

Clasificación de formas por códigos de cadena mediante un algoritmo de búsqueda

Yoselim Cruz Sandoval, José Federico Ramírez Cruz,
Baldemar Zurita Islas, José Crispín Hernández Hernández

Instituto Tecnológico de Apizaco,
División de Estudios de Posgrado e Investigación,
México

{cruz.yoselim, baldemar.zurita}@gmail.com,
{federico_ramirez, josechh}@yahoo.com

Resumen. La clasificación de formas es un proceso utilizado en diversas áreas, para realizar una clasificación a partir de la manipulación de contornos de formas es necesario la extracción de ciertas características comunes para ello se utilizan diferentes técnicas, una de estas es el uso de algoritmos de búsqueda los cuales se encargan de buscar un elemento con ciertas propiedades dentro de una estructura de datos. En este trabajo se presenta una clasificación de formas usando el algoritmo de búsqueda A* y cadenas de código de Freeman, el algoritmo de búsqueda A* busca los pixeles que contienen la información que representa el borde de la forma, esta información es procesada para convertirse en códigos de cadena de Freeman que representan eficientemente los bordes de las formas geométricas, estas técnicas son probadas con diversas formas para comparar las cadenas obtenidas en cada una de ellas y comprobar que se trata de la misma forma con la que se trabajó inicialmente antes de aplicarle las técnicas propuestas.

Palabras clave: algoritmo de búsqueda, códigos de cadena, clasificación.

Classification of Shapes by Chain Codes using a Search Algorithm

Abstract. The classification of shapes is a process used in various areas, to make a classification from the manipulation of contours of shapes is necessary to extract certain common characteristics, for this different techniques are used, one of these is the use of search algorithms which are responsible of search an element with certain properties within a data structure. In this paper we present a classification of shapes using the search algorithm A* and Freeman chains code, the search algorithm A* searches the pixels that contain the information that represents the edge of the shape, this information is processed to become Freeman chain codes that efficiently represent the edges of geometric shapes, these techniques are tested with different shapes to compare the chains obtained in each of them and verify that it is the same shape which we initially worked before applying the proposed techniques.

Keywords: search algorithm, chain codes, classification.

1. Introducción

La clasificación de formas es un problema intrigante y desafiante que se encuentra en el cruce de la visión por computadora, el procesamiento de la geometría y el aprendizaje automático [1]. La forma es una característica intrínseca para la comprensión de la imagen, que es estable a la iluminación y las variaciones en el color y la textura del objeto. Debido a estas ventajas, la forma se considera ampliamente para el reconocimiento de objetos [2]. Los contornos de una forma son características principales y de gran importancia para su clasificación, a partir de estas características podremos describir la forma.

La forma es una señal importante en la percepción humana para el reconocimiento de objetos. Los objetos que no cuentan con brillo, color e información de textura y solo están representados por su silueta, no son difíciles de reconocer por los humanos. Esta simple demostración indica que la forma es estable a las variaciones en el color del objeto, la textura y las condiciones de luz.

El reconocimiento de formas generalmente se considera como un problema de clasificación al que se le da una forma de prueba, para determinar su etiqueta de categoría basada en un conjunto de formas de entrenamiento. Los principales desafíos en el reconocimiento de formas son las grandes variaciones intraclase inducidas por la deformación, la articulación y la oclusión [3].

La representación eficiente de la información para la clasificación de formas es importante. Los códigos de cadena se han convertido en los métodos de representación popular en diversas disciplinas científicas y de ingeniería [4], es por ello que para poder realizar de manera correcta la extracción de características de un contorno es necesario aplicar varias técnicas para finalmente representar el contorno mediante un código de cadena.

2. Marco teórico

2.1. Algoritmo A*

El algoritmo A* combina las ventajas del algoritmo Dijkstra y el algoritmo Best-First-Search. Este algoritmo no sólo intenta tomar el paso más corto entre cada movimiento, sino también le importa el paso de elección ya sea en la dirección que es solo de la fuente al objetivo [5].

2.2. Regresión lineal

Los modelos de regresión lineal son usados frecuentemente para la exploración de la relación entre un resultado continuo y variables independientes.

