

CLASIFICADOR DE FIGURAS GEOMÉTRICAS TRAZADAS A MANO BASADO EN LÓGICA BORROSA

David Palomeque Mangut
dpalomeqma@unex.es

J. Álvaro Fernández Muñoz
jalvarof@unex.es

Dept. Ing. Eléctrica, Electrónica y Automática, Escuela de Ingenierías Industriales,
Universidad de Extremadura, Av. de Elvas s/n, 06006 (Badajoz)

Resumen

El uso de dispositivos periféricos electrónicos tales como digitalizadoras gráficas y pantallas táctiles, permite al usuario realizar de forma eficiente tareas como la inserción rápida de texto, el dibujo de figuras vectoriales mediante puntos, y la comunicación gestual mediante figuras simples, entre otras. En este trabajo se presenta el diseño de un clasificador software en línea para el reconocimiento de formas geométricas simples trazadas a mano con un único trazo. El sistema de clasificación utilizado se basa en la Lógica Borrosa. El sistema permite clasificar rectángulos, círculos, triángulos y líneas con independencia de su tamaño y orientación. Se han creado dos bases de datos de figuras trazadas a mano compuestas por 80 figuras cada una. La primera base de datos corresponde al conjunto de entrenamiento del clasificador, mientras que la segunda corresponde al conjunto de test. Se han obtenido tasas de reconocimiento del 100 % en el conjunto de entrenamiento y del 98.75 % en el conjunto de test.

Palabras clave: Clasificación en línea, Lógica Borrosa, Comunicación gestual táctil.

1. INTRODUCCIÓN

Los dispositivos electrónicos conocidos como digitalizadoras gráficas son actualmente la interfaz de entrada preferida por profesionales dedicados a la producción digital de modelos, planos técnicos, gráficos, fotografía y vídeo. El usuario de este tipo de dispositivos se beneficia de una interfaz hardware/software que permite acelerar en gran medida ciertas acciones repetitivas con un excelente control de la precisión. Otros dispositivos de desarrollo más reciente, como las pantallas tácti-

les, son el medio casi exclusivo de controlar los más modernos *smartphones* y *tablets*, cuyo uso se ha extendido de forma rápida y masiva estos últimos años al mundo educativo, hasta el punto de desplazar en muchas ocasiones al ratón como dispositivo señalador estándar.

El auge del uso de pantallas táctiles como interfaz de entrada en múltiples dispositivos, junto al amplio abanico de potenciales aplicaciones de los mismos, han generado una creciente demanda de algoritmos que faciliten el reconocimiento de los trazos marcados por el usuario al presionar sobre la pantalla. El reconocimiento de figuras geométricas sencillas a partir de algoritmos robustos, rápidos y suficientemente ligeros para permitir su uso en aplicaciones en línea es un primer paso en el camino de conseguir una interacción más fluida entre el usuario y el dispositivo táctil.

Una de las principales utilidades de estos algoritmos de reconocimiento de formas geométricas en dispositivos táctiles es el incremento de productividad a la hora de realizar diagramas de bloques y de control de flujo, cuyo coste en tiempo de edición es normalmente elevado [6]. Sin embargo, la interpretación de estas figuras como base de lenguajes gestuales multitrazo abre una interesante vía de desarrollo de aplicaciones táctiles para niños y personas discapacitadas.

El presente trabajo propone el desarrollo de un clasificador software optimizado para trabajo en línea, que permita categorizar de manera instantánea y transparente al usuario una serie de figuras geométricas, con robustez ante cambios en tamaño, orientación y grado de deformación. El desarrollo del clasificador se basa en la implementación de un Sistema de Lógica Borrosa (*Fuzzy Logic System*, FLS) [5]. El trabajo se organiza como sigue:

- En la Sección 1 se introducen brevemente una serie de conceptos teóricos (diferentes entidades matemáticas, descriptores de figuras geométricas, etc.), necesarios para comprender las bases de diseño del clasificador propuesto.
- En la Sección 2 se expone la metodología seguida en el diseño del clasificador borroso.
- La Sección 3 contiene un resumen de los resultados obtenidos, tanto en la fase de diseño como en la de testeo del sistema.
- Finalmente, en la Sección 4 se exponen las conclusiones extraídas del trabajo, con especial atención a posibles ampliaciones o modificaciones de interés para trabajos futuros.

