# COMPUTER VISION

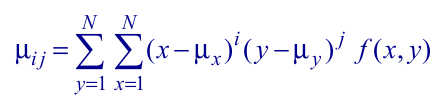
**EXERCISE 5a: REGION DESCRIPTORS**

Concepts: Hu moments

1. **Central moments:** Implement a function that computes the central moments (until order 3) of a grayscale image *I*. The function prototype must be as follows: *Note: it really helps if you implement another function to compute the non-central, or raw moments, and you use it to retrieve the central ones. If not, you can use the expression below.*

[mu00,mu10,mu01,mu11,mu20,mu02,mu21,mu12,mu30,mu03]=momentos\_centrales(I)

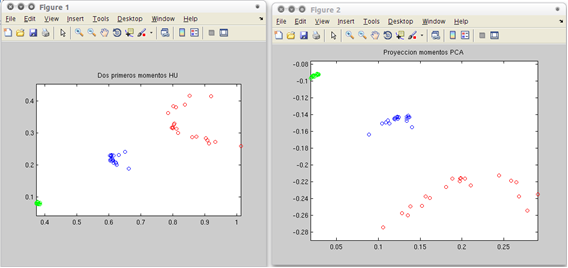
Central moments expression:



To test your code, if you run the function with the *‘botella\_A\_1.bmp’* as argument, you have to obtain the following results:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **mu00** | 1148052 | **mu10** | 0 | **mu01** | 0 |
| **mu11** | -4.3331e+06 | **mu20** | -2.6373e+06 | **mu02** | -7.1258e+06 |
| **mu21** | 1.0521e+07 | **mu12** | 1.9549e+07 | **mu30** | 5.1157e+06 |
| **mu03** | 3.4196e+07 |  |  |  |  |

1. **Hu moments:** For the grayscale images attached, corresponding to 3 different types of bottles, do the following: *Note: use only the first 15 images of each type.*
   1. Compute the Hu moments employing the *“momentos\_Hu”* function included below. This functions relies on the *“momentos\_centrales”* function implemented in the previous point.
   2. Graphically represent the values of the two first Hu moments. You should employ a different color for each bottle type.

****

1. **Centroid:** Compute the centroid (center of mass) of the first two Hu moments of each bottle type and display them in the previous figure. Employ a different mark to distinguish them from the other points.

1. **Euclidean classifier:** Implement a Matlab script that:
   1. Asks for the name of an image through the keyboard and read it.
   2. Computes the two first Hu moments of that image. This will be the descriptors vector.
   3. Compares such a vector with the center of mass of each bottle type retrieved in the previous point. *Note: to compare two vectors employ the Euclidean distance.*
   4. Shows in the screen the type of the bottle in the image.

**Function “momentos\_Hu”**

|  |
| --- |
| function HM=momentos\_Hu(I)  % Calcula los momentos de Hu invariantes de una imagen (I) en niveles de gris  % Si se desea obtener la descripción de momentos de un único objeto de la  % imagen, el resto hay que ponerlos a cero.  %  % Entrada: Imagen I en niveles de gris  % Salida: Vector HM (7x1) de momentos de Hu (invariantes)  %  % Fecha: 2009-2012 Javier Gonzalez    I=double(I)/255;    % Momentos centrales  [mu00,mu10,mu01,mu11,mu20,mu02,mu21,mu12,mu30,mu03] = momentos\_centrales(I);    %Momentos normalizados  u002 = mu00\*mu00;  u0025 = mu00^2.5;  %u0015 = mu00^1.5  n02 = mu02/u002;  n20 = mu20/u002;  n11 = mu11/u002;  n12 = mu12/u0025;  n21 = mu21/u0025;  n03 = mu03/u0025;  n30 = mu30/u0025;    %Momentos invariantes de Hu  f1 = n20+n02;  f2 = (n20-n02)^2 + 4\*n11^2;  f3 = (n30-3\*n12)^2+(3\*n21-n03)^2;  f4 = (n30+n12)^2+(n21+n03)^2;  f5 = (n30-3\*n12)\*(n30+n12)\*((n30+n12)^2 - 3\*(n21+n03)^2) + (3\*n21-n03)\*(n21+n03)\*(3\*(n30+n12)^2 - (n21+n03)^2);  f6 = (n20-n02)\*((n30+n12)^2 - (n21+n03)^2) + 4\*n11\*(n30+n12)\*(n21+n03);  f7 = (3\*n21-n03)\*(n30+n12)\*((n30+n12)^2 - 3\*(n21+n03)^2) - (n30-3\*n12)\*(n21+n03)\*(3\*(n30+n12)^2 - (n21+n03)^2);  HM = [f1 f2 f3 f4 f5 f6 f7];  return; |

**OPTIONAL! EXERCISE 5b: Centroid and principal directions**

Concepts: Centroid and principal direction. Eigenvalue and eigenvector of a dispersion matrix.

Write a function that computes and draws the centroid and principal direction of the segmented region, given by a binary image (input argument to the function) where the background has a value of 0 and the segmented region a value of 1. *Note: for the segmentation step you can use the region growing algorithm.*

**Matlab functions to use:**

|  |  |
| --- | --- |
| **[V,D] = eig(A)** | Computes the eigenvalues and eigenvectors of the matrix A. |
| **line ([x0,x1],[y0,y1], 'LineWidth',1, 'Color',[1 0 0])** | (Optative, this can be done with plot and a pair of points) Draws a line from (xo,y0) to (x1,y1) with the specified parameters. |