La regresión es una tarea de aprendizaje supervisado para inferir una relación funcional subyacente $y = f(x) + e$ a partir de un conjunto de instancias de entrenamiento $D = \{(x_i, y_i)\}_{i=1}^N$, donde x_i es un vector de variables de entrada y y_i es el valor correspondiente de una salida variable continua.

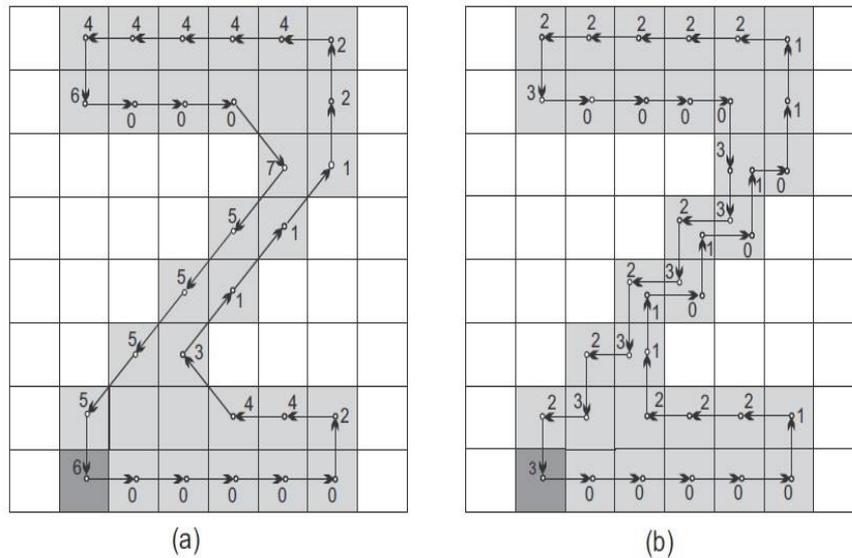


Fig. 1. Códigos cadena Freeman: (a) F8 y (b) F4 [4].

Tan pronto como el trabajo sobre la regresión se basó en modelos lineales, se limitaron a resolver problemas de regresión más complejos, particularmente cuando la relación entre las variables de entrada y salida no era lineal [6].

2.3. Códigos de cadena de Freeman

Es un método que permite la codificación de configuraciones geométricas arbitrarias para facilitar su análisis y manipulación por medio de una computadora digital [7].

El código de cadena se puede considerar como una secuencia de comandos, que controlan el movimiento de un andador virtual en todos los píxeles de contorno de una forma.

El código de cadena más intuitivo fue propuesto por Freeman en 1961 usando la conectividad de 8 píxeles (ver Fig. 1a) conocida como código de cadena de 8 direcciones de Freeman (es decir, F8).

El movimiento a través de los píxeles del límite está codificado con un alfabeto $\Sigma(F8) = \{0,1,2,3,4,5,6,7\}$, donde cada elemento $\sigma \in \Sigma$ representa un $45^\circ \times \sigma$ ángulo desde la dirección positiva del eje de coordenadas x.

Freeman también determinó que los límites de las formas digitalizadas pueden describirse mediante la conectividad de 4 píxeles (código de cadena de 4 direcciones de Freeman (F4)) codificando $90^\circ \times \sigma$ ángulo usando el alfabeto más corto $\Sigma(F4) = \{0,1,2,3\}$ (ver Fig. 1b) [8].

Tabla 1. Resultados de la aplicación del algoritmo de búsqueda a la forma geométrica de un triángulo donde “X” y “Y” son las posiciones en un plano cartesiano y “GP” es la posición global.

X	70	69	68	67	66	65	64	63	62	61	60
Y	10	11	12	12	13	14	14	15	16	16	17
GP	961	1059	1157	1156	1254	1352	1351	1449	1547	1546	1644

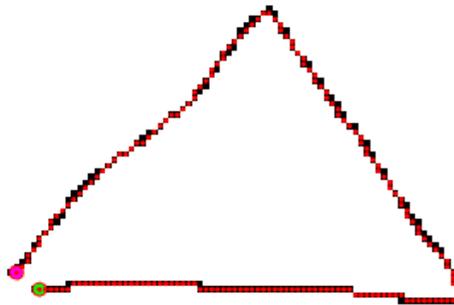


Fig. 2. Visualización de la ruta trazada sobre triángulo original al aplicar el algoritmo A* indicándole el punto de inicio (punto de color rosa) y el punto final (punto de color verde).

