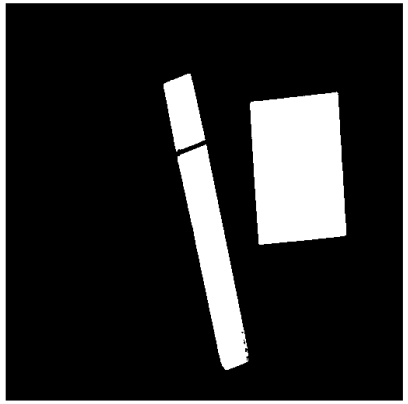
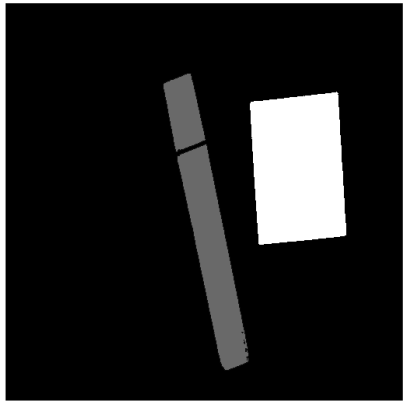
Ingeniería Multimedia

Universitat d'Alacant
Universidad de Alicante

HUMAN ROBOTICS

Segmentación

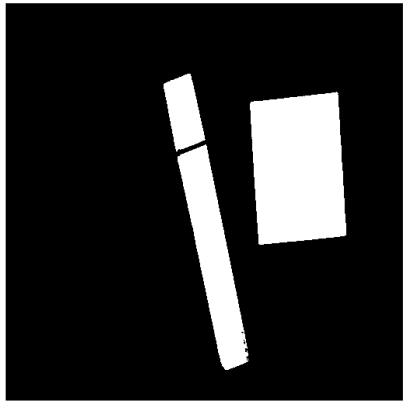
- La segmentación consiste en separar o dividir una imagen en regiones u objetos.
- Es necesaria para facilitar el reconocimiento (se verá en el segundo bloque de esta práctica).
- También facilita el procesamiento de la imagen en regiones de interés (ROI).
- Retos de la segmentación:
  - La definición de una región / objeto es dependiente de la aplicación.
  - Es una de las tareas más complicadas en el Procesamiento de imágenes.
  - Su precisión determina el éxito o fracaso de la aplicación.

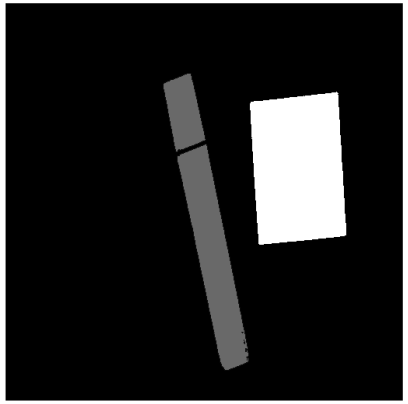




Vicente Morell vicente.morell@ua.es

Técnicas de segmentación

- Segmentación por detección de bordes.
  - Consiste en encontrar los contornos de las diferentes regiones en la imagen.
- Segmentación por umbralización.
  - Consiste en agrupar regiones que tienen niveles de gris similares.
- Segmentación por regiones.
  - Consiste en encontrar regiones directamente utilizando técnicas de crecimiento y división de áreas.





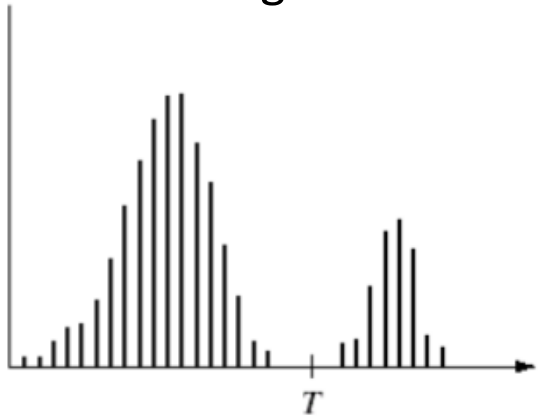
Vicente Morell vicente.morell@ua.es

Umbralización

- En una imagen compuesta de objetos claros sobre fondo oscuro los valores de intensidad se agrupan en el histograma en dos bloques distintos de distribución.
- El método más intuitivo es extraer los objetos (claros) del fondo (oscuro) seleccionando un **umbral**, T, que separe los dos bloques.
- Cualquier punto (x,y) en la imagen que cumpla f(x,y) > T se clasifica como **punto del objeto**, en caso contrario, se clasifica como **punto del fondo**.
- La imagen umbralizada de esta forma, g(x,y), será:

$$g(x,y) = \begin{cases} 1 & si\ f(x,y) > T \\ 0 & si\ f(x,y) \leq T \end{cases}$$

- T es una constante que se aplica sobre la imagen entera → **Umbralización Global**.

