

Representación de Formas Digitales para Reconocimiento y Clasificación de Objetos

María L. Bergamini^{1§}, Jorge A. Kamlofsky¹

Fecha de Recibido: 10/12/2014

Fecha de Aprobación: 20/02/2015

Resumen

En este trabajo se proponen e implementan herramientas de análisis de imágenes orientadas al desarrollo de algoritmos para reconocimiento y clasificación de objetos en imágenes digitales. El análisis se aplica a curvas que constituyen el borde de un objeto digital. Se propone una representación de formas planas basada en una aproximación poligonal de curvas, lo que reduce la cantidad de datos a manejar. Una silueta se representa con un patrón de curvatura que mide el cambio del vector tangente a lo largo del perímetro. Este patrón permite determinar características geométricas de la forma, como son concavidad, convexidad, puntos de inflexión, entre otros. Los puntos de inflexión se toman como puntos de referencia para dividir la curva en partes significativas. La identificación de objetos se realiza por comparación de patrones, de acuerdo a una métrica apropiada; que puede ser aplicada al patrón completo o a una o varias partes significativas. De esta manera se pueden medir similitudes entre formas de objetos en imágenes digitales, así como diferencias en ciertas partes significativas, permitiendo la clasificación de objetos del mismo tipo.

Palabras clave: *Curvas digitales, Poligonalización, Invariancia, Patrón de curvatura.*

Abstract

Image analysis tools oriented to the development of algorithms for object recognition and classification in digital images are proposed and implemented in this work. The analysis is applied to the boundary curves of digital objects. A planar shape representation is proposed, based on polygonal approximation of curves, which significantly reduces the amount of data to handle. A silhouette is represented by a turning pattern that measures the change of the tangent vector along the perimeter. This pattern allows to determine geometric characteristics of the shape,

¹ CAETI – Facultad de Tecnología Informática - Universidad Abierta Interamericana – Montes de Oca 745, Ciudad Autónoma de Buenos Aires, Argentina {Maria.Bergamini, Jorge.Kamlofsky}@uai.edu.ar

[§] Se concede autorización para copiar gratuitamente parte o toda el material publicado en la *Revista Colombiana de Computación* siempre y cuando las copias no sean usadas para fines comerciales, y que se especifique que la copia se realiza con el consentimiento de la *Revista Colombiana de Computación*

such as concavity, convexity, inflection points, among others. Inflection points are considered as reference point to divide the curve in significant sections. Object identification is performed by pattern matching, according to a suitable metric; applied to the entire pattern or to one or more significant parts. Thus, similarities in shapes of objects in digital images can be measured, as well as differences in certain significant parts, allowing the classification of objects of the same type.

Keywords: *Digital curves, Polygonalization, Invariance, Turning pattern.*

1. Introducción

El análisis de imágenes digitales para la identificación de objetos dentro de las mismas muestra una creciente importancia, sobre todo por su aplicación a procesos automatizados de control. Ejemplos de estos procesos son control de calidad en la línea de fabricación, sistema de monitoreo de un predio para detección de intrusos, sistema automatizado de barreras en un estacionamiento privado, control de tránsito en determinada vía, etc. En las mencionadas aplicaciones, reconocer objetos y clasificarlos es una tarea central. Por ejemplo, en el control de calidad de una fábrica, el aporte que se espera del análisis de imágenes tomadas con una cámara en distintos puntos de la línea de fabricación consiste en determinar tempranamente fallas observables en su aspecto físico. Un producto que no respete la forma normalizada para la que fue diseñado debe descartarse.

En un estacionamiento privado que se gestiona automáticamente interesa por un lado tener información acerca de la cantidad de espacios que están disponibles en cada momento, distinguir si el vehículo entrante es uno de los vehículos autorizados, controlar el tipo de vehículo que ingresa y egresa, etc. Y todas esas tareas se pueden llevar a cabo a través del registro de cámaras dispuestas adecuadamente para tal fin.

El proceso de reconocimiento de objetos en imágenes digitales debe identificar los rasgos característicos del objeto que desea reconocer. Actualmente existe una gran cantidad de aplicaciones que usan visión artificial basada en topología digital para la eficiente identificación de objetos. Las estrategias usadas se basan en análisis de bordes, o análisis de toda la región que comprende el objeto [1].

El reconocimiento de formas puede hacerse por una comparación píxel a píxel entre dos conjuntos de puntos digitales, o utilizando ciertos atributos basados en características geométricas para medir la similitud entre dos formas. Claramente, la primera alternativa utiliza recursos (memoria y procesador) en forma intensiva, mientras que la segunda opción aprovecha características distintivas de las formas, para identificarlas con pocos datos, pero sin perder información importante sobre el objeto analizado.

Un atributo de gran utilidad para la identificación de un objeto es su borde o frontera.

Topológicamente, un objeto digital es un conjunto conexo de puntos en una grilla digital, y el borde del mismo está integrado por los puntos del objeto que limitan con puntos que no pertenecen al objeto. El borde siempre resulta ser una o varias curvas cerradas simples.