3. Métodos propuestos

3.1. Algoritmo de búsqueda A*

El análisis de vecindad de píxel se utiliza para la extracción de contorno, donde un píxel se considera un píxel de contorno si tiene al menos un vecino de fondo [9], para poder obtener las coordenadas de estos píxeles que conforman el contorno de la figura se aplicó en algoritmo de búsqueda A*.

Los resultados obtenidos después de aplicar el algoritmo de búsqueda A* son dados en la Tabla 1 en la cual se muestran los primeros resultados de la lista de posiciones en el plano cartesiano, también podemos visualizar cómo el algoritmo realiza el trazado de la trayectoria donde encuentra píxeles que contiene el contorno de la forma (ver Fig. 2).

En general, las representaciones de forma de corriente principal existentes se pueden clasificar en dos clases: basadas en el contorno y en el esqueleto. El primero entrega la información de cómo la distribución espacial de los puntos de frontera varía a lo largo del contorno del objeto. Por lo tanto, captura información de forma más informativa y es estable para la transformación afín. Sin embargo, es sensible a la deformación [3].

3.2. Regresión lineal

La regresión lineal se aplica cada 5 puntos, es decir, cada 5 coordenadas (ver Fig. 3). No se aplicará la regresión lineal siempre y cuando las coordenadas $y_{inicial}$ sea igual a y_{final} es decir $y_1 = y_5$ o cuando $x_{inicial}$ sea igual a x_{final} es decir $x_1 = x_5$.

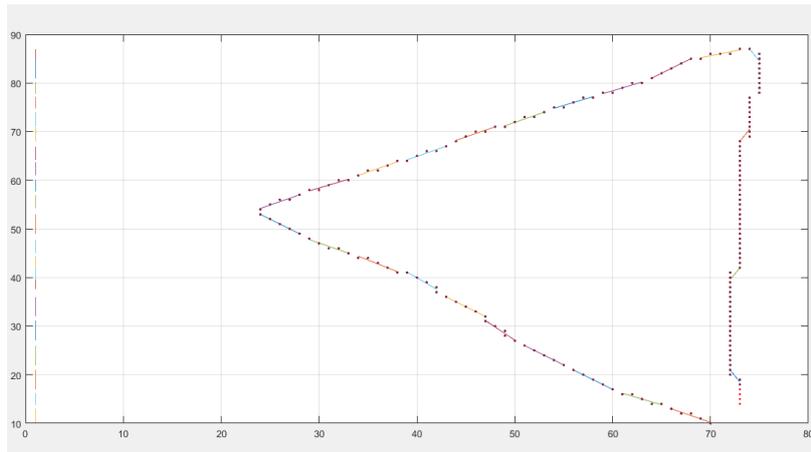


Fig. 3. Regresión lineal aplicada sobre las coordenadas que dibujan el triángulo.

Tabla 2. Condiciones aplicadas en los casos donde el ángulo resultante es menor a cero.

Condición	Orientación	Suma de ángulos
$x_{final} > x_{inicial}$ $y_{final} > y_{inicial}$		Ángulo resultante + 180
$x_{final} < x_{inicial}$ $y_{final} > y_{inicial}$		Ángulo resultante + 180
$x_{final} < x_{inicial}$ $y_{final} < y_{inicial}$		Ángulo resultante + 360
$x_{final} > x_{inicial}$ $y_{final} < y_{inicial}$		Ángulo resultante + 360
$x_{final} == x_{inicial}$ $y_{final} < y_{inicial}$		Ángulo resultante + 360
$x_{final} == x_{inicial}$ $y_{final} > y_{inicial}$		Ángulo resultante + 180

Tabla 3. Condiciones aplicadas en los casos donde el ángulo resultante es mayor a cero.