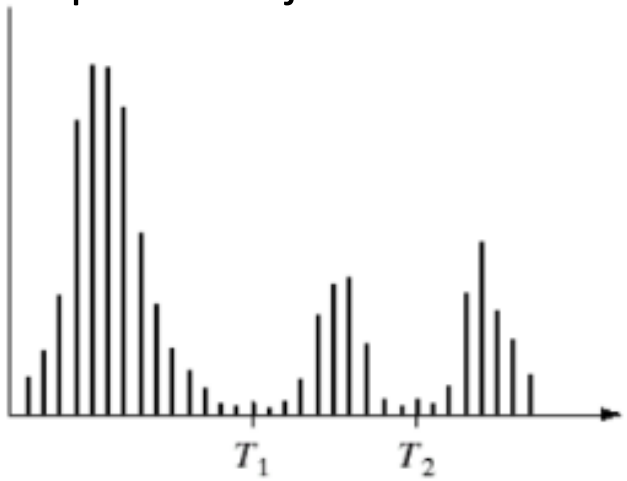


Vicente Morell vicente.morell@ua.es

Umbralización múltiple

- Un histograma con tres bloques dominantes distintos (dos objetos claros sobre un fondo oscuro) clasifica un punto (x,y) como de fondo si f(x,y) ≤ T1, como del primer objeto si T1 < f(x,y) ≤ T2, y como el segundo objeto si f(x,y) > T2.
- La imagen umbralizada de esta forma, g(x,y), será:

$$g(x,y) = \begin{cases} a & si\ f(x,y) > T_2 \\ b & si\ T_1 < f(x,y) \leq T_2 \\ c & si\ f(x,y) \leq T_1 \end{cases}$$



- Los problemas de segmentación que requieren **más de dos umbrales** son difíciles de resolver (a menudo imposible).
- Se obtienen mejores resultados utilizando otros métodos, como **umbralización variable** o **umbralización adaptativa**, donde el valor de T cambia sobre la imagen.

Vicente Morell vicente.morell@ua.es

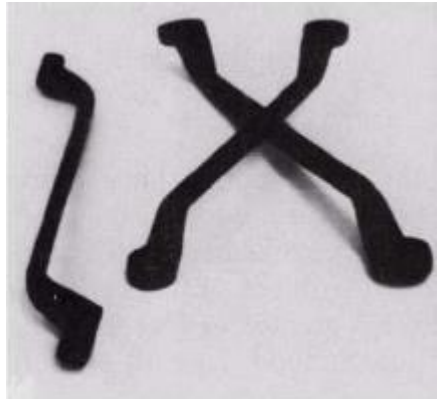
Umbralización global

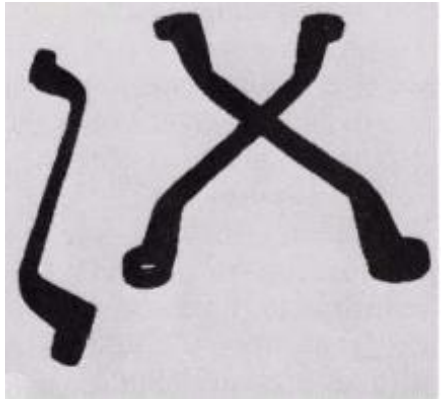
- De manera intuitiva, se puede afirmar que el éxito de la umbralización por intensidad está directamente relacionado con el **ancho y la profundidad de los valles** que separan los distintos bloques del histograma.
- ¿Se puede determinar el umbral, T, por **prueba-error**?
- Si se tienen muchas imágenes para procesar mediante umbralizado, ¿las vamos a procesar una a una mediante el método de prueba-error?
- La umbralización global puede ser suficiente cuando hay **suficiente variación en la imagen** (la distribución de los píxeles del objeto y del fondo son suficientemente distintos).
- Aunque el método de umbralización global sea una aproximación válida, es necesario que se estime de manera **automática** el umbral para cada imagen.

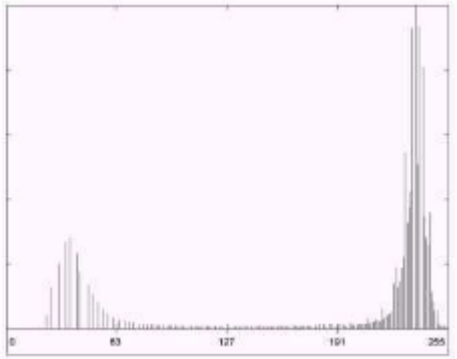
Vicente Morell vicente.morell@ua.es

Algoritmo Básico de Umbralización Global

- Pasos del **algoritmo** iterativo de **umbralización global**:
  1. Se selecciona una estimación inicial del umbral global, T.
  2. Se binariza la imagen usando T. Esto producirá dos grupos de pixels: G_1 formado por todos los pixels con valores de intensidad mayores que T, y G_2 formado por los pixels con valores menores o iguales que T.
  3. Calcular la media de los valores de intensidad m_1 y m_2 para los pixels de G_1 y G_2, respectivamente.
  4. Calcular un nuevo umbral: $T=0.5*(m_1+m_2)$.
  5. Repetir los pasos 2 a 4 hasta que la diferencia entre los valores de T en sucesivas iteraciones sea menor que un parámetro predefinido ΔT.