2. CONCEPTOS PREVIOS

En esta sección se describen los conceptos matemáticos en los que se basan los descriptores de figuras geométricas utilizados en este trabajo, los propios descriptores y la estructura general del clasificador borroso diseñado.

2.1. DEFINICIONES

Los descriptores extraídos de las figuras geométricas dibujadas por el usuario requieren del cálculo de tres entidades geométricas: (1) la envolvente convexa o *convex hull*, (2) la caja límite mínima (*minimum bounding box*, MBB), y (3) el mayor triángulo inscrito en la envolvente convexa (*Maximum Inscribed Triangle*, MIT) de la figura.

La envolvente convexa C de un conjunto finito X , con k puntos de dimensión n , se define como la intersección de todas las combinaciones convexas que contienen a X . En una combinación convexa, se asigna un factor o peso no negativo α_i a cada punto $x_i \in X$, con $i \in [1, k]$, cuya suma es igual a la unidad, y permite calcular una media ponderada de todos los puntos. Para cada selección de coeficientes α_i , la combinación convexa resultante es un punto contenido en C . Por tanto, C puede calcularse como

$$C(X) = \left\{ \sum_{i=1}^k \alpha_i x_i : \alpha_i \geq 0 \wedge \sum_{i=1}^k \alpha_i = 1 \right\} \quad (1)$$

En el caso particular de puntos en el plano ($n = 2$), si todos los puntos de X forman una línea recta, C es el segmento lineal que une los dos puntos más

alejados. En el resto de casos, C es un polígono convexo cuyos vértices son algunos de los puntos del conjunto inicial de puntos, siendo el resto de puntos interiores a C [1].

Una forma intuitiva de visualizar la envolvente convexa de un conjunto de puntos en el plano, es imaginar una banda elástica que inicialmente los encierra a todos, y cuando se libera, toma la forma de la envolvente convexa.

La denominada caja límite mínima de un conjunto de k puntos X de dimensión n es aquella caja con medida mínima (área, volumen o hipervolumen) que contiene todos los puntos de X . A partir de la definición de C en (1), resulta evidente que el MBB de X es equivalente al MBB de $C(X)$, lo que suele utilizarse para acelerar su cómputo [4]. En este trabajo, solo será de interés el MBB para conjuntos con $n = 2$.

Por último, el mayor triángulo inscrito en la envolvente convexa (MIT) se define como aquel triángulo cuyos vértices x_a , x_b y x_c pertenecen a $C(X)$ y cuya área es máxima respecto a todos los triángulos inscritos en $C(X)$ posibles. En [2] se aborda el algoritmo de cálculo de esta entidad geométrica, así como sus limitaciones y tiempo de cómputo.

2.2. DESCRIPTORES DE FIGURAS GEOMÉTRICAS

Los descriptores seleccionados se han utilizado previamente en clasificadores de figuras geométricas de todo tipo [3]. En este trabajo se realizó una selección que, a priori, servían para realizar distinciones entre las figuras geométricas a clasificar. Sin embargo, solo fue posible comprobar el buen funcionamiento de estos descriptores en la etapa de diseño del clasificador.

El primero de los descriptores a definir es el *Thinness Ratio*, que se obtiene mediante la siguiente relación:

$$ThR = \frac{P_{ch}^2}{A_{ch}} \quad (2)$$

donde P_{ch} y A_{ch} son el perímetro y área de la envolvente convexa, respectivamente. Por definición, el *ThR* presenta valores muy cercanos a 4π en circunferencias y polígonos convexas con un gran número de vértices k , alejándose progresivamente de dicho valor para polígonos con k decreciente.