El conjunto de puntos que constituyen el contorno de una figura contiene mucha información acerca de la forma del mismo, por lo que su exploración permite extraer características geométricas de la silueta analizada: convexidad, dimensión, agujeros, puntos angulosos, bordes rectos, circularidad, etc.

Una de las características geométricas más útiles para el estudio de formas es la curvatura. La definición de curvatura para curvas digitales ha dado lugar a diversos estimadores. Considerando una curva digital como la digitalización de una curva suave continua, los métodos de estimación se basan en aproximar la derivada de la tangente, el módulo de la segunda derivada de una parametrización, o la inversa del radio del círculo osculante [2]. Por su parte, otros autores proponen una estimación basada en optimización global [3] o en arcos maximales [4].

Varios estimadores de la curvatura se basan en medir segmentos de rectas digitales [5]. La detección de rectas digitales juega un papel muy importante en el procesamiento de imágenes y geometría discreta. Existen varios algoritmos para determinar si un arco de curva digital es un segmento recto [5, 6, 7]. Los distintos métodos existentes se basan principalmente en preimágenes [8], combinatoria y fracciones continuas [9] o aritmética entera y desigualdades diofánticas [6].

Cuando se trabaja con imágenes provenientes del mundo real, capturadas con algún dispositivo con sensores, suele ocurrir que bordes que son rectos en el objeto real no son digitalizados como un segmento de recta digital. Esto puede atribuirse a la presencia de ruido, o a la sensibilidad a factores de iluminación, contraste y color del objeto y del fondo. En general, la detección de segmentos de líneas rectas digitales es muy sensible a la presencia de ruido en la imagen. En [10] se trata la detección de rectas con ruido mediante el concepto de segmento borroso. Este concepto se fundamenta en la noción de barra digital [11], que es una generalización de la definición de rectas digitales. En [12] se introduce el concepto de línea digital de ancho. Para detectar tales líneas es necesario conocer o calcular la cáscara convexa del conjunto de puntos, y medir su ancho vertical y horizontal (que puede no coincidir con el ancho de la cáscara convexa).

En aplicaciones en tiempo real (por ejemplo, reconocimiento de intrusos en una cámara de seguridad, o en visión robótica) el uso de los recursos de procesamiento y almacenamiento debe ser eficientes. Por eso, un enfoque que motiva muchas investigaciones actuales es reducir la cantidad de puntos del contorno que deben ser tenidos en cuenta, aproximando la curva por un polígono. Esto, en general, se lleva a cabo localizando arcos cuyos puntos están aproximadamente alineados.

Si bien la representación poligonal de curvas digitales es un procedimiento que nació casi al mismo tiempo que el procesamiento de imágenes digitales, aún hoy es una tarea que concentra mucha dedicación de investigadores [13, 14]. Se ha visto que es necesario relajar la condición de rectitud de los bordes, dando lugar al uso de segmento borroso o barra digital [15, 16]. Varios enfoques se orientan a la poligonalización óptima, definiendo diversas medidas del desempeño. Un enfoque apropiado para esto resulta ser la programación evolutiva [17, 18].

En la siguiente sección se presenta un algoritmo de aproximación de curvas digitales por medio de polígonos. El método se basa en detectar arcos de curvas que son aproximadamente rectos; más precisamente arcos que constituyen un segmento borroso. El conjunto de puntos en el arco tiene un ancho perpendicular menor que cierta tolerancia establecida por el implementador.

Asimismo, se define también una representación de formas invariante frente a movimientos rígidos (traslación, rotación, simetría), a fin de ser aplicada para el reconocimiento confiable y eficiente de objetos. Tal representación es construida a partir del polígono, y constituye una aproximación a la curvatura acumulada a lo largo de la curva y resulta un patrón que caracteriza la figura. La similitud entre figuras puede medirse entonces como la distancia entre sus patrones.

Se requiere que el patrón sea invariante frente a movimientos rígidos y también ante escalado uniforme. La invariancia frente a traslación, simetría y escalado se verifica evidentemente. Se han estudiado varias soluciones para lograr invariancia frente a rotaciones, pero de eficacia limitada [19].

Aquí se propone en primer lugar una alternativa para la invariancia frente a rotaciones. Además, se describe el uso de tramos parciales del patrón para focalizar la comparación en ciertas áreas de las formas. Para ello se identifican puntos distintivos, lo que permite segmentar el contorno de un objeto en partes significativas. Esta comparación por tramos permite determinar diferencias entre un conjunto de siluetas de objetos de la misma especie. En la última sección se incluyen resultados de la experimentación de los métodos sobre un conjunto de figuras diversas.