Vicente Morell vicente.morell@ua.es

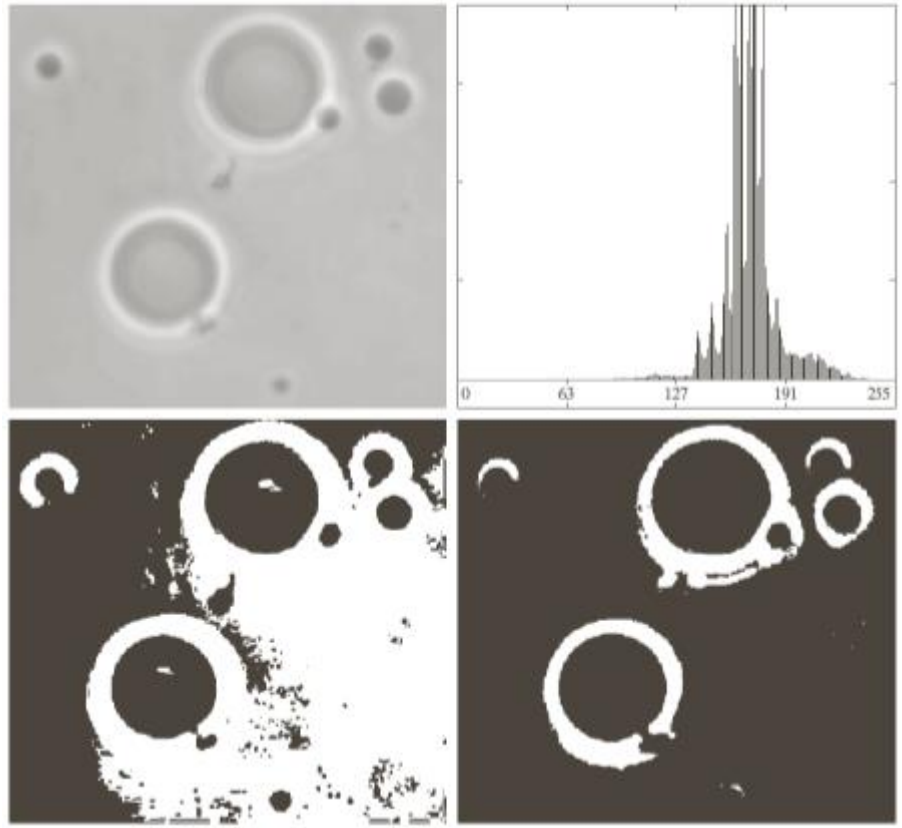
Umbralización global

- Este simple algoritmo **trabaja bien** en situaciones donde hay un valle relativamente claro entre los distintos picos del histograma.
- El **parámetro ΔT** se usa para controlar el número de iteraciones cuando la velocidad es un aspecto importante:
  - Cuanto más grande sea ΔT menos iteraciones necesitará el algoritmo y por lo tanto, la respuesta será **más rápida**.
  - Cuanto más pequeño sea ΔT más iteraciones necesitará el algoritmo y por lo tanto, la respuesta será **más lenta**.

Vicente Morell vicente.morell@ua.es

Método de umbralización de Otsu

- Maximiza la varianza entre objetos.
- La idea básica es que los objetos bien umbralizados deben diferenciarse en los valores de intensidad de sus pixels.
- El umbral que proporcione la mejor separación entre objetos en términos de valores de intensidad será el mejor umbral (más óptimo).
- El histograma de la imagen crea una función de probabilidad de densidad (PDF) de los niveles de intensidad de cada objeto.
- Una medida de separación se utiliza para medir la relación entre la varianza entre objetos y la varianza global.



Vicente Morell vicente.morell@ua.es

Método de umbralización de Otsu

- Pasos del método de umbralización de Otsu (resumido):
  1. Calcular el histograma normalizado de la imagen.
  2. Calcular el sumatorio de todos los valores de intensidad k.
  3. Calcular el sumatorio de las medias para todos los valores de intensidad k.
  4. Calcular la media de la intensidad global.
  5. Calcular la varianza global, y la varianza entre clases para todos los k.
  6. Obtener el umbral de Otsu, k*, como el valor de k para el que la varianza entre clases es máxima.
  7. Obtener la medida de división o separación, η*.

