



	Computación	Docente: Vladimir Robles Bykbaev
	VISIÓN POR COMPUTADOR	Período Lectivo: Octubre 2025 – Febrero 2026

		FORMATO DE GUÍA DE PRÁCTICA DE LABORATORIO / TALLERES / CENTROS DE SIMULACIÓN – PARA DOCENTES	
CARRERA: COMPUTACIÓN		ASIGNATURA: VISIÓN POR COMPUTADOR	
NRO. PRÁCTICA:	3-1 3-2	TÍTULO PRÁCTICA: Análisis comparativo de robustez entre Momentos de Hu y Zernike, y reconocimiento de formas mediante Firmas Espectrales con FFT de Coordenadas Complejas.	
OBJETIVO: Evaluar la robustez y eficiencia de descriptores globales para el reconocimiento de formas geométricas (círculo, triángulo, rectángulo) mediante el análisis de momentos ortogonales y la implementación de una firma de la figura basada en la transformada de Fourier de coordenadas complejas en un entorno móvil.			
INSTRUCCIONES:		1. Revisar el contenido teórico del tema	
		2. Profundizar los conocimientos revisando los libros guías, los enlaces contenidos en los objetos de aprendizaje y la documentación disponible en fuentes académicas en línea	
		3. Deberá desarrollar un conjunto de <i>scripts</i> y programas que permitan realizar reconocimiento de formas usando momentos de HU y Zernike y la firma de la figura (Shape Signature).	
		4. La práctica se subdividirá en tareas específicas que guardarán relación con la representación de varios conceptos como: momentos de Hu, momentos de Zernike y la firma de la figura (Shape Signature).	
ACTIVIDADES POR DESARROLLAR			
Objetivos específicos:			
<ul style="list-style-type: none">• Evaluar la invarianza a la rotación, escala y ruido de los Momentos de Hu frente a los Momentos de Zernike.• Implementar un sistema de reconocimiento de formas en tiempo real mediante Descriptores de Fourier (FFT) utilizando la variante de coordenadas complejas.• Analizar el impacto del ruido sintético en la degradación de características globales.			
PARTE 1. Cuaderno de Kaggle - Análisis de Robustez (Hu vs. Zernike)			
Desarrollar y publicar un cuaderno de Kaggle (<i>en Python</i>) clonando para ello el cuaderno “ UPS-Writing-Skills ”:			
<ul style="list-style-type: none">• Preprocesamiento y Aumento de Datos:			
1. Cargar las imágenes de las clases: <i>Circle</i> , <i>Triangle</i> , <i>Rectangle</i> como las que se muestran en la Ilustración 1 (disponibles en el enlace “ UPS-Writing-Skills ”):			
			
Círculo		Triángulo	
			
		Cuadrado	
Ilustración 1. Ejemplo de 3 categorías de imágenes del dataset “ <i>UPS-Writing-Skills</i> ” ¹			

¹El daset está disponible en la siguiente dirección:

<https://www.kaggle.com/datasets/adolfogavilanes/ups-writing-skills/data>

2. Diseñar una función de "Inyección de Ruido": Aplicar ruido tipo **Gaussian** y **Salt & Pepper** con 3 niveles de intensidad (Bajo, Medio, Alto).

3. Generar variantes con rotaciones aleatorias de 0° a 360° para cada imagen.

- **Extracción de Características:**

1. **Hu Moments:** Calcular los 7 momentos. Aplicar la transformación **log** (vista en clase) para mitigar la diferencia de órdenes de magnitud.

2. **Zernike Moments:** Implementar la extracción utilizando un grado de polinomio $n=8$ para capturar detalles finos.

- **Evaluación de Clasificación:**

1. Entrenar un clasificador simple (KNN o SVM) para cada tipo de descriptor.

2. Generar una **Matriz de Confusión** para cada nivel de ruido.


PARTE 2. Aplicación Móvil - Shape Signature (Coordenadas Complejas)

Debe crear una **aplicación móvil** en una librería nativa de C++ donde realizará todos los cálculos y referenciará al corpus y cuando se le muestre una imagen indicará qué tipo de figura es, presentando en pantalla la etiqueta correspondiente ("triángulo", "círculo" o "cuadrado"). Para ello, la aplicación móvil presentará un área donde se dibujará la forma y al presionar un botón (o componente similar) indicará qué tipo de forma es la que se ha dibujado (Ilustración 2).