2. Metodología

2.1. Poligonalización de Curvas Digitales

Un arco digital es un conjunto de puntos $p_1, p_2, p_3, \dots, p_k$ de coordenadas $p_i = (n_i, m_i)$ tal que $d(p_i, p_j) = 1$ si y sólo si $|i - j| = 1$, y $d(p_i, p_j) > 1$ si $|i - j| \neq 1$ siendo d la distancia definida por

$$d(p_i, p_j) = \max \{ |n_i - n_j|, |m_i - m_j| \} \quad (1)$$

Esta definición implica que el segmento entre p_i y su sucesor en la curva, p_{i+1} es un segmento horizontal o vertical de longitud 1, o un segmento diagonal de longitud $\sqrt{2}$.

Una curva digital es un arco digital cerrado. Es decir, que verifica además que $d(p_1, p_k) = 1$.

La curva digital que delimita un objeto en una imagen es un conjunto de puntos discretos, y en general, presenta ruido, producto del proceso de captura de la imagen, o del proceso de segmentación que separa un objeto del fondo de la imagen. Para suavizar la curva y para reducir la cantidad de puntos necesarios para describirla, se realiza una poligonalización, dividiendo la curva en tramos casi rectos. Este procedimiento simplifica la curva, haciendo que la cantidad de datos a manejar se reduzca notablemente, sin perder información relevante. Además, es capaz de ignorar los efectos del ruido en la imagen (siempre que la profundidad del ruido sea menor que la tolerancia de aproximación).

Para ello, deben detectarse arcos en la curva dada que se puedan aproximar por segmentos rectos. La intención no es reconocer exactamente los conjuntos de puntos provenientes de la digitalización de rectas (tarea altamente sensible al ruido), sino identificar conjuntos de puntos que están aproximadamente alineados. Esta identificación se hace teniendo en cuenta que una recta continua es un conjunto de ancho 0, y por lo tanto, puntos aproximadamente alineados forman un conjunto de ancho relativamente chico. Por lo tanto, especificada una tolerancia, cuando un arco de la curva tiene ancho menor que esa tolerancia, se considera que forma un segmento recto.

El algoritmo de poligonalización construye sucesivamente los lados del polígono. Para cada punto de la curva, se analiza si este puede ser admitido en el segmento que se está construyendo, o debe pertenecer al siguiente segmento. Siendo $\{p_s, p_{s+1}, p_{s+2}, \dots, p_{s+r}\}$ los puntos ya admitidos en el segmento y p_{s+r+1} el siguiente punto en la curva, el procedimiento evalúa si el ancho del conjunto $\{p_s, p_{s+1}, p_{s+2}, \dots, p_{s+r}, p_{s+r+1}\}$ es menor a la

tolerancia, y en tal caso, admite el punto en el lado. La salida del algoritmo es el conjunto $\{v_1, v_2, \dots, v_n\}$ de puntos iniciales de los arcos casi rectos detectados, que constituyen los vértices del polígono aproximante.

Claramente, el método propuesto depende del punto inicial de la curva de borde, pero esa dependencia no hace perder características globales fundamentales de la geometría del objeto. También depende de la tolerancia especificada. Imponiendo tolerancias grandes, se generan polígonos con pocos vértices y lados que abarcan muchos puntos de la curva. Así, podrían perderse detalles de menor escala, aunque también hace al método menos sensible al ruido. Con tolerancias más pequeñas se obtienen polígonos de más vértices. En el caso límite de tolerancia 0, el proceso de poligonalización elimina los puntos intermedios de segmentos de rectas digitales de pendiente múltiplo de $\pi/4$.

La Fig. 1 muestra aproximaciones de una silueta, usando tolerancias 0, 1 y 3. La curva tiene 1065 puntos, y los polígonos tienen 315, 71 y 23 puntos, respectivamente.

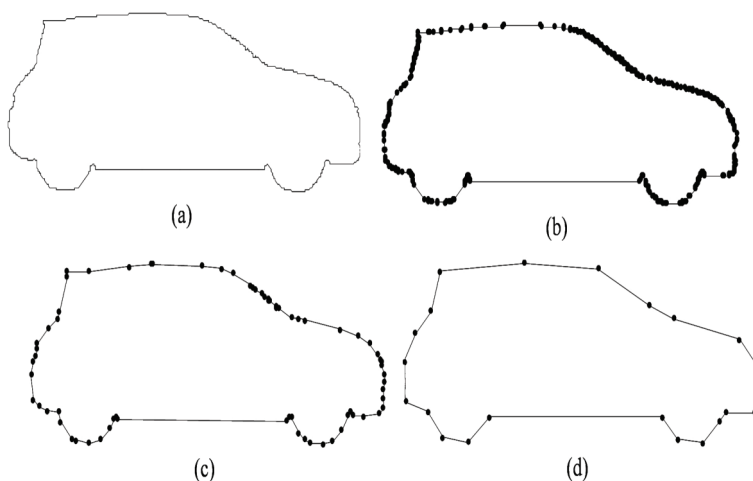


Fig. 1. Una silueta (a) y las poligonalizaciones con tolerancias: b) $\epsilon = 1$, d) $\epsilon = 3$.