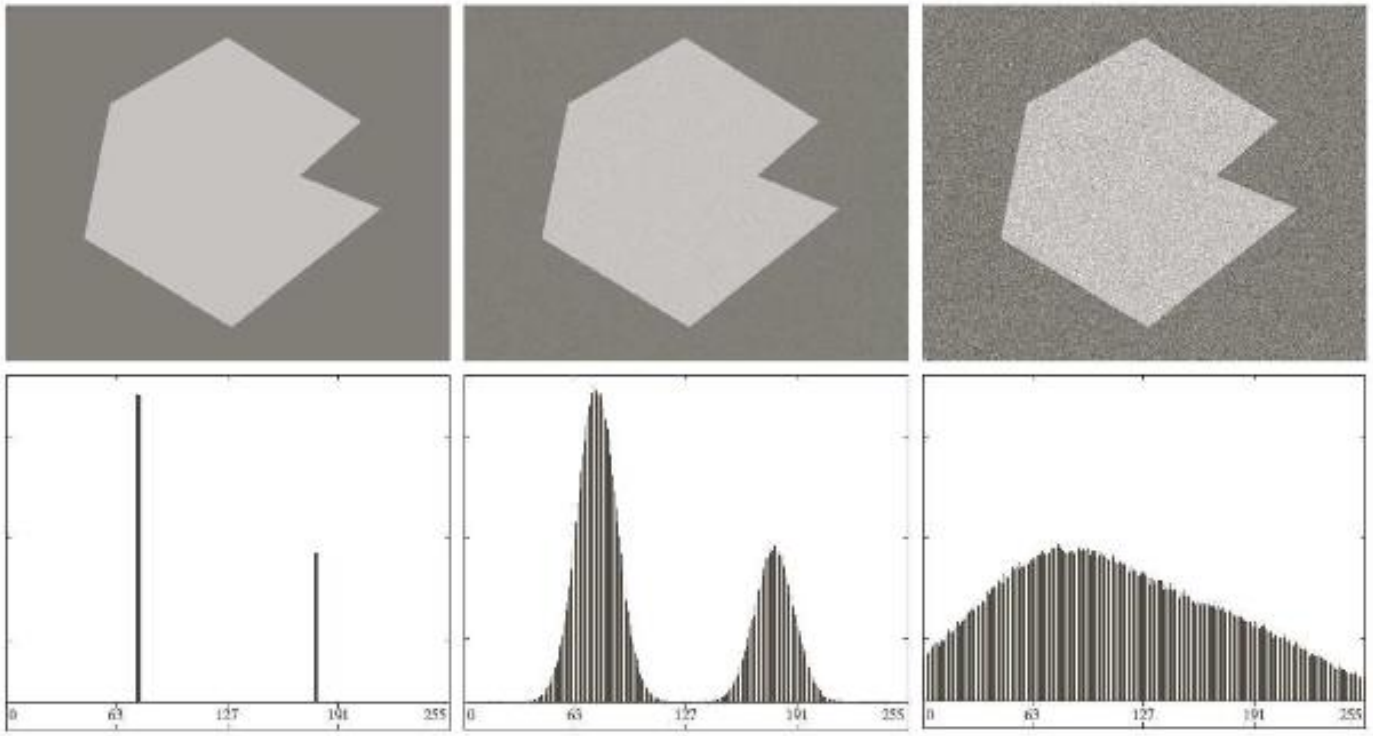
```
>>  I =
imread('mano_ua_gris.jpg');
>> T = graythresh(I);
```

```
>> help graythresh;
```


Vicente Morell vicente.morell@ua.es

Efecto del ruido

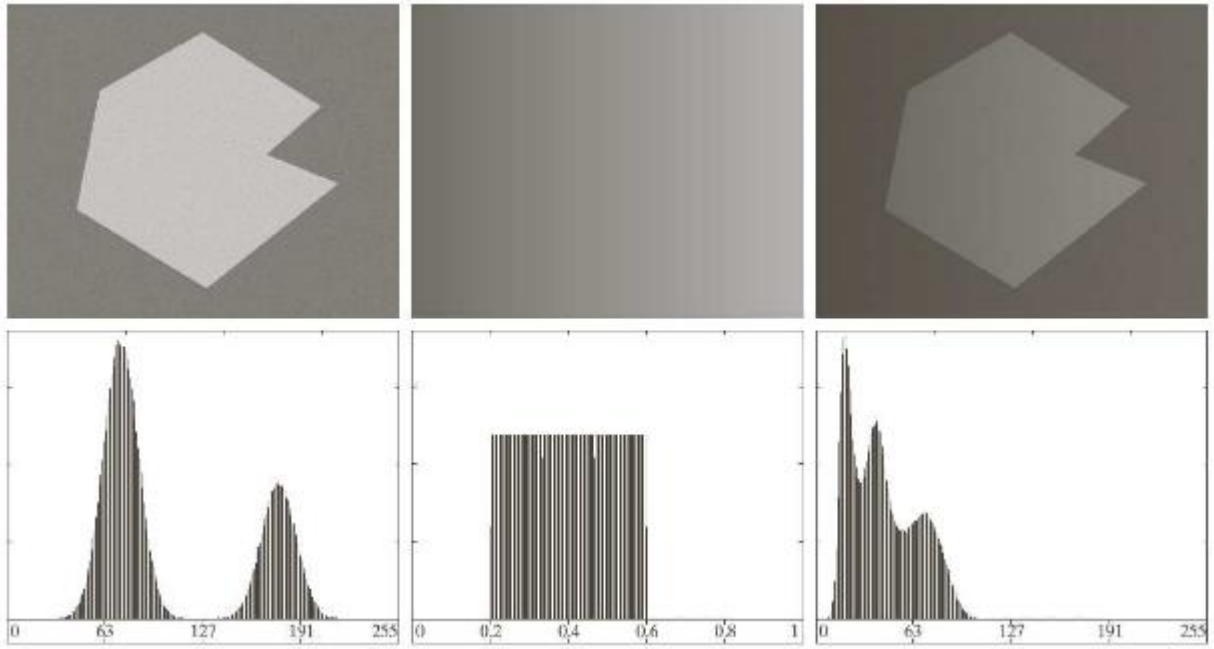
- Añadiendo ruido Gaussiano a la imagen es **imposible diferenciar** entre las dos zonas diferenciadas de objetos del histograma.
- Solución: **preprocesamiento** de la imagen. Utilización de filtros para mejorar la imagen antes de obtener el umbral.



Vicente Morell vicente.morell@ua.es

Efecto de la iluminación no uniforme

- La iluminación no uniforme causa que la zona de valle entre los picos del histograma se pierda de tal manera que sea **imposible realizar la separación** de los dos bloques de objeto y fondo.
- Solución: **Umbralización local por zonas**.



