



**UNIVERSIDAD AUTÓNOMA  
DE AGUASCALIENTES**

**Centro de Ciencias Básicas**

**Departamento: Ciencias de la Computación**

**Área Académica: Inteligencia Artificial**

**Graficación**

**Ingeniería en Computación Inteligente**

**8° “A”**

**“Tarea 2:**

**Transformaciones en objetos binarios”**

**Alumnos:**

Cesar Eduardo Elías del Hoyo

José Luis Sandoval Pérez

Diego Emanuel Saucedo Ortega

Carlos Daniel Torres Macías

**Maestro:** Hermilo Sánchez Cruz

Aguascalientes, Ags. Marzo, 24 de 2025

## Selección de imágenes y conteo de 1-píxeles

Para el conteo de 1-píxeles, lo haremos a través del uso de histogramas en GIMP, filtrando el histograma a través de la selección de píxeles por valor e indicando que busque los píxeles con un rango de valor 255 a 255, correspondiente al blanco. De esta forma indica cuantos píxeles hay en cada imagen y cuantos de ellos son 1-píxeles.

Nombre	Conteo de píxeles totales	Conteo de 1- píxeles
butterfly-2.gif	459000	94506
cattle-3.gif	613530	153599
confusions.gif	159506	110018
deer-4.gif	970224	158591
dog-7.gif	214608	45838
elephant-4.gif	979038	215379
guitar-3.gif	281502	46641
lizzard-19.gif	416412	82676
pocket-11.gif	194376	86371
shapedata.gif	202924	123134

## Determinación de factor alpha para normalizar cantidad de 1-píxeles

Ahora, determinaremos el factor alpha para que cada imagen tenga aproximadamente la misma cantidad de 1-píxeles, primeramente, estableceremos el promedio de 1-píxeles de las diez imágenes, con el fin de que todas busquen tener esa misma cantidad de píxeles blancos

$$P = \frac{94506 + 153599 + 110018 + 158591 + 45838 + 215379 + 46641 + 82676 + 86371 + 123134}{10} \\ = 111675.3$$

Redondeamos el promedio a 111,675, por lo que usaremos este valor como referencia para escalar cada imagen. Con esto calculamos el valor de  $\alpha$  para cada imagen. Este se calcula con la siguiente formula

$$\alpha = \sqrt{\frac{Ndeseado}{Nactual}}$$

Calculamos el factor para cada imagen

Imagen	Factor escalar de $\alpha$
butterfly-2.gif	1.08704838
cattle-3.gif	0.85267665
confusions.gif	1.0075038
deer-4.gif	0.83914941
dog-7.gif	1.56086641

elephant-4.gif	0.7200736
guitar-3.gif	1.54737166
lizzard-19.gif	1.16222131
pocket-11.gif	1.13708934
shapedata.gif	0.95233461

Escalamos la imagen en GIMP, en la sección de Imagen -> Escalar imagen y multiplicamos los valores de Anchura y Altura por el factor escalar de la respectiva imagen. Cada imagen reescalada estará en el ZIP, con el mismo nombre, pero con el prefijo "escl" y cambiando el formato de la imagen a PNG, ya que no tiene movimiento, ni diversas capas para que sea necesario conservarla en GIF. Tenemos la nueva tabla de conteo de 1-píxeles.

Nombre escalado	Nuevo conteo 1-píxeles
esclbutterfly-2.png	111628
esc1cattle-3.png	111612
esc1confusions.png	111048
esc1deer-4.png	111780
esc1dog-7.png	111636
esc1elephant-4.png	111846
esc1guitar-3.png	111447
esc1lizzard-19.png	111548
esc1pocket-11.png	111623
esc1shapedata.png	111073

Calcular el invariante de escala siguiente, antes y después del escalamiento

Este proceso se hará con la formula siguiente:

$$\eta_{pq} = \frac{\mu_{pq}}{\mu_{00}^{\frac{p+q}{2}+1}}$$

con  $p, q = 0, 1$  y  $2$ .