Ilustración 2. Ejemplo de las opciones que debe tener la aplicación móvil para realizar la clasificación del patrón que se dibuja en el dispositivo.

Debe preprocesar las imágenes, convirtiéndolas en imágenes a blanco y negro, donde el trazo debe estar con color blanco y el fondo negro. Puede usar funciones de skeletización o rellenar el área interior de la figura geométrica. Debe buscar aplicar alguna técnica que permita mejorar los resultados. Se sugiere

	Computación	Docente: Vladimir Robles Bykbaev
	VISIÓN POR COMPUTADOR	Período Lectivo: Octubre 2025 – Febrero 2026

revisar el siguiente artículo:

- https://link.springer.com/chapter/10.1007/978-3-031-19647-8_22

Implementar en C++ (vía JNI para Android o Wrapper para iOS) el reconocimiento de formas basado en el contorno.

- **Variante Matemática (Coordenadas Complejas):** En lugar de medir distancias al centroide, el contorno se trata como una señal en el plano complejo:

$$s(n) = (x(n) - x_c) + j(y(n) - y_c)$$

- Donde (x_c, y_c) es el centroide de la figura. Esta representación es más rica ya que conserva la relación espacial completa entre los puntos del contorno.
- Flujo en Tiempo Real:
 - **Segmentación:** Binarización mediante Umbral Adaptativo.
 - **Contornos:** cv::findContours para extraer el contorno más externo.
 - **DFT:** Aplicar cv::dft a la señal compleja $s(n)$.
 - **Normalización:** Dividir los coeficientes por el primer armónico $|F(1)|$ para lograr invarianza a escala y descartar la fase para invarianza a la rotación.
- Considerar el uso de la librería [Albumentations](#) para generar más imágenes para el entrenamiento.
- Determinar el nivel de precisión para clasificar las imágenes al menos **30 imágenes dibujadas** de prueba (en el dispositivo móvil y luego guardadas) comparando el descriptor de cada imagen de entrenamiento con el descriptor de cada imagen del dataset. Para ello, puede usar la **distancia Euclídea** (u otra medida de similitud que desee) vista en clase (en el tema de los momentos de HU).
- Se sugiere revisar el cálculo de la transformada rápida de Fourier con la función propia de OpenCV en C++:

```
cv::Mat input = cv::Mat(signature).reshape(1, 1);
input.convertTo(input, CV_32F);
cv::Mat complexI;
cv::dft(input, complexI,
cv::DFT_COMPLEX_OUTPUT);
```

- Deberá generar un reporte que contenga la siguiente información:
 - Código fuente de la solución planteada.
 - Matriz de confusión de las categorías seleccionadas.
 - Nivel de precisión del sistema planteado.
 - Ejemplos gráficos de formas que confunde.

- Al igual que en el caso anterior, un ejemplo de dataset se puede observar en la Ilustración 3:












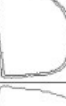












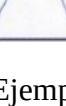


Original	Negative	Contour	Rating
			Score: 0 – The line is not closed and not predominantly circular.
			Score: 1 – There are no straight lines or angles; the two ends don't meet or cross.
			Score: 2 – The circle is made correctly; there are no straight lines or angles; the two ends meet.
			Score: 0 – The line is not closed and not predominantly circular.
			Score: 1 – Figure with 4 sides and 4 angles.
			Score: 2 – 4 right angles, between 80 and 100 degrees; closed figure without crossing lines; sides equal in length; lines without undulations.
			Score: 0 – The figure is not related to the proposed model.
			Score: 1 – The lines are not straight; the vertex is not to the center of the base.
			Score: 2 – Closed figure, with 3 well-defined sides of equal length and 3 vertices; lines without undulations.