Hemos explorado dos enfoques para medir el ancho de conjuntos. El primero se fundamenta en la definición de barra digital como el conjunto de puntos de coordenadas enteras entre dos rectas paralelas. El segundo enfoque se basa en el ancho de la cáscara convexa de un conjunto.

2.1.1 Barras Digitales.

En geometría digital se define barra digital [6] como el conjunto

$$B = \{ (x, y) \in Z^2: c \leq ax - by < c + d \} \quad (2)$$

con a, b, c, d enteros, $d > 0$ y $mcd(a, b) = 1$. La pendiente de la barra es a/b , el ancho es d . El conjunto está limitado por las rectas

$$y = \frac{a}{b}x - \frac{c}{b} \quad (3)$$

y

$$y = \frac{a}{b}x - \frac{c+d-1}{b} \quad (4)$$

Si $b \neq 0$ La distancia vertical entre estas rectas es $w = \frac{d-1}{|b|}$, y la distancia perpendicular es $\delta = \frac{aw}{\sqrt{a^2+b^2}}$.

En el método propuesto en este trabajo se consideran barras digitales de ancho $\delta = 2\varepsilon$, donde ε es la tolerancia especificada. El procedimiento de admisión de un punto en un lado mide la distancia perpendicular de cada uno de los puntos $p_{s+1}, p_{s+2}, \dots, p_{s+r}$ a la recta que pasa por p_s y p_{s+r+1} . Si todas estas distancias son menores que la tolerancia ε , entonces todos los puntos entre p_s y p_{s+r+1} están dentro de la barra digital de eje en el segmento con extremos p_s y p_{s+r+1} , de ancho 2ε . En este caso, el punto p_{s+r+1} , queda admitido en el lado. En caso contrario, se comienza un nuevo segmento, con vértice inicial en p_{s+r} (que es el vértice final del segmento anterior).

La salida del algoritmo es el conjunto $\{v_1, v_2, \dots, v_n\}$ de extremos iniciales de los lados, que son los vértices del polígono que aproxima la curva dada.

En la Fig. 2 se observa un conjunto de puntos que forman una curva. Los puntos p_1, p_2, \dots, p_{10} están en el mismo lado del polígono, ya que la distancia de cada punto al segmento $p_1 - p_{10}$ es mayor a ε , siendo ε la tolerancia preestablecida. El punto p_{11} no se admite en el lado anterior, ya que la distancia de algunos puntos intermedios (p_7, p_8, p_9 y p_{10}) al segmento $p_1 - p_{11}$ es mayor a ε .

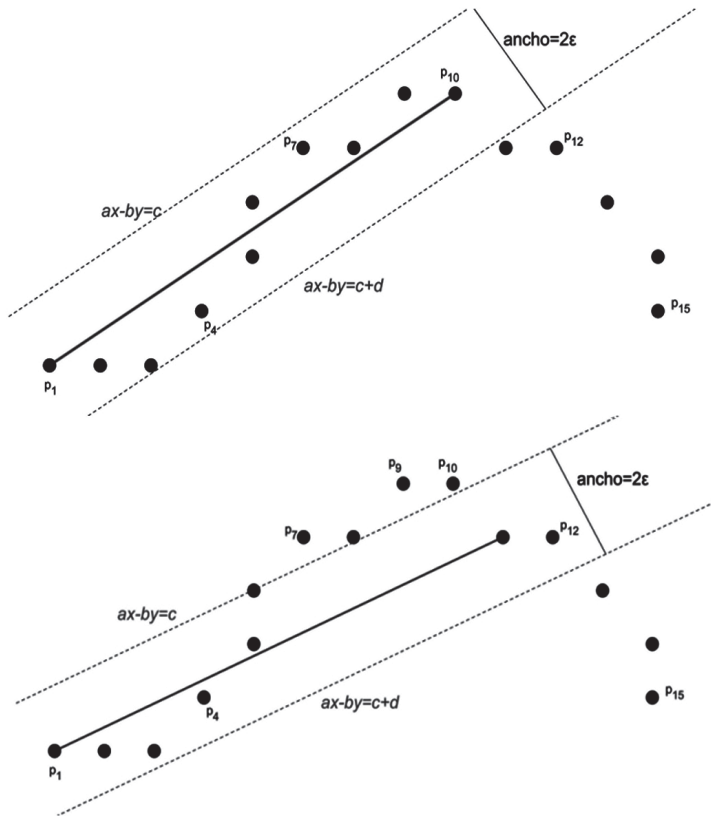


Fig. 2. Los puntos $p_1 - p_{10}$ están en una barra digital de ancho 2ϵ , pero los puntos $p_7 - p_{11}$ no lo están, ya que p_7, p_8, p_9 y p_{10} no caen dentro de la barra correspondiente.

Una ventaja del método es su sencillez a la hora de programarlo. La admisión de punto en lado se basa en el cálculo de distancias de puntos a rectas, cálculo sumamente sencillo.