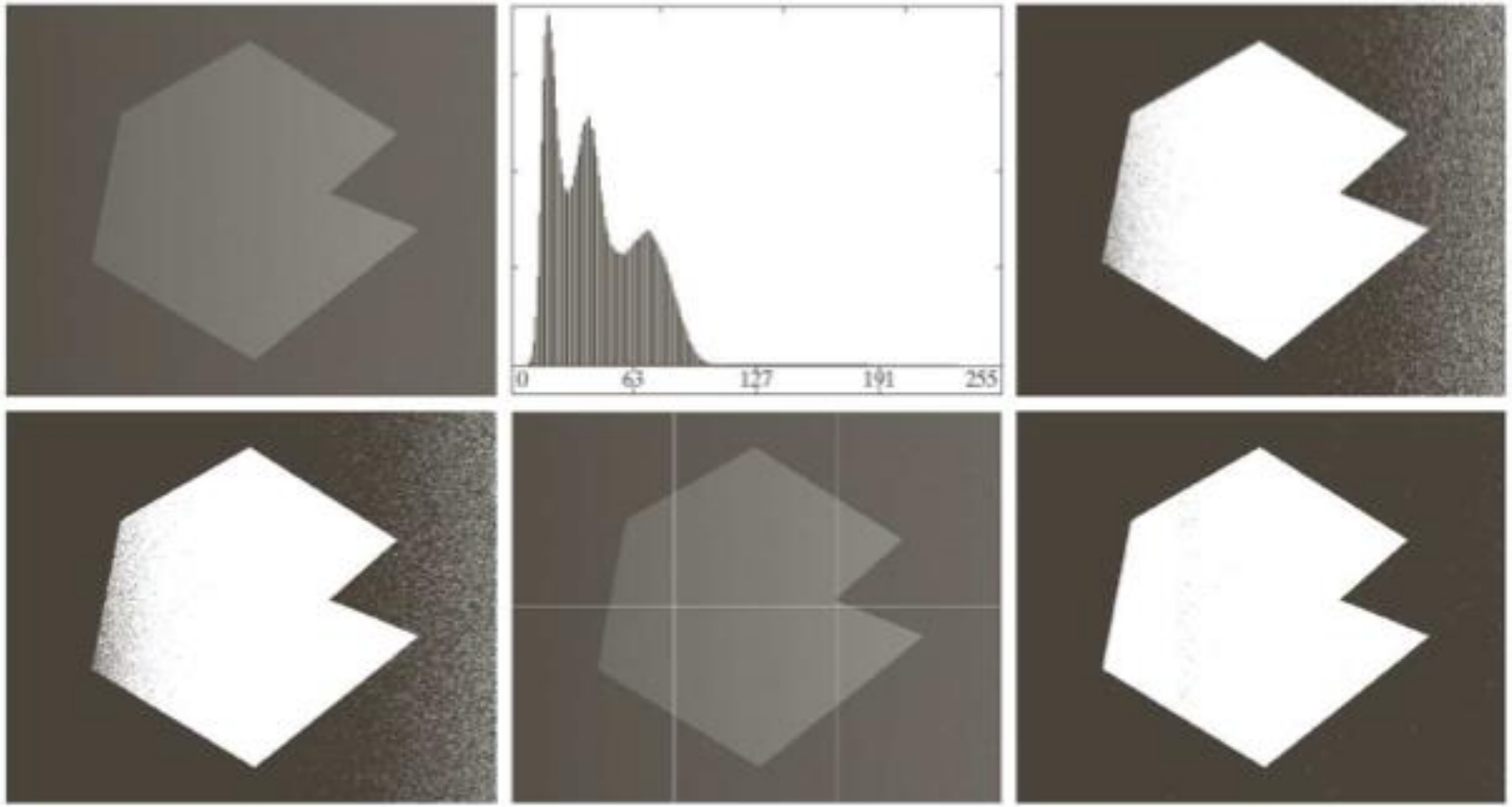
Vicente Morell vicente.morell@ua.es

Umbralización local por zonas

- En ciertos casos, los métodos de **preprocesamiento** pueden no ser prácticos o simplemente inútiles para mejorar la precisión de la umbralización.
- En estos casos, se pueden emplear técnicas de **umbralización variable o adaptativa**.
- Los valores de umbral se pueden cambiar o variar de acuerdo a zonas locales de la imagen.
- Una de las aproximaciones más simples de esta técnica es la **umbralización local por zonas**.
- ¿Cómo proceder?
  - Dividir la imagen en **subimágenes**.
  - Calcular el **histograma** de cada subimagen.
  - Seleccionar un **umbral** para el histograma de cada subimagen.
  - **Aplicar el umbral** a cada subimagen.
  - Proceso de **agrupación** para homogenizado de objetos o partes del objeto (para que el objeto aparezca con el mismo valor de intensidad asignado). Este proceso de homogeneizado se realiza seleccionando un umbral para cada píxel obtenido a partir de la interpolación de las subimágenes.