El descriptor *Line Ratio* se obtiene mediante la

siguiente expresión:

$$LiR = \frac{H_{bb}}{W_{bb}} \quad (3)$$

donde H_{bb} y W_{bb} son, respectivamente, la altura y anchura de de la MBB, siendo $H_{bb} < W_{bb}$. El LiR presenta valores muy bajos en segmentos lineales, aproximándose a 1 en polígonos convexos regulares y circunferencias.

El descriptor *Rectangle Ratio* se define mediante:

$$RR = \frac{A_{ch}}{A_{bb}} \quad (4)$$

donde A_{ch} y A_{bb} son, respectivamente, el área de la envolvente convexa y del MBB. El RR presenta valores muy cercanos a 1 para rectángulos.

El *Triangle Ratio 1* es un descriptor orientado a triángulos, que responde a la ecuación:

$$TR1 = \frac{A_{lt}}{A_{ch}} \quad (5)$$

donde A_{lt} el área del MIT. El $TR1$ presenta valores muy cercanos a 1 para triángulos.

Finalmente, el *Triangle Ratio 2* se obtiene mediante la siguiente expresión:

$$TR2 = \frac{P_{lt}}{P_{ch}} \quad (6)$$

donde P_{lt} y P_{ch} son el perímetro del MIT y de la envolvente convexa, respectivamente. Al igual que el descriptor $TR1$, $TR2$ presenta valores muy cercanos a 1 para los triángulos.

2.3. ESTRUCTURA DEL CLASIFICADOR BORROSO

Los datos de entrada al clasificador consisten en un vector de coordenadas XY de longitud variable que contiene los puntos de la forma geométrica introducida por el usuario. La salida del sistema es la categoría (línea, circunferencia, rectángulo o triángulo) que se asigna a la forma geométrica dibujada.

Así, el clasificador borroso comprende dos etapas fundamentales: una primera etapa de preprocesamiento de los datos de entrada y una segunda etapa de clasificación con el FLS. El preprocesamiento de los datos consiste en la aplicación de los descriptores definidos en la Sección 2 a los datos de entrada. Estos descriptores constituyen las variables de entrada del FLS, el cual se divide a su

vez en tres partes: (1) fuzzyficador, (2) sistema de inferencias borroso (*Fuzzy Inference System*, FIS), que está gobernado por una serie de reglas lógicas, y (3) defuzzyficador.

El fuzzyficador se encarga de transformar las variables de entrada (los descriptores) en variables borrosas que activarán las reglas con base lingüística del FIS. Este se limita a convertir las reglas con base lingüística a reglas borrosas mediante la implicación borrosa. La función de pertenencia de cada regla se obtiene como el producto de cada una de las implicaciones. A partir estas salidas, el defuzzyficador obtendrá la categoría a la que se asigna la figura dibujada.

3. DISEÑO DEL CLASIFICADOR

El diseño del clasificador comprende tres etapas fundamentales: (1) diseño del fuzzyficador, es decir, definición de las funciones de pertenencia de las variables de entrada, (2) definición de las reglas del FIS y (3) diseño del defuzzyficador, es decir, selección del método de cálculo de la categoría de clasificación a partir de las salidas del FIS.

3.1. DISEÑO DEL FUZZYFICADOR Y DEFINICIÓN DE LAS REGLAS DEL FIS

Para establecer las funciones de pertenencia de las variables de entrada se observan los valores adoptados por los descriptores de las figuras geométricas en el conjunto de entrenamiento. El conjunto de entrenamiento está extraído de una base de datos propia, formada por 80 figuras trazadas por una única persona a través de una herramienta programada sobre MATLAB. La información extraída de estos descriptores servirá también para diseñar las reglas del FIS. Así, el diseño de estas dos etapas del FLS se realiza de manera iterativa, comprobando qué funciones de pertenencia y reglas del FIS dan un mejor resultado en la clasificación del conjunto de entrenamiento.