2.1.2 Cáscara Convexa.

Se define el ancho de un conjunto como el ancho de su cáscara convexa. La cáscara convexa de un conjunto de puntos es el menor conjunto convexo que lo contiene. Y su ancho es la menor distancia entre un par de rectas de soporte paralelas. Existen diversos algoritmos muy conocidos para obtener la cáscara convexa de un conjunto de puntos. En este trabajo se implementó un algoritmo incremental [20].

En la Fig. 3 se muestra el conjunto de puntos p_1, p_2, \dots, p_{10} y su cáscara convexa. El ancho de ésta es 0.7. La cáscara convexa de los puntos p_1, p_2, \dots, p_{11} tiene ancho 1.1, de modo que si la tolerancia es, por ejemplo, $\varepsilon = 1$, $p_1 - p_{10}$ están en un lado, y p_{11} no se admite en el mismo.

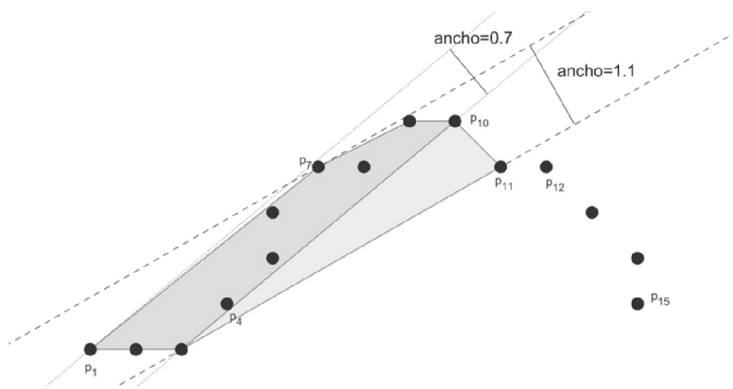


Fig. 3. La cáscara convexa de los puntos $p_1 - p_{10}$ y de los puntos $p_1 - p_{11}$.

2.2 Patrón Normalizado de Curvatura

Para comparar polígonos y medir la similitud entre ellos, se define el patrón normalizado de curvatura. Para cada i , se define λ_i como la longitud normalizada de los lados del polígono desde el vértice v_i al vértice v_{i+j} . Es decir;

$$\lambda_i = \sum_{j=1}^i \frac{\text{long}(v_j v_{j+1})}{L} \quad (5)$$

Y $\lambda_i = 0$; siendo L el perímetro del polígono.

Por otro lado, se define una estimación de la curvatura acumulada desde el punto inicial al vértice v_i como

$$k_i = \sum_{j=1}^i \Delta\theta_j \quad (6)$$

donde $\Delta\theta_j$ es el ángulo orientado en v_j ; es decir, es el ángulo determinado por $v_{j-1}v_jv_{j+1}$.

La orientación del ángulo en v_j se considera positiva si al recorrer los vértices en orden $v_{j-1} - v_j - v_{j+1}$ se produce un giro en sentido antihorario; y se considera negativa si el giro es en sentido horario.

Claramente, para el último vértice del polígono v_n la longitud normalizada es $\lambda_n=1$ y la curvatura acumulada es $kn = -2\pi$ (recorriendo el borde de la figura en sentido horario).

La curva lineal por tramos en el plano $\lambda - k$ determinada por los puntos $\beta_n = (\lambda_i - k_j)$ con inicio en $\beta_0 = (0 - 0)$ es el *patrón normalizado de curvatura*. Como esta curva lineal por tramos no tiene segmentos verticales, también se puede pensar que define a κ como función de λ obteniéndose la función $\kappa(\lambda)$ para todo λ en el intervalo $[0, 1]$.

La Fig. 4 muestra una silueta poligonalizada de un automóvil, y el patrón de curvatura resultante. En líneas de puntos se marcan puntos correspondientes en ambos esquemas.

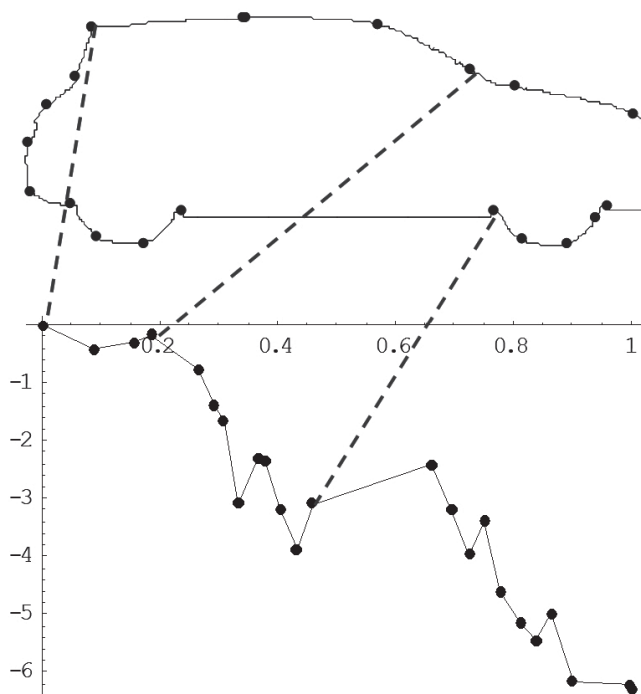


Fig. 4: Una silueta poligonalizada y su patrón de curvatura.

