

Práctica 4

Segmentación de Imagen (I)

Curso 2016 – 2017

El objetivo de esta práctica es comenzar a familiarizar al alumno con las herramientas básicas de segmentación de imagen en entorno MATLAB. Para ello se trabajará con la imagen en escala de grises ‘calculadora.tif’, que acompaña al material de esta práctica.

Utilice la ayuda de MATLAB (`help+` comando) para conocer el funcionamiento de los comandos utilizados en este guión. Tenga en cuenta que a lo largo de la práctica puede utilizar las instrucciones `clear all` y `close all` para evitar posibles interferencias con otras variables o ventanas.

I. Histograma y umbralización

Utilice la instrucción `imread` para leer en MATLAB la imagen ‘calculadora.tif’ proporcionada como material adjunto en la práctica. Examine visualmente la imagen y responda a las siguientes preguntas:

- ¿son todas las teclas del mismo tamaño?, ¿cuál es la de mayor tamaño?
- ¿tienen todas las teclas alguna letra/número en su interior?, ¿existe conectividad entre ellas?
- ¿aparecen letras/números en el exterior de las teclas?
- ¿qué letras/números presentan mayor intensidad luminosa, los del interior de las teclas o los del exterior?

Tras examinar la imagen, ¿qué propiedad cree que puede ser útil para segmentar las teclas de la calculadora?

Como sabe, el histograma de una imagen (función `imhist`) permite analizar la distribución de niveles de intensidad. Represente el histograma de la imagen ‘calculadora.tif’ e indique cuántas “crestas” se pueden identificar (tenga en cuenta que es posible que deba expandir el eje de ordenadas para no tener saturación en la representación del histograma). Identifique cada “cresta” del histograma con una etiqueta e indique qué zona de la imagen corresponde con cada etiqueta. Puede resultar útil hacer uso de la función `imtool` (en la esquina inferior izquierda de la representación proporciona el valor de intensidad del píxel sobre el que se sitúa el cursor del ratón).

Determine razonadamente un nivel umbral que permita separar adecuadamente el interior de las teclas del resto. Justifique el valor seleccionado y utilícelo para obtener una imagen binaria (en adelante variable `I_U`) con el comando `im2bw`, mostrando el resultado obtenido en una figura.

II. Segmentación y caracterización de regiones

MATLAB proporciona la función `bwlabel` para realizar la segmentación de una imagen binaria. Haga uso de la ayuda de esta función para indicar la vecindad considerada si se utiliza la siguiente instrucción

```
[Seg_I_U, Nobjetos] = bwlabel(I_U);
```

Utilice la instrucción anterior para segmentar la imagen `I_U`. Para visualizar la etiqueta de cada región haga uso de la función `imtool` como

```
imtool(Seg_I_U, [])
```

y pase el cursor del ratón sobre la región cuyo número de etiqueta desea conocer. El valor de la etiqueta aparece en la esquina inferior izquierda, justo tras las coordenadas del píxel sobre el que se sitúa el ratón.

La imagen en falso color que se genera a partir de la capa de segmentación (almacenada en la variable `Seg_I_U`) se puede visualizar con las instrucciones

```
RGB_Segment = label2rgb(Seg_I_U);  
figure, imshow(RGB_Segment)
```

Como ya se indicó en las sesiones teóricas, recuerde que MATLAB representa la etiqueta 0 con el color blanco. Teniendo en cuenta las instrucciones anteriores y sabiendo que cada color identifica un objeto diferente, indique cuántos objetos se obtienen como resultado de la segmentación. ¿Se obtiene un objeto por cada tecla? Como resultado de la segmentación, ¿se han extraído objetos en la parte externa de las teclas?

Es posible analizar las propiedades de las regiones segmentadas para eliminar aquellas que no interesan, por ejemplo las regiones externas a las teclas (de tamaño muy reducido). La función `regionprops` permite analizar determinadas propiedades de los objetos segmentados. Esta función toma como uno de los argumentos de entrada la capa de etiquetas obtenida de la segmentación (variable `Seg_I_U`), y como otro argumento la característica de los objetos/regiones que se desea conocer. Puesto que estamos interesados en el tamaño de los objetos consideraremos la propiedad `'Area'`, que proporciona el número de píxeles de un objeto.

Haga uso de la ayuda de MATLAB para conocer qué **otras** propiedades se pueden obtener de cada una de las regiones segmentadas.

