

Práctica 6

Morfología Binaria

Curso 2018 – 2019

El objetivo de esta práctica es familiarizar al alumno con las herramientas básicas de análisis de imagen, en concreto segmentación y morfología matemática para imágenes binarias.

Se trabajará con la imagen ‘Board_Recorte.tif’, imagen *true color* que acompaña al material de esta práctica. El objetivo es aplicar herramientas de segmentación (utilizadas en la práctica 4) y morfología matemática binaria para identificar de forma automática los chips del circuito impreso.

Para conocer el funcionamiento de los comandos utilizados en este guión, utilice la ayuda de MATLAB. 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. Transformación del espacio de representación

Utilice la instrucción `imread` para leer en MATLAB la imagen ‘Board_Recorte.tif’. Visualice la imagen en color e identifique manualmente los 7 chips de mayor tamaño. Para seleccionar manualmente los chips tenga en cuenta que todos deben compartir las mismas (o similares) características.

Visualice cada componente (R, G, B) e indique los comandos MATLAB utilizados. ¿Considera que alguna de las componentes podría ser más relevante para segmentar la imagen y obtener únicamente los 7 chips indicados anteriormente? Justifique su respuesta.

Considere la transformación a otro espacio de representación, en concreto el espacio HSI. Utilice para ello la función `rgb2hsi` proporcionada como material de la práctica. Utilice la ayuda de MATLAB para obtener las componentes en el espacio transformado. Visualice cada componente e indique qué rango de variación tiene cada una de las nuevas componentes. ¿Cómo ha determinado el rango dinámico?

A partir de la visualización de las componentes de la imagen en el espacio transformado, elija únicamente una componente (que denotaremos ‘Componente’) para continuar con el proceso de extracción de los 7 chips de mayor tamaño. Justifique razonadamente su elección.

II. Umbralización y filtrado

Observe el histograma de ‘Componente’. Puesto que no parece muy clara la elección de un valor umbral a partir del histograma, utilizaremos la instrucción `graythresh` para obtener automáticamente un valor umbral utilizando el método de Otsu. Haga uso de la ayuda de Matlab para explicar qué criterio sigue el método de Otsu para determinar el umbral. ¿Qué valor umbral se obtiene? Utilice ese valor umbral para umbralizar la imagen ‘Componente’. Escale la imagen

resultante para que la escala de variación sea $[0,255]$ y convierta la variable resultante a una de tipo `uint8`. Anote la secuencia de comandos utilizada y visualice la imagen resultante.

Analice con detalle el resultado y justifique qué operador, relacionado con filtrado espacial y utilizado en prácticas anteriores, se puede aplicar para homogeneizar la región interna de los chips de modo que la imagen resultante siga siendo una imagen binaria tras la aplicación del filtro. Aplique el operador considerando una máscara cuadrada de tamaño 5×5 . Indique la instrucción utilizada para realizar el filtrado y visualice la imagen filtrada, comentando las principales diferencias (relacionadas con las regiones de los objetos de interés) con la imagen sin filtrar.

III. Aplicación de operadores morfológicos

Como sabe, la aplicación de operadores morfológicos requiere la definición de un elemento estructurante (EE), que en MATLAB se genera mediante la instrucción `strel`. Utilice la ayuda de MATLAB para generar un EE cuadrado de lado 35 píxeles que almacenaremos en la variable 'EE_cuadrado'.

Utilice 'EE_cuadrado' para aplicar sobre el resultado final de la etapa II los siguientes operadores morfológicos:

- Erosión (instrucción `imerode`)
- Dilatación (instrucción `imdilate`)
- Apertura (instrucción `imopen`)
- Cierre (instrucción `imclose`)

Visualice la imagen obtenida tras aplicar cada uno de los operadores anteriores y justifique razonadamente el resultado teniendo en cuenta los contenidos presentados en las sesiones teóricas. Justifique qué imagen (de las cuatro anteriores) se debe considerar para delimitar lo mejor posible cada uno de los 7 chips seleccionados manualmente al inicio de la práctica. Denotaremos a la variable que almacena esta imagen como 'Im_Res_Morf'.

IV. Segmentación y caracterización de objetos

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

```
IM_Seg = bwlabel(IM_Res_Morf);
```

Utilice la instrucción anterior para segmentar la imagen 'Im_Res_Morf'. La capa de segmentación (almacenada en la variable `IM_Seg`) se puede visualizar con las instrucciones

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

¿Cuántos objetos se obtienen como resultado de la segmentación? En este caso el número de objetos es muy reducido y se puede contabilizar teniendo en cuenta el número de colores (recuerde que cada color identifica un objeto diferente). Para obtener el número de objetos mediante una única instrucción se puede utilizar la siguiente expresión en línea de comandos:

```
Num_objetos = max(IM_Seg(:))
```

¿Qué es lo que sucede? ¿Coincide el resultado con el número de chips obtenidos manualmente? Justifique el resultado.

Realice alguna modificación sobre la imagen binaria a segmentar para que, cuando se realice la segmentación binaria, el número de objetos segmentados sea 7. Incluya en la memoria la secuencia de instrucciones utilizada, junto con la capa de segmentación resultante.

Para determinar el número de chips con forma cuadrada y forma rectangular se puede recurrir a examinar las propiedades de los objetos segmentados mediante la función `regionprops`. La función `regionprops` toma como uno de los argumentos de entrada la capa de etiquetas obtenida de la segmentación (variable `IM_Seg`), y como otro argumento la característica de los objetos que se desea conocer. En esta práctica consideraremos la propiedad de excentricidad (`'Eccentricity'`), relacionada con la forma de las regiones.

Haga uso de la ayuda de MATLAB para explicar qué representa esta característica (`'Eccentricity'`) desde el punto de vista geométrico, y qué valores son esperables (altos o bajos) para cada una de las regiones segmentadas, teniendo en cuenta la etiqueta de cada región. Para obtener visualmente la etiqueta de cada región haga uso de la función `imview` como

```
imview(IM_Seg, [])
```

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.

Tras identificar la etiqueta de cada región indique, para cada etiqueta, la excentricidad asociada con el objeto que ésta representa. Para obtener la excentricidad del objeto con etiqueta *et* (*et* es un valor entero positivo) utilice la instrucción

```
Props = regionprops(IM_Seg, 'Eccentricity')  
Props(et).Eccentricity
```

En base a esta información, **indique qué procedimiento seguiría** para separar los chips rectangulares de los cuadrados. Justifique su respuesta.

Las técnicas de morfología matemática también se pueden utilizar para delimitar las fronteras de los objetos. Explique razonadamente cómo obtener los contornos (finos) de los chips utilizando únicamente un elemento estructurante, dos operadores morfológicos y un operador aritmético punto a punto. Indique también la secuencia de instrucciones y la imagen resultante del proceso. Tenga en cuenta que la imagen de partida es la imagen binaria cuya segmentación ha proporcionado 7 objetos.

Explique la influencia del tamaño y forma del EE.