El patrón de curvatura encierra toda la información de la silueta analizada. Las zonas crecientes del patrón representan las partes cóncavas de la curva, y las decrecientes las zonas convexas. De esta forma, los mínimos y máximos relativos del patrón representan puntos de inflexión. Estos puntos son tomados como puntos de referencia para segmentar la forma en tramos cóncavos y tramos convexos, e identificar en el objeto analizado los tramos característicos. Esto es de gran utilidad para llevar a cabo clasificación de objetos de cierto tipo, que son en gran parte similares, y las disimilitudes se centran en una zona determinada del borde.

Por otra parte, los puntos de referencia y la segmentación en tramos característicos permiten determinar la posición del objeto; o más específicamente, comparar dos objetos independientemente de su posición en la imagen.

El patrón definido depende del orden en que se tomen los puntos del polígono, en particular, de cuál es el primer punto, que es el primer punto del objeto, detectado por el algoritmo de localización de bordes. En consecuencia, el primer punto depende de la posición en la que aparece el objeto en la imagen.

Obviamente, al cambiar el punto de inicio del polígono, cambiará el patrón, pero no la información que contiene.

Si se tiene un patrón obtenido a partir del vértice v_l , y se quiere obtener el patrón del mismo polígono, pero a partir del vértice v_m , todos los puntos del patrón se desplazan de acuerdo a las relaciones:

Para: $k = m, m+1, \dots, n$:

$$\beta_{k-m+1} = (\lambda_k - \lambda_{m-1}, k - k_{m-1}) \quad (7)$$

Para $k = 1, 2, \dots, m-1$:

$$\beta_{k-m+n+1} = (\lambda_k - \lambda_{m-1}, +1 k - k_{m-1} - 2\pi) \quad (8)$$

Con el punto de inicio $\beta_0 = (0, 0)$.

Para comparar dos siluetas, independientemente de la posición, se busca establecer una correspondencia entre los puntos de referencia de las dos curvas, es decir, entre los puntos de inflexión de las siluetas analizadas.

Sean $x = \{x_1, x_2, \dots, x_n\}$ e $y = \{y_1, y_2, \dots, y_m\}$ los conjuntos de puntos de referencia de dos objetos. Estos son los valores de que son mínimos

relativos en el patrón (se toman sólo los mínimos del patrón, ya que se ha verificado en la experimentación que son suficientes para identificar los tramos característicos). Se establece una correspondencia entre los puntos de esos conjuntos, tomando la asignación que logre mejor correlación entre patrones, de acuerdo con una función de desviación.

Se define la desviación de la correspondencia $x_j \sim y_k$ como

$$f(x_j, y_k) = \sum_{i=1}^n wx_{ijk} + \sum_{i=1}^m wy_{ijk} \tag{9}$$

donde $wx_{ijk} = \min_{h=1, \dots, m} |\tilde{x}_{ij} - \tilde{y}_{hk}|$ y $wy_{ijk} = \min_{h=1, \dots, m} |\tilde{x}_{hj} - \tilde{y}_{ik}|$ siendo $\tilde{x}_{ij} = x_i - x_j$ $\tilde{y}_{hk} = y_h - y_k$; es decir, las coordenadas centradas en (x_j, y_k) .

Finalmente, se selecciona la correspondencia con menor desviación; y esos puntos se toman como inicio de los patrones correspondientes.

En la Fig. 5 se observan dos patrones correspondientes a dos formas a comparar. Las formas a analizar son los perfiles de dos vehículos similares. Todos los puntos de referencia en cada patrón se muestran en gris, mientras que los puntos que ofrecen menor desviación aparecen en negro.

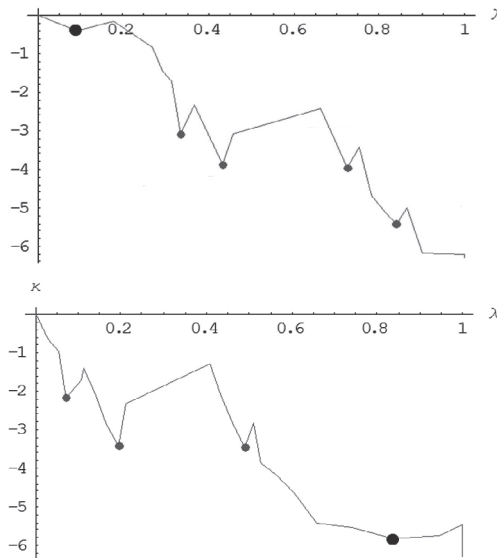


Fig. 5. Dos patrones de objetos similares y los puntos de referencia asociados.

