

## PRÁCTICA 5

### “Descripción matemática de objetos: ejemplo en reconocimiento de formas geométricas”

#### Objetivo:

El objetivo de esta práctica es presentar una metodología básica de selección de características y aplicarla en un problema de reconocimiento de formas, determinando las características más significativas para dicho reconocimiento.

#### Descriptores o características a analizar:

- **Compacticidad, Excentricidad, Solidez (bounding box), Solidez (convex hull), Número de Euler.** Estos descriptores se calcularán utilizando la función de matlab `regionprops` (ver definiciones en ayuda matlab).

Ejemplo de cálculo de área y centroide mediante el uso de `regionprops`:

```
[Ietiq N]=bwlabel(Ib); %Ib = matriz binaria ;Ietiq = matriz
etiquetada

stats=regionprops(Ietiq,'Area','Centroid');
% stats.Area  contiene información de las áreas de cada objeto

% stats.Centroid - contiene información de la coordenada x
(columna) y la coordenada y (fila) del centroide de cada objeto

areas=cat(1,stats.Area); % vector columna con las áreas de cada
objeto
centroides=cat(1,stats.Centroid); % matriz de dos columnas
(x,y) y tantas filas como objetos etiquetados distintos de cero
haya
```

- **Momentos invariantes de Hu:** para su cálculo se debe implementar la función `Funcion_Calcula_Hu` (ver Documentación Anexo I ; además se facilita en el material de la práctica, una variable matlab que contiene un objeto, dado a través de una imagen binaria, y los resultados de sus Momentos de Hu):

```
Hu = Funcion_Calcula_Hu (Matriz_Binaria)
```

donde:

- `Matriz_Binaria`: matriz binaria con dos posibles valores, *0*'s, píxeles que no son del objeto que se quiere caracterizar y *1*'s, píxeles del objeto a caracterizar.

- $H_u$ : Vector con los valores de los 7 momento invariantes de  $H_u$  del objeto.

**Imágenes a utilizar (disponibles en el material facilitado de la práctica):**

- **Imágenes de entrenamiento:** para obtener características de muestras cuya clase es conocida, seleccionar aquellas que sean las más adecuadas y diseñar, a partir de ellas, el clasificador (Circ\_ent.jpg, Cuad\_ent.jpg y Trian\_ent.jpg).
- **Imágenes de test:** para comprobar en primera instancia el funcionamiento del clasificador diseñado y ajustar, si fuese necesario, los parámetros que utiliza (Test1.jpg, Test2.jpg y Test3.jpg).
- **Imágenes de funcionamiento:** para comprobar el funcionamiento del clasificador diseñado (Func1.jpg y Func2.jpg).

**PRIMERA PARTE: Obtención de patrones de entrenamiento y selección de características adecuadas (fase de entrenamiento)**

**1.- Obtención de patrones de entrenamiento:**

- En esta fase se obtendrá una representación numérica de todos los objetos de las imágenes de entrenamiento disponibles (distintas imágenes con objetos cuya forma geométrica es conocida). Esta representación consistirá en el cálculo de todas las características que serán objeto de análisis.
- La información se guardará en dos matrices, `inputs` y `outputs`. `Inputs` contendrá tantas filas como características se estén calculando y tantas columnas como muestras de entrenamiento haya. Por otra parte, `outputs` será una matriz que asignará a cada columna la codificación utilizada para la clase del objeto caracterizado por la descripción matemática dada en la columna correspondiente de `inputs`.

- Normalización de datos: una vez obtenidos todos los patrones de entrenamiento, se normalizan para que cada uno de los descriptores presente media cero y desviación típica 1.

## **2.- Selección de características adecuadas:**

- **Visualización de datos: representación de los mismos en espacios de características bidimensionales:**

Para el conjunto de características seleccionadas (compacticidad, excentricidad, solidez-bounding box, solidez-convex hull, número de Euler, 7 momentos invariantes de Hu), representar en espacios de características bidimensionales los patrones de entrenamiento de cada clase (el conjunto de representaciones deben involucrar a todas las características). Decidir de forma cualitativa las características que son más adecuadas en este problema de reconocimiento.