3.2. DISEÑO DEL DEFUZZYFICADOR

Para obtener la salida del clasificador basta con seleccionar la regla borrosa cuya respuesta tiene un valor más alto, es decir, se obtiene la salida como el máximo de las funciones de pertenencia de salida.

4. RESULTADOS

4.1. FUNCIONES DE PERTENENCIA Y REGLAS OBTENIDAS

En las Figuras Figuras 1, 2, 3, 4 y 5 se ilustra mediante diagrama de cajas la distribución de valores obtenidos en los descriptores ThR (2), LiR (3), RR (4), TR1 (5) y TR2 (6), para cada una de las categorías consideradas en el conjunto de entrenamiento.

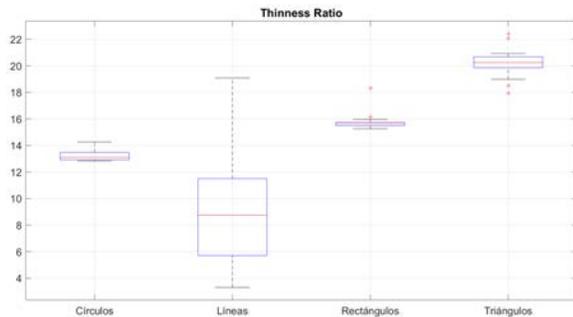


Figura 1: Distribución del parámetro *Thinness Ratio* en cada categoría en el conjunto de entrenamiento

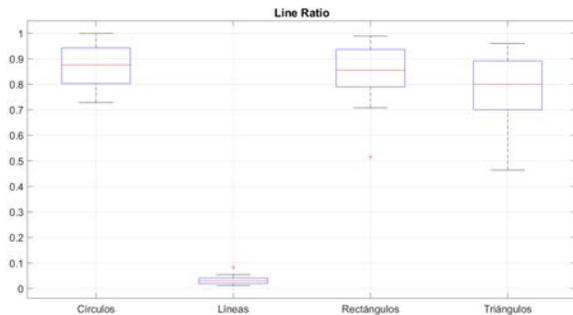


Figura 2: Distribución del parámetro *Line Ratio* en cada categoría en el conjunto de entrenamiento

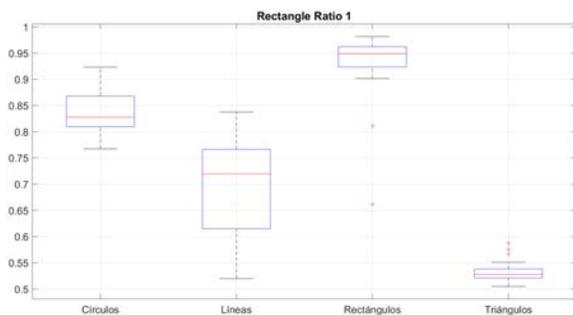


Figura 3: Distribución del parámetro *Rectangle Ratio* en cada categoría en el conjunto de entrenamiento

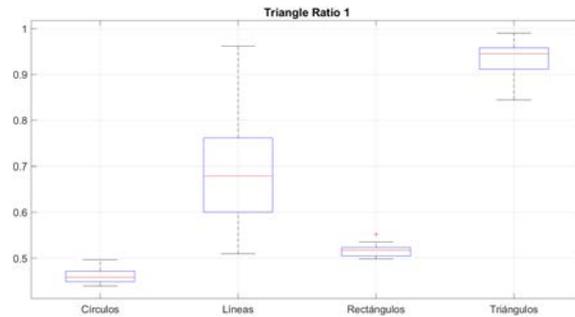


Figura 4: Distribución del parámetro *Triangle Ratio 1* en cada categoría en el conjunto de entrenamiento