De esta forma, se puede tomar el inicio de los patrones en el punto obtenido al hacer la correlación entre los puntos de referencia. Los patrones de la Fig. 5, con sus respectivos nuevos inicios se muestran en la Fig. 6. Se puede observar la similitud entre los patrones con los respectivos nuevos puntos iniciales, hecho que es esperable, dado que corresponden a objetos similares.

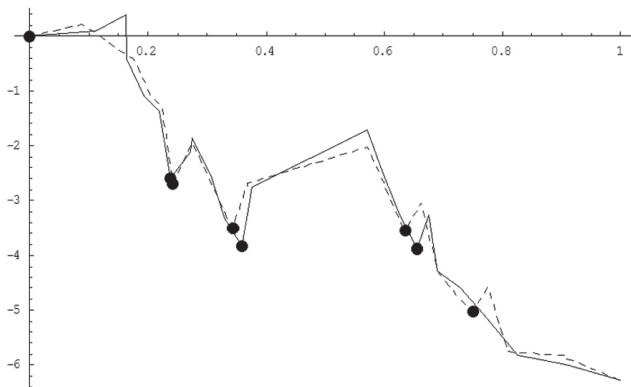


Fig. 6. Los dos patrones de la Fig. 5, tomando como punto de inicio los puntos de referencia asociados.

2.3. Identificación de Objetos Usando el Patrón de Curvatura

Definimos la distancia entre dos patrones k_1 y k_2 como

$$D(k_1, k_2) = \int_0^1 (k_1(\lambda) - k_2(\lambda))^2 d\lambda \quad (10)$$

El patrón de curvatura es evidentemente invariante frente a traslaciones y escalados uniformes. Entonces, dos patrones obtenidos a partir de distintas instancias de la misma silueta, donde una se obtiene por traslación y/o escalado de la otra en la imagen digital, tienen distancia cero entre sí. Pero esto no ocurre en general, en figuras relacionadas por otra transformación.

La rotación es una transformación rígida en un espacio euclídeo, pero no en el plano digital. Al rotar una curva digital se exige a los puntos caer en puntos grilla, introduciendo así error de discretización. Si se comparan instancias de una figura, afectadas por una rotación, el error de

discretización afectará el patrón de curvatura, haciendo que la distancia así definida no sea cero.

Esta discrepancia entre distintas instancias del mismo objeto, ubicado en distintas posiciones, se hace más notable si se procede con imágenes reales, tomadas con scanners o cámaras, ya que el proceso de captura o las condiciones de iluminación podrían introducir ruido.

La clave para que el método de reconocimiento sea efectivo es que estas diferencias debido a la discretización son mínimas frente a diferencias debidas a cambios perceptibles en las formas.

Sea $k(\lambda)$ el patrón de curvatura de una forma S que se quiere examinar. La identificación se realiza por comparación con patrones conocidos. Sean $\alpha_j(\lambda)$ patrones de formas candidatas a identificarse con la forma analizada. Entonces, se hace corresponder la forma S con aquella cuyo patrón α_j diste de k en menos de un valor umbral, es decir:

$$D(k, \alpha_j) \leq \mu \quad (11)$$

La métrica definida también puede aplicarse a tramos parciales del patrón. La idea es analizar el tramo del borde comprendido entre dos puntos de referencia. Para hacer la comparación, se normaliza el tramo y se mide la distancia entre los patrones correspondientes.

Sean $\beta_j = (\lambda_j, k_j)$ y $k = (\lambda_k, k_k)$ los puntos que delimitan la zona a estudiar en el borde de un objeto. Los puntos del patrón comprendidos entre β_j y β_k se normalizan de acuerdo a

$$\tilde{\beta}_i = \left(\frac{\lambda_i - \lambda_j}{\lambda_k - \lambda_j}, k_i - k_j \right) \quad (12)$$

Luego, se mide la distancia entre el tramo considerado en las figuras que se están analizando.

Esto es aplicable al análisis y clasificación de objetos similares, que concentran las diferencias en una zona del borde, como es el caso de los distintos tipos de vehículos (bicuerpo, tricuerpo, furgón, etc).

3. Resultados

Los métodos y algoritmos descritos en las secciones anteriores fueron pensados para ser parte de un sistema de reconocimiento y clasificación de alcance general.

A fin de experimentar con los mismos, se aplicó el método propuesto sobre un conjunto de curvas de borde de 19 vehículos de distintos tipos, y un conjunto de otras curvas correspondientes a objetos digitales de diversas formas (formas abstractas, muebles, frutas, etc).

Es claro que el método de poligonalización depende del punto inicial de la curva digital. Para medir la dimensión de la dependencia entre la cantidad de vértices y el punto inicial, se obtuvieron poligonalizaciones de varias curvas, a partir de distintos puntos iniciales. Los resultados indican que el número de vértices varía muy poco cambiando el punto inicial, y a mayor tolerancia, menor variación en la cantidad de vértices. En general, la cantidad de lados del polígono no varía en más de 3 vértices, y con tolerancias mayores a $\epsilon = 3$, la variabilidad es de un vértice. Por lo tanto, a los fines de la aplicación que motiva este trabajo, se puede ignorar esta dependencia.