Vicente Morell vicente.morell@ua.es

Umbralización local por zonas



Vicente Morell vicente.morell@ua.es

ÍNDICE

1. Sesión 1: Segmentación por umbralización
2. **Sesión 2: Detección de formas: descriptores**
    1. Descripción de una región
    2. Obtención de objetos
    3. La función regionprops
    4. Selección de objetos

Vicente Morell
vicente.morell@ua.es

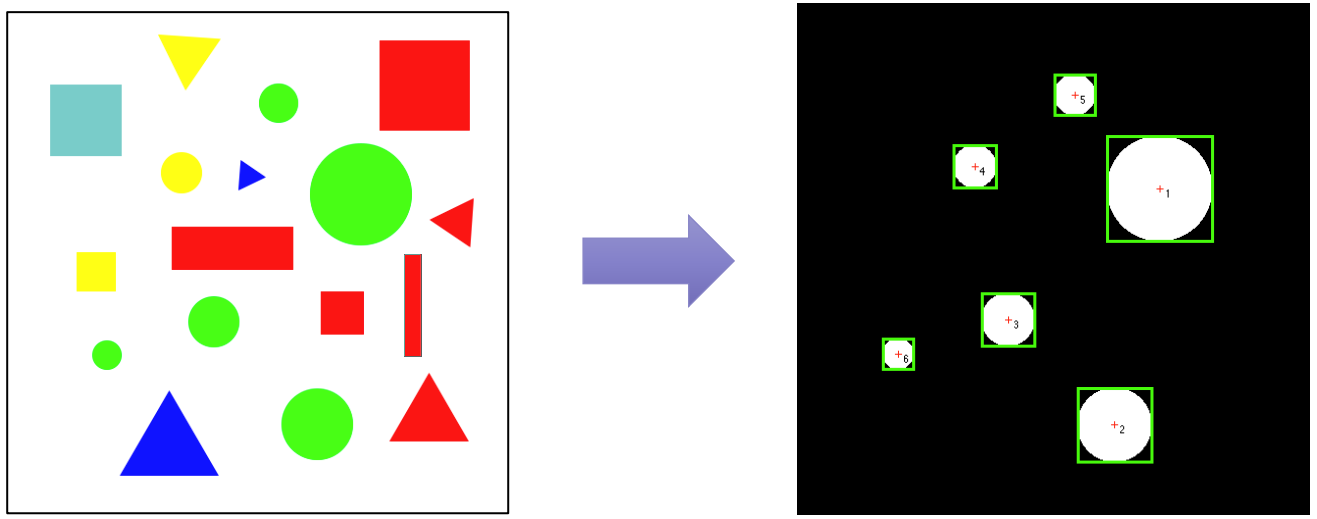
Ingeniería Multimedia

Universitat d'Alacant
Universidad de Alicante

HUMAN ROBOTICS

Reconocimiento de objetos

- **Motivación** del reconocimiento de objetos:
  - Identificar un **objeto** específico.
  - Identificar una **clase de objeto** (avión, coche, cara, bicicleta, etc.).
  - Determinar la **localización de un objeto** (múltiples instancias en una misma imagen).
  - **Separar unos objetos de otros** en la imagen (determinar y reconocer cada zona de una segmentación).

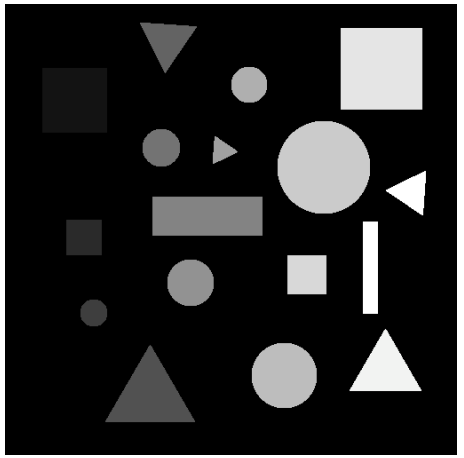


Vicente Morell vicente.morell@ua.es

Obtención de objetos en una imagen

- La función **bwconncomp** permite obtener los objetos o componentes conexas encontrados en una imagen binaria.
- También se pueden utilizar para encontrar componentes conexas las funciones **bwlabel** y **bwlabeln**.
- Sin embargo, el uso de bwconncomp sustituye el uso de bwlabel y bwlabeln debido al **uso más eficiente de la memoria**.
- Mediante **labelmatrix** se puede obtener una imagen donde cada píxel del primer objeto encontrado tendrá valor 1 de intensidad, los pixels del segundo objeto tendrán valor 2 de intensidad y así sucesivamente.

```
>> cc = bwconncomp(bw);
>> L = labelmatrix(cc);
>> imshow(imadjust(L));
```



Vicente Morell vicente.morell@ua.es

La función regionprops

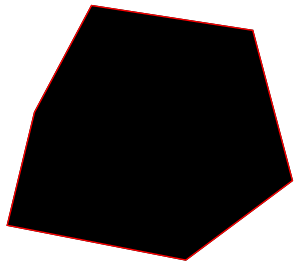
- La función **regionprops** permite obtener información de los objetos encontrados en una imagen binaria.
- Se puede utilizar sobre las siguientes variables:
  - Una **imagen binaria**.
  - Una estructura de componentes conexas devuelta por **bwconncomp**.
  - Una matriz **labelmatrix** con información de los distintos objetos encontrados en una imagen.
- La función devuelve un vector con la **información** requerida de los objetos o regiones de la imagen. El vector tendrá tantas componentes como objetos había en la imagen binaria, la estructura de componentes conexas o la matriz labelmatrix.
- Mediante **properties** se pueden indicar los distintos descriptores que se quieren obtener de cada objeto:
  - Con '**all**' se obtienen todos los descriptores. Computacionalmente es el más costoso.
  - Con '**basic**' (por defecto) se obtienen sólo los descriptores de 'Area', 'Centroid', and 'BoundingBox'.
  - También se pueden indicar **sólo los descriptores que realmente se necesiten**, evitando consumir más recursos de lo necesario para nuestra aplicación. Para ello, se pueden indicar los distintos descriptores separados por comas.