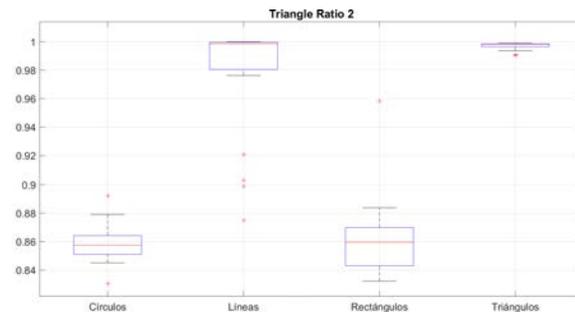


Figura 5: Distribución del parámetro *Triangle Ratio 2* en cada categoría en el conjunto de entrenamiento

Los valores obtenidos del descriptor *ThR* a partir de las figuras de categoría *Línea* se han dividido entre 25 con el objeto de hacer una representación ilustrativa de cómo se distribuye el parámetro en todas las categorías.

Teniendo en cuenta la distribución de los valores de los parámetros para cada una de las categorías, las funciones de pertenencia se han establecido como trapezoidales y no ha sido necesario definir las en todas las categorías de todas las variables de entrada para construir un clasificador robusto y eficaz. Los parámetros de estas funciones de pertenencia trapezoidales (a, b, c y d) adoptan los valores que se muestran en la Tabla 1.

Por otra parte, a partir de las funciones de pertenencia construidas y la distribución de los descriptores, se establecen una serie de reglas lingüísticas que definen el funcionamiento del FIS.

- Regla 1: SI ThR es CÍRCULO, ENTONCES salida es CÍRCULO.
- Regla 2: Si ThR es LÍNEA y LiR es LÍNEA, ENTONCES salida es LÍNEA.

- Regla 3: SI RR1 es RECTÁNGULO, ENTONCES salida es RECTÁNGULO.
- Regla 4: Si TR1 es TRIÁNGULO y TR2 es TRIÁNGULO y ThR no es LÍNEA y LiR no es LÍNEA, ENTONCES salida es TRIÁNGULO.

4.2. EJEMPLO DE CLASIFICACIÓN DE TRIÁNGULO TRAZADO A MANO

Con el objeto de ilustrar el funcionamiento del clasificador diseñado, se toma un triángulo de la base de datos correspondiente al conjunto de test. Este conjunto está formado por 80 figuras dibujadas por tres personas distintas con la herramienta creada en MATLAB. En la Figura 6 se representa el triángulo trazado a mano.

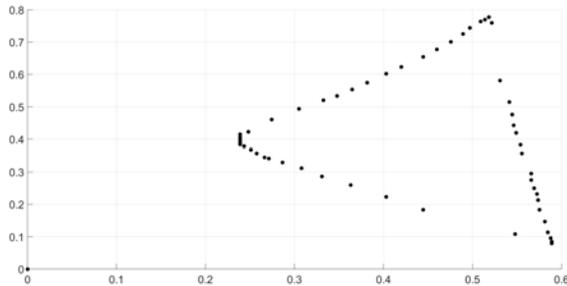


Figura 6: Representación de un triángulo trazado a mano incluido en el conjunto de entrenamiento.

La aplicación de los descriptores definidos en la Sección 1 a esta figura arroja los resultados mostrados en la Tabla 2.

Puede observarse cómo los descriptores TR1 y TR2 toman valores muy cercanos a 1 para un triángulo trazado a mano con ciertas “irregularidades”, lo que demuestra que los descriptores

Tabla 1: Valores fijados a los parámetros de las funciones de pertenencia trapezoidales.

Variable de entrada	Parámetros			
	a	b	c	d
ThR (Círculos)	12.2	12.4	14.5	15
ThR (Líneas)	55	60	800	1000
LiR (Líneas)	0	0	0.1	0.15
RR (Rectángulos)	0.65	0.85	1	1.2
TR1 (Triángulos)	0.8	0.9	1.1	1.2
TR2 (Triángulos)	0.98	0.99	1.1	1.2

Tabla 2: Valores que toman los descriptores en el triángulo de ejemplo.

Descriptor	Valor
ThR	22.61
LiR	0.45
RR	0.54
TR1	0.92
TR2	0.99

funcionan notablemente bien. Estos valores conforman el vector de datos de entrada del clasificador, cuyo FIS proporciona cuatro salidas, tal y como se introdujo en la Sección 1. Los valores que adoptan estas salidas se muestran en la Tabla 3.