- **Cuantificación individual de características y primera selección de las mismas:**

Para cada característica, determina el grado de separabilidad que proporciona de forma individual mediante CSM - ("Class Scatter Matrix"), esto es, mediante la estimación de las matrices de dispersión entre y dentro de las clases (ver metodología de cálculo Documentación Anexo II). Para ello, utilizar la función facilitada `indiceJ`:

$$J = \text{indiceJ}(\text{inputs}, \text{outputs})$$

donde:

- `inputs` y `outputs`: datos entrada-salida generados de acuerdo al punto anterior.
- `J`: grado de separabilidad.

Utiliza los valores de `J` para ordenar el conjunto de características de mayor a menor importancia de acuerdo a este criterio de evaluación individual de características. De las 12 características, selecciona las 6 más relevantes.

- **Selección final de características:**

Encontrar las 3 características que proporcionan de forma conjunta, de acuerdo al criterio CSM, la mayor separabilidad en los datos y que formarán el vector de características que se utilizará en la etapa de clasificación.

Observación: la función de Matlab `combnk(1:N, k)` devuelve todas las combinaciones de  $N$  elementos tomados en grupos de  $k$ .

## **SEGUNDA PARTE: Diseño del Clasificador (fase de test)**

- A partir de los patrones de entrenamiento de las características elegidas, obtenidos para todo el conjunto de muestras disponibles de cada clase (distintos tipos de formas geométricas), diseñar y ajustar, si fuese necesario, un clasificador utilizando el conjunto de imágenes de test. Se deberán utilizar los siguientes clasificadores:
  - Clasificador Mínima Distancia Euclídea.
  - kNN “k vecinos más próximos” (knn, k-nearest-neighbours): dado un conjunto de objetos prototipo de los que ya se conoce su clase (es decir, dado un conjunto de formas geométricas de muestra) y dado un nuevo objeto cuya clase no conocemos (forma a reconocer) se busca entre el conjunto de prototipos los “k” más parecidos al nuevo objeto. A este se le asigna la clase más numerosa entre los “k” objetos prototipo seleccionados.

Programación del Clasificador KNN:

- ✓ Instrucciones Matlab de utilidad: sort, tabúlate

- ✓ Función knn:

**function salida = knn (inputs, outputs, vector, k)**

- ✓ Ejemplo: suponiendo un problema de reconocimiento en el que hay 4 clases y se disponen de 3 objetos para cada una de ellas:

inputs:

0.79	0.78	0.77	0.46	0.45	0.45	0.52	0.51	0.50	0.65	0.63	0.62
3.22	3.31	3.38	1.04	1.03	1.04	1.73	1.74	1.74	2.10	2.12	2.12
12.63	12.88	12.39	3.64	3.61	3.61	4.88	4.79	4.70	4.54	4.45	4.38
13.15	12.91	12.92	3.85	3.83	3.84	5.77	5.62	5.49	7.28	7.21	7.12
27.83	27.19	25.45	7.94	7.92	7.94	11.94	11.76	11.48	11.37	11.13	10.93
16.01	14.74	15.30	4.61	4.60	4.62	8.73	8.49	8.23	7.05	6.93	6.83
27.26	26.25	25.54	7.98	7.96	7.99	12.99	12.47	12.10	12.90	12.45	12.11

outputs:

0	0	0	1	1	1	2	2	2	3	3	3
---	---	---	---	---	---	---	---	---	---	---	---

Vector:

Salida:

K:

0.7719  
3.3892  
13.3058  
13.5537  
27.2191  
17.3492  
27.8899

0

1-3

### **TERCERA PARTE: Fase de funcionamiento:**

- Se utilizará el clasificador diseñado en la etapa anterior para implementar un sistema que reconozca los distintos objetos (formas geométricas) extraídos de una imagen (imágenes de funcionamiento). Estas imágenes se utilizarán únicamente cuando esté finalizado el diseño del sistema de reconocimiento para comprobar su correcto funcionamiento. Las imágenes de funcionamiento han sido obtenidas de la misma forma que el conjunto de imágenes de test y, por tanto, reúnen sus mismas características. De ellas se conoce el número de objetos presentes.