Condición	Orientación	Suma de ángulos
$x_{final} == x_{inicial}$ $y_{final} > y_{inicial}$	↑	Ángulo resultante = Ángulo resultante
$x_{final} == x_{inicial}$ $y_{final} < y_{inicial}$	↓	Ángulo resultante + 180
$x_{final} < x_{inicial}$ $y_{final} < y_{inicial}$	↙	Ángulo resultante + 180
$x_{final} > x_{inicial}$ $y_{final} < y_{inicial}$	↘	Ángulo resultante + 180

Tabla 4. Condiciones aplicadas en los casos donde el ángulo resultante es mayor e igual a cero.

Condición	Orientación	Suma de ángulos
$x_{final} < x_{inicial}$ $y_{final} == y_{inicial}$	←	Ángulo resultante + 180

3.3. Cálculo de ángulos

Una característica importante de una recta es su ángulo de inclinación, en nuestro caso este ángulo se obtendrá de cada pequeña recta obtenida con la regresión lineal para el caso de que existan puntos dispersos. Para poder calcular los ángulos de inclinación de las rectas de colores (ver Fig. 3) se calcula la inversa de la tangente del valor de la pendiente, es decir, $\alpha = \text{ang tan}(m) = \text{ang tan}\left(\frac{y_2 - y_1}{x_2 - x_1}\right)$, de este modo obtenemos el ángulo de la recta.

Es importante recalcar que el ángulo de inclinación depende de la orientación hacia donde se esté realizando el trazado de la figura, dicho de otra manera, la orientación que tiene el punto de inicio con respecto al punto final, para ello se realizaron tres tablas para estos casos particulares.

En la Tabla 2 se muestran los casos donde el ángulo obtenido es menor a cero, en Tabla 3 donde el ángulo es mayor a cero y en la Tabla 4 donde el ángulo es mayor o igual a cero. Para todos los resultados que no se encuentren dentro de estas condiciones el ángulo resultante quedara igual.

3.4. Códigos de cadena de Freeman

Los códigos cadena Freeman es un tipo de estructura de datos para representar el contorno de un objeto en una imagen binaria mediante una secuencia de segmentos, conectados consecutivamente, de longitud y orientación específica, que conectan

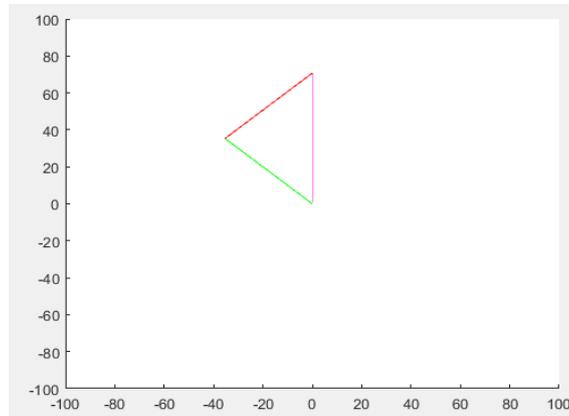


Fig. 4. Triángulo trazado a partir del código de cadena obtenido.

Tabla 5. Resultados obtenidos de figuras a las que se le aplicó la metodología propuesta.

Forma original	Regresión lineal	Códigos cadena	Forma hecha a partir de los códigos cadena
		[2 2 2 2 2 2 2 2 0 0 0 0 0 0 0 5 5 5 5 5 5 5 5 5]	
		[7 7 7 7 7 7 7 1 1 1 1 1 1 1 1 1 3 3 3 3 3 3 3 3 3 5 5 5 5 5 5 5]	
		[1 1 1 1 1 1 1 1 1 4 4 4 4 5 5 5 5 6 6 6]	
		[0 0 0 1 1 1 1 1 1 2 2 2 3 4 4 4 4 5 5 5 5 5 6 6 6 6]	

píxeles adyacentes, para la solución de este problema se utilizó Freeman de 8 direcciones. Un ejemplo de la cadena resultante obtenida para el caso del triángulo es: [2 2 2 2 2 2 2 2 2 0 0 0 0 0 0 0 0 0 5 5 5 5 5 5 5 5 5 5 5].