Tabla 3: Valores que toman las salidas del FIS.

Regla	Valor
Regla 1 (Círculos)	0
Regla 2 (Línea)	0
Regla 3 (Rectángulos)	0
Regla 4 (Triángulos)	1

Puede observarse cómo el clasificador asigna valores muy bajos a las categorías erróneas, asignando a la categoría Triángulo la salida de mayor valor.

4.3. RESULTADOS DE CLASIFICACIÓN

A partir de las funciones de pertenencia de entrada y las reglas obtenidas, se aplica el clasificador a conjunto de entrenamiento, lo cual permite conocer el error de entrenamiento del clasificador. Los resultados se muestran en forma de matriz de confusión en la Tabla 4.

Por otra parte, el clasificador diseñado también se ha aplicado al conjunto de test. La clasificación del conjunto de test permite obtener el error de generalización o de test. Los resultados se muestran en la Tabla 5.

5. CONCLUSIONES Y TRABAJOS FUTUROS

- Se ha diseñado un clasificador fuzzy muy sencillo que permite clasificar correctamente geometrías del tipo línea, circunferencia, rectángulo y triángulo que, además, soporta figuras de cualquier escala y orientación.
- El clasificador muestra una tasa de aciertos

Tabla 4: Matriz de confusión obtenida a partir del conjunto de entrenamiento.

	Categoría asignada			
	Círculos	Líneas	Rectángulos	Triángulos
Círculos	20	0	0	0
Líneas	0	20	0	0
Rectángulos	0	0	20	0
Triángulos	0	0	0	20

Tabla 5: Matriz de confusión obtenida a partir del conjunto de test.

	Categoría asignada			
	Círculos	Líneas	Rectángulos	Triángulos
Círculos	20	0	0	0
Líneas	0	20	0	0
Rectángulos	0	0	20	0
Triángulos	0	0	1	19

del 100 % y de un 98.75 % para los conjuntos de entrenamiento y de test, respectivamente. El error de entrenamiento es por lo tanto nulo, mientras que el error de generalización es muy bajo.

- Se han implementado sobre MATLAB distintos algoritmos que permiten extraer parámetros de las figuras dibujadas.

Agradecimientos

Este trabajo ha sido financiado por la Junta de Extremadura a través de los Fondos Europeos de Desarrollo Regional (FEDER-GR10097).

Referencias

- [1] Andrew, A. M. (1979) "Another efficient algorithm for convex hulls in two dimensions", *Information Processing Letters*, 9(5), pp 216-219. DOI:10.1016/0020-0190(79)90072-3
- [2] Dobkin, D. P., Snyder, L. (1979) "On a general method for maximizing and minimizing among certain geometric problems", *Proc. 20th Annual Symposium on Foundations of Computer Science*, pp 9-17. DOI:10.1109/SFCS.1979.28
- [3] Fonseca, M. J., Jorge, J. A. (2000) "Using fuzzy logic to recognize geometric shapes interactively", *Proc. 9th IEEE International Conference on Fuzzy Systems*, pp 291-296. DOI:10.1109/FUZZY.2000.838674
- [4] O'Rourke, J. (1985) "Finding minimal enclosing boxes", *International Journal of Computer and Information Sciences*, 14(3), pp 183-199. DOI:10.1007/BF00991005
- [5] Mendel, J. M. (1995) "Fuzzy Logic Systems for Engineering: A tutorial", *Proceedings of the IEEE*, 83(3), pp 345-377. DOI:10.1109/5.364485
- [6] Refaat, K. S., Helmy, W. N., Ali, A. H., AbdelGhany, M. S., Atiya, A. F. (2008) "A new approach for context-independent handwritten offline diagram recognition using support vector machines", *Proc. IEEE International Joint Conference on Neural Networks*, pp 177-182. DOI:10.1109/IJCNN.2008.4633786