Ilustración 3. Ejemplo completo de las figuras del dataset “[UPS-Writing-Skills](#)”.

RESULTADO(S) OBTENIDO(S):

Al finalizar esta práctica, los estudiantes habrán desarrollado la capacidad técnica para evaluar la **invarianza y robustez** de descriptores globales frente a condiciones adversas de ruido y transformaciones geométricas. Se espera que consoliden su habilidad para diferenciar el rendimiento entre momentos estadísticos (**Hu**) y momentos ortogonales (**Zernike**) mediante el análisis de datos en la plataforma Kaggle con el dataset *UPS-Writing-Skills*. Asimismo, habrán dominado la implementación de **Descriptores de Fourier (FFT)** en dispositivos móviles, utilizando la variante de **Coordenadas Complejas**, lo que les permite procesar contornos como señales en el plano complejo. Al cierre, el estudiante podrá diseñar sistemas de visión robustos que integran preprocesamiento avanzado, extracción de características espectrales y clasificación basada en métricas de similitud, validando sus resultados mediante matrices de confusión y análisis de dispersión de características.

CONCLUSIONES:

- Superioridad de la Ortogonalidad:** Se concluye que los **Momentos de Zernike** ofrecen una representación menos redundante y más robusta ante el ruido sintético en comparación con los Momentos de Hu, debido a su base de polinomios ortogonales que captura detalles de alta

	Computación	Docente: Vladimir Robles Bykbaev
	VISIÓN POR COMPUTADOR	Período Lectivo: Octubre 2025 – Febrero 2026

frecuencia de forma más eficiente.

- **Eficacia Espectral en Móviles:** La implementación de la **FFT de Coordenadas Complejas** demuestra ser una técnica altamente eficiente para entornos móviles, permitiendo una representación compacta del contorno que es intrínsecamente fácil de normalizar para lograr invarianza a escala y rotación sin un alto costo computacional.
- **Vínculo Teórico-Práctico:** La práctica logra conectar la teoría matemática de las transformadas de frecuencia y la geometría analítica con aplicaciones reales de asistencia educativa, transformando trazos manuales en datos digitales clasificables, lo que prepara al estudiante para retos complejos en IA y robótica.
- **Análisis de Degradación:** El uso de Kaggle para inyectar ruido controlado permite comprender que la calidad de la segmentación previa es crítica; incluso los descriptores más avanzados pierden eficacia si el preprocesamiento no logra preservar la topología del contorno.

RECOMENDACIONES:

- **Optimización de Armónicos:** Se recomienda a los estudiantes experimentar con el número de coeficientes de Fourier (armónicos) utilizados en la App móvil; usar muy pocos suaviza demasiado la forma (pierde detalles), mientras que usar demasiados introduce ruido de alta frecuencia en la clasificación.
- **Refinamiento del Preprocesamiento:** Ante niveles altos de ruido "Salt & Pepper" en el dataset de Kaggle, se sugiere implementar filtros de mediana antes de la extracción de momentos para evitar que los píxeles aislados distorsionen los cálculos del centroide.
- **Exploración de Métricas de Similitud:** Ampliar la evaluación en el móvil probando no solo la distancia Euclídea, sino también la **distancia de Manhattan** o la **correlación cruzada** para comparar las firmas de Fourier, identificando cuál ofrece una mejor separación entre clases similares (como rectángulos y cuadrados).
- **Normalización Estricta:** Asegurar que en la implementación de coordenadas complejas se realice la división por el componente $|\mathbf{F}(1)|$ (frecuencia fundamental) para garantizar que la aplicación reconozca los dibujos independientemente de qué tan cerca o lejos se encuentre el papel de la cámara.
- **Análisis Estadístico:** Utilizar herramientas de visualización como **PCA (Análisis de Componentes Principales)** en el cuaderno de Kaggle para observar visualmente cómo el ruido "mezcla" los clusters de las diferentes figuras geométricas.

Docente / Técnico Docente: Ing. Vladimir Robles, Bykbaev

Firma: _____