Imagen	mu_10 (Antes)	mu_01 (Antes)	mu_11 (Antes)	mu_20 (Antes)	mu_02 (Antes)	mu_10 (Despues)	mu_01 (Despues)	mu_11 (Despues)	mu_20 (Despues)	mu_02 (Despues)
butterfly-2.png	290.406821	324.981102	0.004063	0.003814	0.005239	315.726449	353.392312	0.004067	0.003817	0.005245
cattle-3.png	429.861288	372.09851	0.004041	0.005207	0.004214	366.560415	317.065602	0.004041	0.005211	0.004214
confusing.png	235.81314	171.411635	0.001429	0.002607	0.001407	237.304284	172.460754	0.001433	0.002615	0.001411
deer-4.png	481.51134	505.315693	0.00607	0.006751	0.006646	404.031893	424.258096	0.006068	0.006745	0.006648
dog-7.png	218.337558	210.375998	0.004217	0.005642	0.004546	341.144075	328.834596	0.004226	0.005651	0.004558
elephant-4.png	435.332855	462.655199	0.003707	0.0037	0.004172	313.567164	333.3223	0.003705	0.003699	0.004171
guitar-3.png	192.054351	388.905684	0.007387	0.003931	0.015197	297.126697	601.148142	0.007391	0.003936	0.015193
lizzard-19.png	268.257548	371.802155	0.004317	0.003718	0.008344	311.754357	431.89158	0.004319	0.003721	0.008345
pocket-11.png	183.227495	321.110801	0.002664	0.001915	0.005423	208.448026	365.041999	0.002666	0.001918	0.005423
shapedata.png	268.932707	192.090844	0.001637	0.003075	0.001591	256.114672	182.095577	0.001638	0.003091	0.001587

## Generación de gráficos de celdas binarias

Para visualizar la distribución espacial de los 1-píxeles en cada objeto binario, se generaron gráficos en los que cada celda cuadrangular representa un píxel de la imagen binarizada. La finalidad de esta representación es resaltar la estructura y ocupación de las regiones activas en la imagen procesada.

Cada gráfico muestra la imagen binaria con una cuadrícula superpuesta, permitiendo una mejor interpretación de la segmentación de los píxeles. Además, se incluye información sobre el número total de celdas, la cantidad de píxeles activos y el porcentaje de ocupación dentro de la imagen.

Este proceso se hizo con el programa que hicimos en Python llamado `Obtencion_de_graficos.py` y almacena los resultados en la carpeta `mapas` que está dentro de la carpeta de trabajo.

## Detección y Visualización de Contornos Binarios con Vecindad-8

Para analizar la estructura de los objetos binarios en las imágenes, se extrajeron sus contornos utilizando la técnica de **vecindad-8**. Esto permite una mejor representación de los bordes al considerar todos los píxeles adyacentes a un punto dado.

### Procedimiento:

1. Se cargaron las imágenes en escala de grises y se binarizaron para asegurar que solo contengan valores de **0 (negro)** y **255 (blanco)**.
2. Se aplicó el método `cv2.findContours()` de OpenCV con los parámetros:
  - a. **cv2.RETR\_EXTERNAL**: para extraer únicamente los contornos externos de los objetos.
  - b. **cv2.CHAIN\_APPROX\_SIMPLE**: para optimizar la representación del contorno eliminando puntos redundantes.
3. Los contornos detectados se dibujaron sobre una imagen en blanco para facilitar su visualización.

4. Finalmente, se generaron gráficos computacionales de las celdas correspondientes al contorno de cada objeto, resaltando su estructura en una cuadrícula.

## Resultados:

Cada imagen procesada produjo un gráfico donde se muestran los contornos de los objetos binarios en celdas cuadrangulares. Esto permite un análisis visual más detallado de la morfología de los objetos en la imagen.

Los resultados obtenidos pueden consultarse en la siguiente carpeta: mapas\_contornos.

## Cálculo del Centro de Masa y Momentos Centrales

Para cada objeto binario presente en las imágenes procesadas, se calculó el centro de masa utilizando los momentos espaciales de primer orden. Posteriormente, se trasladó la imagen de manera que el centro de masa coincidiera con un punto de referencia, y se calcularon los momentos centrales siguiendo la ecuación:

$$\mu_{pq} = \sum_{x=1}^M \sum_{y=1}^N (x - x_{cm})^p (y - y_{cm})^q$$

Estos momentos permiten analizar la distribución de la masa del objeto con respecto a su centro geométrico, lo que resulta útil para la normalización y comparación entre diferentes formas. A continuación, se presentan los valores obtenidos para cada imagen analizada:

**Tabla X.** Centro de masa y momentos centrales de los objetos binarios.

Imagen	x_cm	y_cm	mu_01	mu_10	mu_11	mu_20	mu_02
butterfly-2.png	290.406821	324.981102	-3.75E-09	4.44E-10	334972213	716917814	1950225432
cattle-3.png	429.861288	372.09851	5.39E-09	5.10E-10	2384762249	2942263399	4087293202
confusions.png	235.81314	171.411635	1.05E-09	-2.49E-09	13524424.1	2042641876	1171955460
deer-4.png	481.51134	505.315693	-1.96E-09	3.26E-09	344243854	6530707635	2131731163
dog-7.png	218.337558	210.375998	4.73E-11	8.55E-11	153805025	837943186	406938793
elephant-4.png	435.332855	462.655199	-1.37E-09	-1.14E-08	471103955	6931967430	3253911387
guitar-3.png	192.054351	388.905684	-1.38E-09	3.89E-10	613819438	460124779	1375538418
lizzard-19.png	268.257548	371.802155	-1.36E-09	-6.72E-10	-721073989	530381009	3114092251
pocket-11.png	183.227495	321.110801	3.03E-09	1.42E-09	-14198804.1	743791263	1410732898
shapedata.png	268.932707	192.090844	2.10E-09	6.98E-10	-33437795.3	2981439478	1606095152

## Cálculo de Momentos de Hu Antes y Después de una Rotación

Para cada objeto binario identificado en las imágenes, se realizó una transformación mediante una rotación de un ángulo  $\theta$  *theta*. Posteriormente, se calcularon los tres primeros momentos de Hu antes y después de la rotación.

Los momentos de Hu son invariantes geométricas utilizados para describir la forma de los objetos de manera independiente de su posición, escala y orientación. En este caso, se consideran las siguientes expresiones:

$$\varphi_1 = \mu_{20} + \mu_{02}$$

$$\varphi_2 = (\mu_{20} - \mu_{02})^2 + 4\mu_{11}^2$$

$$\varphi_3 = (\mu_{30} - 3\mu_{12})^2 + (3\mu_{21} - \mu_{03})^2$$

Estos valores permiten analizar la estabilidad de la forma de los objetos tras la transformación. Los resultados obtenidos se presentan en la siguiente tabla, en la que se comparan los momentos de Hu calculados para cada objeto antes y después de la rotación aplicada.

Imagen	Hu1_Original	Hu2_Original	Hu3_Original	Hu1_Rotado	Hu2_Rotado	Hu3_Rotado
butterfly-2.png	6.8012E+11	1.28E+23	3.43E+27	6.8006E+11	1.28E+23	3.43E+27
cattle-3.png	1.7925E+12	1.56E+24	2.97E+27	1.7925E+12	1.56E+24	2.98E+27
confusions.png	8.1972E+11	4.93E+22	3.88E+24	4.9929E+11	3.01E+22	3.07E+24
deer-4.png	2.2089E+12	1.29E+24	2.60E+28	2.209E+12	1.29E+24	2.60E+28
dog-7.png	3.1744E+11	1.82E+22	8.56E+26	2.7287E+11	1.97E+22	3.07E+26
elephant-4.png	2.5974E+12	9.37E+23	2.47E+28	2.5971E+12	9.37E+23	2.47E+28
guitar-3.png	4.6809E+11	1.52E+23	3.31E+27	1.1605E+11	4.57E+21	8.77E+25
lizzard-19.png	9.2934E+11	5.69E+23	2.92E+27	9.2934E+11	5.69E+23	2.92E+27
pocket-11.png	5.494E+11	2.90E+22	1.33E+27	4.1781E+11	2.17E+22	3.02E+26
shapedata.png	1.1698E+12	1.23E+23	1.28E+25	6.8913E+11	6.02E+22	1.03E+25

Además, los valores han sido almacenados en un archivo CSV para su posterior análisis.

## Aplicación de Operadores Morfológicos a los Objetos Binarios

### a) Quitar Ruido.

A partir de los operadores morfológicos previamente estudiados, el siguiente punto busca hacer uso de las características de estos operadores. Para este caso, la erosión permite afinar los bordes evitando que aquellos irregulares se ajusten a una forma suave.

#### b) Suavizar bordes.

Para lograr este efecto sobre la imagen, es necesario recurrir a la combinación de dos operadores morfológicos: erosión y dilatación. De esta forma, se elimina el ruido de los bordes y posteriormente se aumenta el ancho del borde resultante. Esta operación es conocida como apertura.

#### c) Rellenar Huecos.

Recurrimos a una combinación de operadores morfológicos: dilatación y erosión, lo cual es llamado "Clausura" o "Cierre". Mediante esta operación, el objeto cierra espacios pequeños y conecta bordes cercanos sellando espacios de la imagen original.