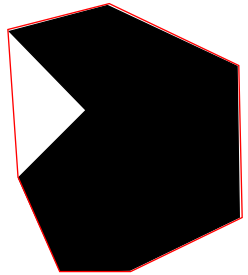
```
>> s=regionprops(bw,'Area','Perimeter');
```


Vicente Morell vicente.morell@ua.es

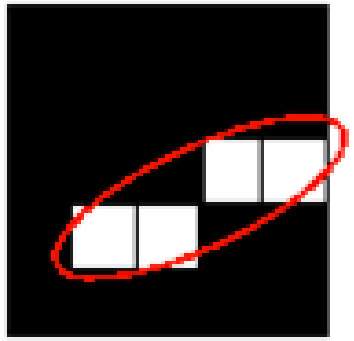
Los descriptores en regionprops

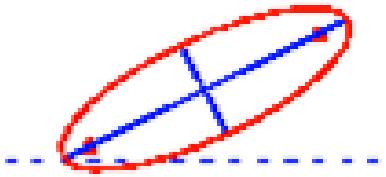
- '**Area**': número total de pixels del objeto.
- '**BoundingBox**': El rectángulo más pequeño que contiene al objeto.
- '**Centroid**': Especifica el centro de gravedad del objeto.
- '**ConvexHull**': Matriz de px2 que almacena el polígono convexo más pequeño que contiene al objeto. Cada una de las p filas contiene las coordenadas (x,y) de un vértice del polígono.





- '**ConvexImage**': Imagen binaria con todos los pixels del convex hull a 1.
- '**ConvexArea**': Número de pixels en 'ConvexImage'.
- '**Eccentricity**': Escalar que especifica la excentricidad de la elipse que tiene los mismos momentos de segundo orden que el objeto. La excentricidad vale 0 para un círculo y 1 para una línea.
- '**EquivDiameter**': Escalar que indica el diámetro de un círculo con la misma área que el objeto. Se calcula como sqrt(4*Area/pi).



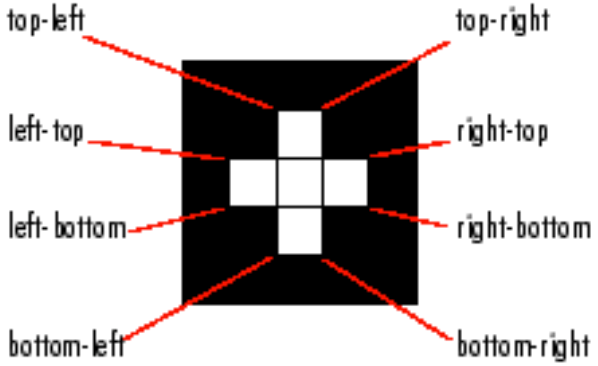


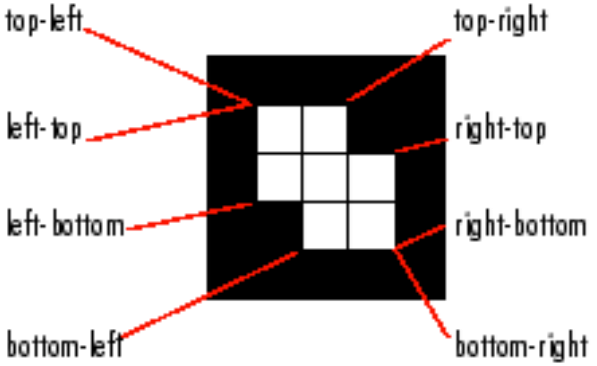
- '**EulerNumber**': Escalar que indica el número de objetos en la región menos el número de agujeros en esos objetos.

Vicente Morell vicente.morell@ua.es

Los descriptores en regionprops

- '**Extent**': Escalar que indica la relación de pixels en la región y pixels en el bounding box. Se calcula como el Area dividido entre el área del bounding box.
- '**Extrema**': Una matriz de 8x2 que contiene los puntos extremo en una función. Cada fila de la matriz contiene las coordenadas x,y de uno de los puntos. El formato del vector es [top-left top-right right-top right-bottom bottom-right bottom-left left-bottom left-top].



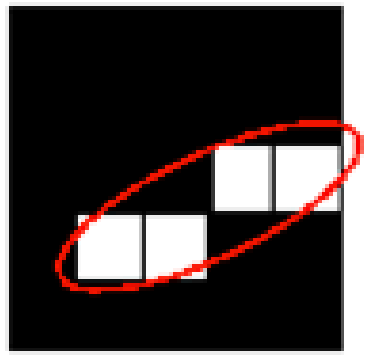


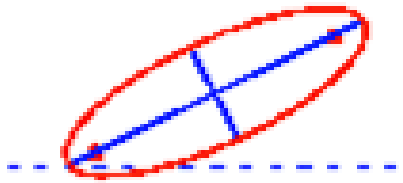
- '**FilledArea**': Escalar que indica el número de pixels en FilledImage.
- '**FilledImage**': Imagen binaria del mismo tamaño que el bounding box del objeto. Los píxeles a 1 se corresponden con la región del objeto, con todos los agujeros rellenos.