## ANEXO I - MOMENTOS INVARIANTES DE HU

Sea  $f$  una imagen binaria con dos posibles valores,  $0$ 's, píxeles que no son del objeto que se quiere caracterizar y  $1$ 's, píxeles del objeto a caracterizar:

- ❖ Momentos ordinarios de orden  $(p+q)$ : 
$$m_{pq} = \sum_x \sum_y x^p y^q f(x, y) \quad p, q = 0, 1, 2, \dots$$
- ❖ Momentos centrales de orden  $(p+q)$ : 
$$\omega_{pq} = \sum_x \sum_y (x - \bar{x})^p (y - \bar{y})^q f(x, y) \quad , \quad \bar{x} = \frac{m_{10}}{m_{00}} \quad ; \quad \bar{y} = \frac{m_{01}}{m_{00}}$$
- ❖ Momentos centrales normalizados: 
$$\eta_{pq} = \frac{\omega_{pq}}{(\omega_{00})^\gamma} \quad ; \quad \gamma = \frac{p+q}{2} + 1 \quad ; \quad (p+q) = 2, 3, \dots$$
- ❖ Momentos invariantes de Hu:

$$\phi_1 = \eta_{20} + \eta_{02}$$

$$\phi_2 = (\eta_{20} - \eta_{02})^2 + 4\eta_{11}^2$$

$$\phi_3 = (\eta_{30} - 3\eta_{12})^2 + (3\eta_{21} - \eta_{03})^2$$

$$\phi_4 = (\eta_{30} - \eta_{12})^2 + (\eta_{21} - \eta_{03})^2$$

$$\phi_5 = (\eta_{30} - 3\eta_{12})(\eta_{30} + \eta_{12}) \left[ (\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \right] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \left[ 3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right]$$

$$\phi_6 = (\eta_{20} - \eta_{02}) \left[ (\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right] + 4\eta_{11}(\eta_{30} + \eta_{12})(\eta_{21} + \eta_{03})$$

$$\phi_7 = (3\eta_{21} - \eta_{03})(\eta_{30} + \eta_{12}) \left[ (\eta_{30} + \eta_{12})^2 - 3(\eta_{21} + \eta_{03})^2 \right] + (3\eta_{21} - \eta_{03})(\eta_{21} + \eta_{03}) \left[ 3(\eta_{30} + \eta_{12})^2 - (\eta_{21} + \eta_{03})^2 \right]$$

- Para que todos los momentos se encuentren dentro del mismo orden de magnitud, determinar los momentos normalizados:

$\phi_i^* = \text{abs} \left\{ \ln \left[ \text{abs}(\phi_i) \right] \right\}$ . Si algún momento resulta ser 0, antes de calcular el momento normalizado reemplazar su valor por  $1 * \exp(-100)$ .

## ANEXO II

### MEDIDA DEL GRADO DE SEPARABILIDAD ENTRE CLASES A PARTIR DE LA ESTIMACIÓN DE MATRICES DE DISPERSIÓN ENTRE Y DENTRO DE LAS CLASES

The within-class scatter matrix ( $S_w$ ) indicates the distribution of sample points around their respective mean vectors and is defined as:

$$S_w = \sum_{i=1}^C S_i \quad (4.2)$$

$$S_i = \sum_{n \in C_i} (X_n - M_i)(X_n - M_i)^T \quad \text{and} \quad M_i = \frac{1}{N_i} \sum_{n \in C_i} X_n \quad (4.3)$$

where  $C$  is the number of classes,  $N_i$  represents the number of examples in class  $C_i$ ,  $X_n$  refers to the sample  $n$ , and  $M_i$  is the mean of class  $C_i$ . The between-class scatter matrix ( $S_b$ ) represents the scatter of samples around the mean vector of the class mixture and is defined as:

$$S_b = \sum_{i=1}^C N_i (M_i - M)(M_i - M)^T \quad (4.4)$$

$$M = \frac{1}{N} \sum_{n=1}^N X_n = \frac{1}{N} \sum_{i=1}^C N_i M_i \quad (4.5)$$

while  $N = \sum_i N_i$  shows the total number of sample points in the dataset. After within-class and between-class matrices are measured the following metric  $J$  can be obtained:

$$J = \text{trace} \left( \frac{S_b}{S_w} \right) \quad (4.6)$$