

Proyecto de reconocimiento de piezas 2D

Rafael Carbonell Lázaro

racarla1@inf.upv.es

Resumen

Este proyecto trabajo de visión en 2D consiste en la identificación de fichas de domino.



Índice

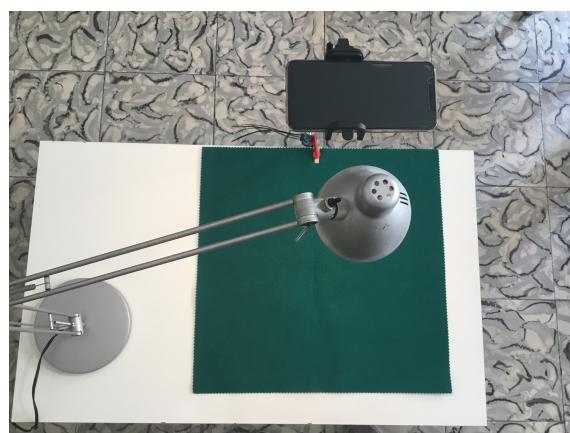
1. Adquisición y digitalización	3
2. Preprocesado y binarización	5
3. Extracción de características y reconocimiento	7

1. Adquisición y digitalización

Para el sistema de digitalización hemos empleado:

- Tapete de fieltro verde de 50 x 50 cm
- Mordaza + Bridas + Palo selfie (a modo de trípode)
- Móvil Xiaomi Mi 8 con una cámara de 12 MP f/1.8 (Sony IMX 363) con una resolución de 4032 x 3024 px
- Flexo de estudio
- Juego de domino de 28 piezas

Se adjunta una serie de imágenes.



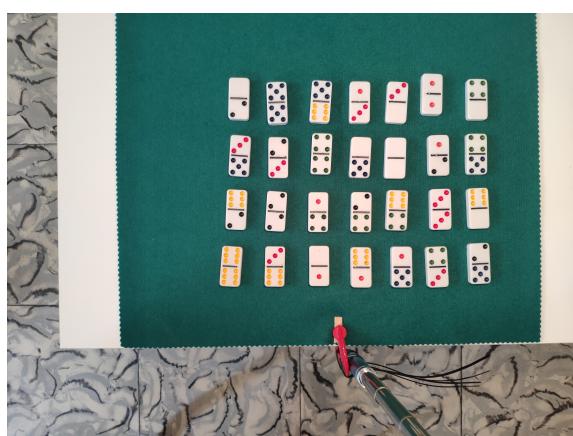
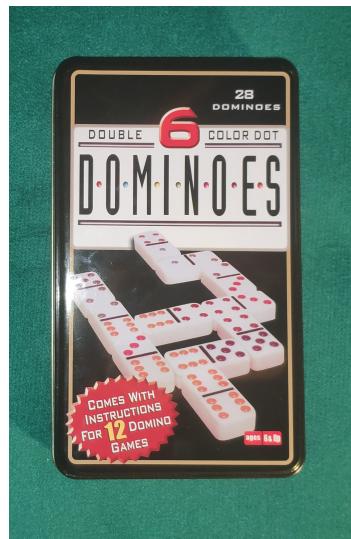
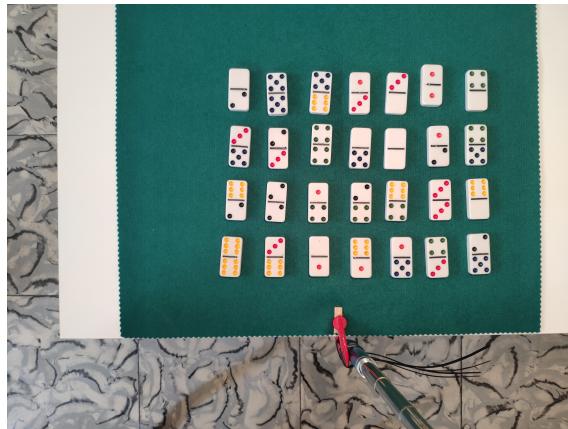


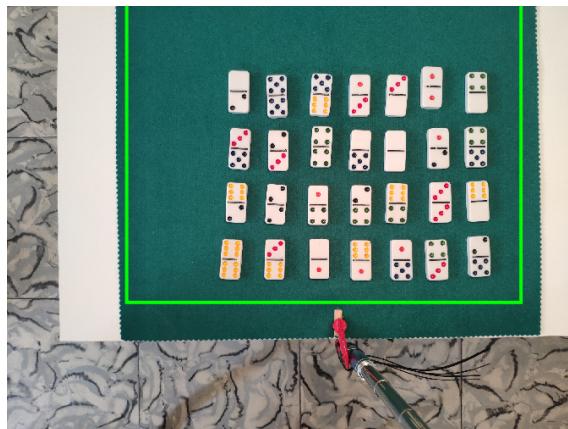
Figura 1: Imagen tomada desde la cámara del móvil.

2. Preprocesado y binarización

En esta etapa cubriremos desde la adquisición de la imagen hasta la binarización.



En primer lugar, partiendo como ejemplo la figura 1, debemos seleccionar un área de interés a analizar en todas la imágenes. La área interés esta delimitada en la imagen por un rectángulo verde. La imagen pasa de 3024x4032 a 2100x2800 con un offset en el eje X de 850.