Para corroborar que la cadena obtenida anteriormente se trata en realidad de un triángulo ahora se realiza el proceso a la inversa, es decir, que a partir de esta cadena dibujaremos la figura geométrica (ver Fig.4).

4. Resultados

Con las técnicas propuestas, se muestran en la Tabla 5 otros ejemplos de formas geométricas a las que se les aplicó las técnicas para la extracción y representación de bordes mediante códigos cadena.

Se puede observar desde la imagen original hecha a mano alzada con la que trabajo el algoritmo de búsqueda, los códigos cadena Freeman y estos mismos representados para comprobar que se trata de la misma forma inicial.

5. Conclusiones

Las técnicas aplicadas a las formas geométricas dieron como resultado códigos cadena que representan de manera correcta las características de los bordes de cada forma.

Una técnica importante para la obtención correcta de códigos cadena fue el cálculo de regresión lineal que permite que las líneas que contienen el contorno de la forma sean mejoradas para que sean clasificadas de manera correcta dentro los códigos cadena de Freeman.

Los resultados de cada figura obtenidos indican que para las diferentes figuras los códigos cadena serán distintos ya que representan una trayectoria distinta de los bordes debido a que no todas las figuras son iguales.

Es importante decir que una misma figura puede tener un código cadena distinto ya que la trayectoria que recorrerá el algoritmo de búsqueda A* no será la misma; por ejemplo, para el caso particular de un cuadrado, no dará el mismo resultado iniciar en la parte superior izquierda hacia abajo de la forma o en la parte inferior derecha hacia arriba, cada uno de estas trayectorias recorridas nos darán resultados distintos de códigos de cadenas que nos representarían una misma figura geométrica.

Los códigos de cadena no se utilizan solo para representar los límites de formas geométricas, sino que sirven para diversas operaciones en ellos, como, por ejemplo: registro de imágenes, representación de funciones de forma libre, estimación de propiedades físicas, representaciones de fuentes en sistemas integrados, reconocimiento de caracteres escritos a mano, descripción del eje medial, y otros.

Tradicionalmente, los códigos de cadena son los más utilizados en el procesamiento de imágenes [10], es por ellos que se pretenden utilizar estos códigos cadena para crear una base de entrenamiento con etiquetas y posteriormente poder usar esta información en una red neuronal o un clasificador bayesiano ingenuo para la predicción de formas.

Referencias

1. Hamza, A.B.: A graph-theoretic approach to 3D shape classification. *Neurocomputing* 211, pp. 11–21 (2016)

2. Wang, X., Feng, B., Bai, X., Liu, W., Lateck, L.J.: Bag of contour fragments for robust shape classification. *Pattern Recognition* 47(6), pp. 2116–2125 (2014)
3. Shen, W., Jiang, Y., Gao, W., Zeng, D., Wang, X.: Shape recognition by bag of skeleton-associated contour parts. *Pattern Recognition Letters* 83, pp. 321–329 (2016)
4. Žalik, B., Mongus, D., Rizman, K. Ž, Lukač, N.: Chain code compression using string transformation techniques. *Digital Signal Processing* 53, pp. 1–10 (2016)
5. Lei, N. Guobin, Z.: An improved real 3D A algorithm for difficult path finding situation. In: *Proceeding of the International Archives of the Photogrammetry, Remote Sensing and Spatial Information Sciences*, 37 (2008)
6. Seokho, K. Pilsung, K. Locally linear ensemble for regression. *Information Sciences* 432, pp. 199–209 (2018)
7. Freeman, H.: On the encoding of arbitrary geometric configurations. *IRE Transactions on Electronic Computers* 2, pp. 260–268 (1961)
8. Žalik, B., Mongus, D., Lukač, N.: A universal chain code compression method. *Journal of Visual Communication and Image Representation* 29, pp. 8–15 (2015)
9. Chatbri, H., Kameyama, K., Kwan, P.: A comparative study using contours and skeletons as shape representations for binary image matching. *Pattern Recognition Letters* 76, pp. 59–66 (2016)
10. Žalik, B., Mongus, D., Yong-Kui, L., Lukač, N.: Unsigned Manhattan chain code. *Journal of Visual Communication and Image Representation* 38, pp. 186–194 (2016)