#### d) Mostrar esqueleto.

Para esta operación utilizamos en repetidas ocasiones un solo operador: erosión. Para obtener el esqueleto de la imagen basta con erosionar la imagen, sustraer el resultado con la imagen original y sustituir ese resultado con la imagen original hasta un punto definido. El resultado es un contorno mínimo que define la figura del objeto.

Para consultar los resultados, adjuntamos una carpeta con las diferentes imágenes modificadas: [https://github.com/DannyTM12/Graficacion Tarea 2](https://github.com/DannyTM12/Graficacion_Tarea_2)

## Conclusiones

En este trabajo se han aplicado diversas técnicas de procesamiento de imágenes para analizar objetos binarios a través de sus momentos invariantes, su centro de masa y su comportamiento ante transformaciones geométricas y morfológicas. A continuación, se presentan las conclusiones obtenidas a partir del análisis de los resultados.

### *Impacto de las transformaciones morfológicas en los objetos binarios*

Las operaciones morfológicas han permitido modificar la estructura de los objetos binarios mediante eliminación de ruido, suavización de bordes, relleno de huecos y obtención de esqueletos. Cada una de estas transformaciones introduce cambios en la imagen que pueden alterar las propiedades de los objetos analizados:

- **Eliminación de ruido:** Redujo la presencia de píxeles aislados que podrían haber influido en el cálculo de los momentos de Hu y otros parámetros geométricos.
- **Suavización de bordes:** Modificó la forma de los objetos, lo que puede afectar directamente a los momentos centrales y a la distribución de los píxeles en la imagen.
- **Relleno de huecos:** Transformó la estructura interna de los objetos, alterando la distribución del área y afectando los cálculos de centro de masa.
- **Obtención del esqueleto:** Redujo la imagen a su representación estructural mínima, lo que genera una transformación radical en los momentos de Hu y en otras métricas geométricas.

## ***Evaluación de la invariancia de los momentos ante transformaciones morfológicas***

Uno de los aspectos más relevantes del análisis es evaluar si las ecuaciones (1), (2) y (3) mantienen su invariancia después de aplicar operaciones morfológicas.

### **1. Ecuación (1) - Cálculo de los momentos invariantes de Hu**

Los momentos de Hu son teóricamente invariantes ante transformaciones geométricas como traslación, rotación y escalamiento. Sin embargo, las transformaciones morfológicas modifican la estructura del objeto, lo que puede alterar significativamente estos momentos. En particular, la eliminación de ruido tiene un impacto mínimo, pero operaciones como el suavizado de bordes o el esqueleto pueden alterar drásticamente los valores obtenidos.

### **2. Ecuación (2) - Cálculo del centro de masa y momentos centrales**

El centro de masa de un objeto se basa en la distribución de los píxeles dentro de la imagen. Transformaciones como el relleno de huecos o la eliminación de ruido pueden modificar la ubicación del centro de masa, afectando directamente los momentos centrales. Esto indica que esta ecuación no es invariante ante modificaciones morfológicas que alteren la densidad de los píxeles en la imagen.

### **3. Ecuación (3) - Cálculo de momentos de Hu antes y después de la rotación**

La rotación, como se ha comprobado, no altera los momentos de Hu debido a su propiedad de invariancia geométrica. No obstante, cuando se aplican operaciones morfológicas, la estructura del objeto cambia, lo que provoca una alteración en los momentos de Hu. En particular, el suavizado de bordes y la obtención de esqueletos generan modificaciones en la distribución de los píxeles, lo que conlleva variaciones en los momentos de Hu calculados.

## ***Conclusión general***

Los momentos de Hu cumplen su propiedad de invariancia bajo transformaciones geométricas como rotación y traslación, pero pueden verse afectados por transformaciones morfológicas que alteren la estructura del objeto. En contraste, el centro de masa y los momentos centrales no son invariantes bajo estas modificaciones, ya que dependen directamente de la distribución de los píxeles en la imagen.

En términos prácticos, esto implica que, si se busca utilizar los momentos de Hu para el reconocimiento de objetos, se deben aplicar con precaución en imágenes que hayan sido sometidas a transformaciones morfológicas. Además, si se busca analizar la estabilidad de los momentos centrales, es recomendable definir un conjunto de preprocesamientos que minimicen las alteraciones estructurales de los objetos antes de su análisis.

En conclusión, mientras que los momentos de Hu pueden conservar algunas de sus propiedades ante transformaciones geométricas estándar, las operaciones morfológicas introducen variaciones que deben ser consideradas en aplicaciones prácticas como el reconocimiento de patrones y la clasificación de formas.