Vicente Morell vicente.morell@ua.es

Los descriptores en regionprops

- '**MajorAxisLength**': Escalar que indica la longitud (en pixels) del eje mayor de la elipse que tiene los mismos momentos de segundo orden normalizados que el objeto.
- '**MaxIntensity**': Escalar que especifica el valor del pixel con mayor valor de intensidad del objeto.
- '**MeanIntensity**': Escalar que indica la media de todos los valores de intensidad del objeto.
- '**MinIntensity**': Escalar que especifica el valor del pixel con menor valor de intensidad del objeto.
- '**MinorAxisLength**': Escalar que indica la longitud (en pixels) del eje menor de la elipse que tiene los mismos momentos de segundo orden normalizados que el objeto.
- '**Orientation**': Ángulo (en grados desde -90 a 90 grados) entre el eje X y el eje mayor de la elipse que tiene los mismos momentos de segundo orden normalizados que el objeto.





Vicente Morell vicente.morell@ua.es

Los descriptores en regionprops

- '**Perimeter**': Perímetro del objeto.
- '**PixelIdxList**': Vector de p elementos que contiene los índices lineales de los pixels de la región.
- '**PixelList**': Matriz de p filas donde cada fila indica las coordenadas x,y de un pixel del objeto.
- '**PixelValues**': Vector de p elementos, donde cada elemento del vector contiene el valor de intensidad de un pixel del objeto.
- '**Solidity**': Escalar que especifica la proporción de pixels en el convex hull que también están en el objeto.Se calcula como Area/ConvexArea.
- '**WeightedCentroid**': Centro de masas ponderado a partir de la posición de los pixels del objeto y el valor de intensidad de cada pixel del objeto.

Vicente Morell vicente.morell@ua.es

Los descriptores en regionprops

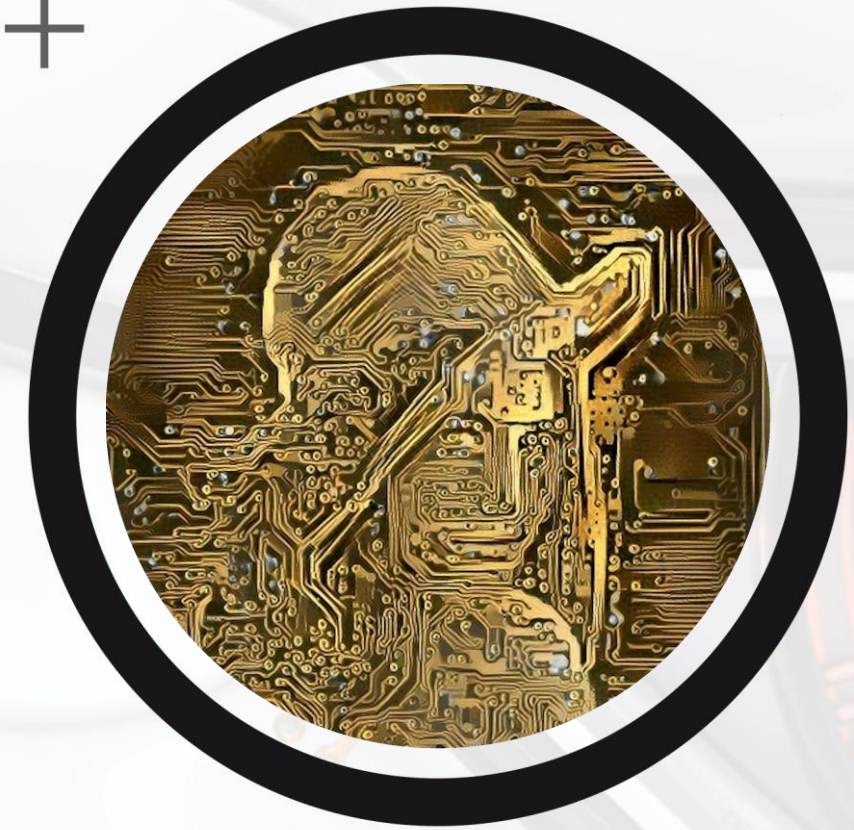
- El uso de la función **ismember** permite seleccionar objetos en función de ciertos criterios.
- Se puede usar para la **selección** de uno de los descriptores vistos para regionprops o se pueden **crear nuevos descriptores** utilizando fórmulas sobre los descriptores que proporciona regionprops.
- Por ejemplo, para **seleccionar círculos** independientemente de la escala, se puede utilizar la siguiente ecuación:

$$circularidad=(4*pi*Area)/Perimeter^2$$

- Un círculo tendrá un valor de circularidad próximo a 1.
- El siguiente ejemplo selecciona los objetos que tienen más de 80 pixels en la imagen:

```
>> cc = bwconncomp(BW);
>> stats = regionprops(cc, 'Area');
>> a = [stats.Area];
>> idx = find(a > 80);
>> BW2 = ismember(labelmatrix(cc), idx);
```


Vicente Morell vicente.morell@ua.es



P3

Imagen y Vídeo por Computador

. . .

Práctica 3: Segmentación y descriptores

Vicente Morell
vicente.morell@ua.es

Ingeniería Multimedia

Universitat d'Alacant
Universidad de Alicante

HUMAN ROBOTICS