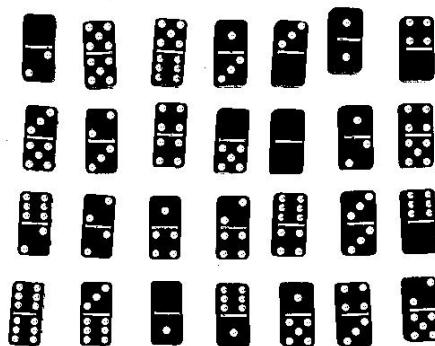


En segundo lugar, convertimos esta área de imágenes la convertimos a RGB a binaria. Para ello generamos un histograma para cada canal de la imagen y realizamos una umbralización global por cada canal, y con los valores de los canales resultantes hacemos una .^and"para cada una de las posiciones de la imagen, como podemos observar en el siguiente pedazo de código.

```
1 % Convert RGB image to chosen color space
2 I = RGB;
3
4 % Define thresholds for channel 1 based on histogram settings
5 channel1Min = 101.000;
6 channel1Max = 255.000;
7
8 % Define thresholds for channel 2 based on histogram settings
9 channel2Min = 149.000;
10 channel2Max = 255.000;
11
12 % Define thresholds for channel 3 based on histogram settings
13 channel3Min = 81.000;
14 channel3Max = 255.000;
15
16 % Create mask based on chosen histogram thresholds
17 sliderBW = (I(:,:,1) >= channel1Min) & (I(:,:,1) <= channel1Max) & ...
```

```
18      (I(:,:,2) >= channel2Min ) & (I(:,:,2) <= channel2Max) & ...
19      (I(:,:,3) >= channel3Min ) & (I(:,:,3) <= channel3Max);
20 BW = sliderBW;
```

Como resultado obtenemos la siguiente imagen:



3. Extracción de características y reconocimiento

Para la extracción de las características, empleamos la función 'regionprops' que sirve para medir las propiedades de las regiones de imagen. A esta función le puedes indicar que propiedades buscar y extraer de una imagen y tiene un conjunto muy extenso de características a extraer.

En este caso, en un principio, extraímos todas las características y tras un análisis de las características extraídas llegamos a la conclusión de que las propiedades que nos ayudarían a discriminar y determinar los elementos de la imagen son: 'Area', 'BoundingBox', 'Centroid', 'Eccentricity', 'Orientation'.

Estas características están asociadas a:

- Area - Número real de píxeles en la región, devueltos como escalar.
- BoundingBox - Rectángulo más pequeño que contiene la región, devuelto como un vector 1 por Q^*2 , donde Q es el número de dimensiones de la imagen.
- Centroid - Centro de la masa de la región, regresó como un 1-por-Q Vector.
- Eccentricity - Excentricidad de la elipse que tiene los mismos segundos momentos que la región, regresó como un escalar. La excentricidad es la relación entre la distancia entre los focos de la elipse y su longitud de eje principal.
- Orientation - Ángulo entre el eje xy el eje principal de la elipse que tiene los mismos segundos momentos que la región, devuelto como escalar.

Una vez hemos extraído las características de una imagen, ocurren varios problemas. El primero de ellos es que la imagen contiene ruido con regiones muy grandes o muy pequeñas, ya que la umbralización no es perfecta, entonces para solventar este problema creamos un primer filtro que consiste en delimitar la área máxima y mínima que nos interesan.

Entonces nuestro objetivo sigue siendo detectar las fichas de domino y que valores tienen en cada cara. Por lo tanto, para identificar si es un ficha y el numero de fichas la estrategia que seguiremos es detectar la barra intermedia de cada ficha. Junto con diferenciarlas de los puntos.

Retomando la idea de filtrar las áreas, analizamos que las barras y los puntos están comprendidos entre áreas de 500 y 1500 px de área. Como consecuencia tenemos una primera clasificación, a raíz de esto surge un segundo problema a solventar que consiste en discriminar entre puntos y barras.

Para discriminar que característica es una barra o un punto emplearemos la propiedad de 'Eccentricity', ya que cuando la característica es un círculo el valor se aproxima a 0, mientras que si es una barra esta se aproxima a 1.

Llegados a este punto, ya tenemos resuelto los siguientes problemas resueltos, cuantas fichas hay porque sabemos el numero de barras y cuantos puntos hay en total. Aunque queda otro aspecto a determinar como que valor tiene cada ficha en cada mitad de la pieza.

Dicho lo anterior, para la clasificación de los puntos con sus respectivas mitad incluso a que pieza van asociado aplicamos la siguiente estrategia que consiste en aproximar el tamaño en píxeles de una pieza con un alto 286 y un ancho de 143 píxeles. Con esta información, tenemos una ficha base rectangular la cual cogemos 6 puntos, como son las esquinas del rectángulo y dos puntos laterales a la altura de la mitad de la pieza, todos ellos se calculan respecto al centroide de cada barra obtenida en la extracción de características, de hecho se calcula y guarda en una estructura a la vez que se procesan las características.

Con cada uno de estos puntos aplicamos la operación de rotación también con respecto al centroide de esta forma, si ploteamos los puntos, de forma aproximada queda la pieza marcada con puntos en la imagen. Con estos seis puntos podemos formar dos areas respectivamente a cada mitad. En una segunda iteración sobre el numero de fichas obtenidas con una función 'inpolygon'

que determina si un punto esta o no dentro de la área tenemos el numero de puntos obtenidos en cada mitad y por tanto la pieza identificada. Como podemos apreciar en la siguiente imagen.