Tras identificar la etiqueta de cada región, indique su tamaño (número de píxeles). Para obtener el tamaño del objeto con etiqueta *et* utilice las siguientes instrucciones, donde *et* hace referencia al número de etiqueta (valor entero no negativo).

```
Props = regionprops(Seg_I_U, 'Area')  
Props(et).Area
```

Puesto que el número de objetos es muy numeroso, puede resultar conveniente obtener el tamaño de cada región a través de la siguiente secuencia de instrucciones:

```
V_Area = [];  
  
for ind_obj=1:Nobjetos  
    V_Area = [V_Area Props(ind_obj).Area];  
end
```

Explique el objetivo del fragmento de código anterior y, tras ejecutarlo, utilice el comando `stem` para representar en una gráfica el tamaño de cada objeto (identificado a través de su número de etiqueta en el eje de abscisas). Analice la gráfica resultante y establezca un valor de tamaño umbral que permita eliminar las regiones externas a las teclas. Justifique la elección de este valor.

Construya de manera manual (no automática) un array uni-dimensional (vector) denominado `V_No_Interes`, cuyos elementos correspondan a los valores de las etiquetas que desea eliminar (filtrar).

Utilice la siguiente secuencia de comandos para filtrar, de la imagen binaria `I_U`, las regiones de no interés.

```
[n_filas, n_cols] = size(I_U);  
  
for ind_nfila=1:n_filas  
    for ind_ncol=1:n_cols  
        if I_U(ind_nfila,ind_ncol)  
            numero_et = Seg_I_U(ind_nfila,ind_ncol);  
            if sum(ismember(V_No_Interes,numero_et)) > 0  
                I_U(ind_nfila,ind_ncol) = 0;  
            end  
        end  
    end  
end
```

Explique el fragmento de código anterior (se recomienda realizar un diagrama de flujo). Visualice la imagen binaria resultante. En este punto debería tener al menos una región en el interior de cada tecla y no debería haber píxeles de primer plano en el exterior de las teclas. De no ser así, repita el proceso hasta conseguir el resultado deseado.

III. Procesado para identificación de la tecla 'Enter'

Puesto que en el interior de las teclas puede haber uno, dos o más caracteres, y la distancia entre ellos es relativamente pequeña (y siempre inferior a la distancia entre caracteres de otras teclas), se propone procesar la imagen binaria `I_U` (resultante de la etapa anterior) para agrupar en una misma región los caracteres asociados a la misma tecla, lo que permitiría caracterizar cada tecla con una región diferente.

Para conseguir el objetivo anterior se propone realizar un filtrado espacial de media (con una máscara de tamaño 5x5) sobre la imagen binaria `I_U`. Indique y justifique la secuencia de

instrucciones utilizadas para definir la máscara y realizar el filtrado. Tenga en cuenta el tipo de variable correspondiente a la imagen que se desea filtrar (probablemente ‘logical’) y de salida (de tipo ‘uint8’). Almacene el resultado en la variable `I_U_Fmedia`.

Aplique un umbral a la imagen `I_U_Fmedia`, de modo que sólo tengan valor nulo los píxeles “suficientemente alejados” de las letras/números superpuestos a las teclas. Justifique el umbral seleccionado y visualice la imagen binaria resultante (que denominaremos `I_U_Fmedia_Th`). Utilice esta imagen para determinar automáticamente el número de teclas (indique en la memoria la secuencia de instrucciones utilizada).

Haciendo uso de los comandos presentados y utilizados en esta práctica, proponga un procedimiento para identificar automáticamente la localización de la tecla ‘Enter’ a partir de `I_U_Fmedia_Th`.

Implemente el procedimiento y proporcione como resultado una imagen `I_Enter` en escala de grises y del mismo tamaño que la imagen original. La imagen `I_Enter` debe tener un nivel de intensidad nulo en todos los píxeles salvo en los correspondientes a la región donde se encuentra la tecla ‘Enter’, en cuyo caso debe mantenerse el nivel de intensidad de la imagen original. Anote la secuencia de instrucciones utilizada. La imagen resultante debe ser similar a la imagen de la Figura 1.

Justifique razonadamente qué habría sucedido si el tamaño de la máscara utilizada en el filtrado espacial hubiera sido mucho mayor, por ejemplo 35x35.



Figura 1. Imagen con la tecla ‘Enter’.