Otro ítem que se estudió fue la reducción de puntos de una curva al hacer una aproximación poligonal. La Tabla 1 muestra los porcentajes de puntos que permanecen luego de la aproximación, con distintas tolerancias para distintas curvas. La tolerancia $\epsilon = 0$ produce un polígono que coincide con la curva original, es decir, elimina puntos de arcos de curva que están perfectamente alineados. Entonces, el porcentaje de reducción en este caso representa una medida de rectitud en los bordes de la figura. Por ejemplo, la curva 5 de la tabla es el borde de una letra *c* (casi no tiene bordes rectos) y es el caso en el que se produce la menor reducción de puntos. En el otro extremo, la curva 6 es la silueta de una silla (gran parte de los bordes son rectos), donde se produce una gran reducción de puntos con tolerancia mínima.

	Puntos	Tolerancia				
		0	1	2	3	5
Curva 1	1177	37.6	5.7	2.9	2.2	1.8
Curva 2	1427	42.0	5.2	3.4	2.7	1.9
Curva 3	452	24.3	4.8	3.5	3.3	2.8
Curva 4	1482	15.6	3.6	1.9	1.7	1.1
Curva 5	1700	52.5	11.8	3.11	1.9	1.3
Curva 6	549	12.9	3.6	2.9	2.9	2.5

Tabla 1. Puntos luego de la aproximación poligonal con distintas tolerancias (en porcentajes).

De los valores de reducción obtenidos es evidente que uno de los principales resultados de la poligonalización, es poder representar una curva con mucha menos información. Se puede observar de los

resultados que se produce una marcada reducción de puntos usando tolerancia $\varepsilon = 1$ o $\varepsilon=2$, y que la reducción no mejora demasiado usando tolerancias mayores. Por lo tanto, en las experimentaciones siguientes se usó tolerancia $\varepsilon=2$.

La aplicación del método propuesto al conjunto de curvas digitales de prueba arrojó los siguientes resultados.

Se obtuvo que, en general, la distancia entre las siluetas de un par de vehículos no supera el valor 1.2, siendo el promedio de estas distancias 0.54. Por ejemplo, los vehículos cuyos patrones se muestran en la Fig. 6 distan 0.216.

Los pocos casos en que la distancia supera el valor de 1.2 son ocasiones en que el método de identificación de puntos de referencia establece la correspondencia entre puntos que no están en la misma posición en el borde del vehículo. El motivo principal de esto es el hecho que ciertos objetos traen una cantidad considerable de ruido, lo que hace que aparezcan en los correspondientes patrones puntos de referencia ficticios, que no representan puntos de inflexión en la forma, alterando la identificación.

La distancia entre el perfil de un vehículo y cualquier otra forma, en nuestra experimentación, es siempre mayor que 2. La Fig. 7 muestra el patrón de un vehículo, junto con el patrón correspondiente a una forma abstracta, semejante a una flor. Puede observarse la disparidad de los patrones en toda su traza.

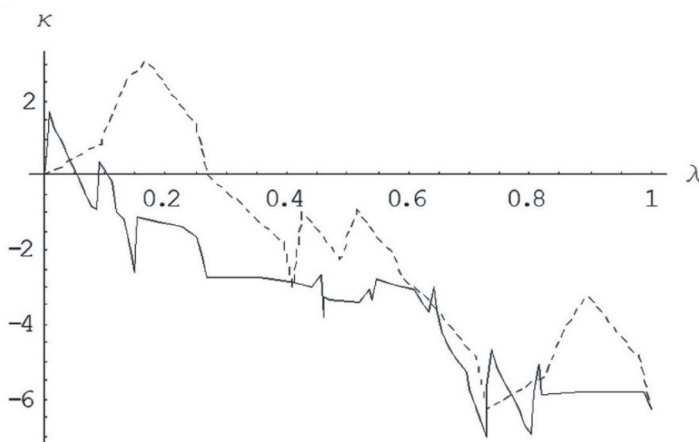


Fig. 7. Patrón de un vehículo (línea llena) y de otro objeto (línea punteada).

Haciendo un análisis por sectores del borde del objeto digital, encontramos que la diferencia entre pares de patrones de vehículos, en ciertos casos, se concentra en una determinada zona. Por ejemplo, en la Fig. 8 se comparan patrones de un vehículo tipo furgón y un vehículo tricuerpo. La distancia global es 0.289. Se puede observar que la mayor diferencia se encuentra para valores de λ entre 0.2 y 0.3. Numéricamente, en ese intervalo se acumula el 30% de la distancia total. Esa zona de concentración de la diferencia corresponde a la parte trasera del vehículo, donde uno de los vehículos presenta baúl y el otro no.

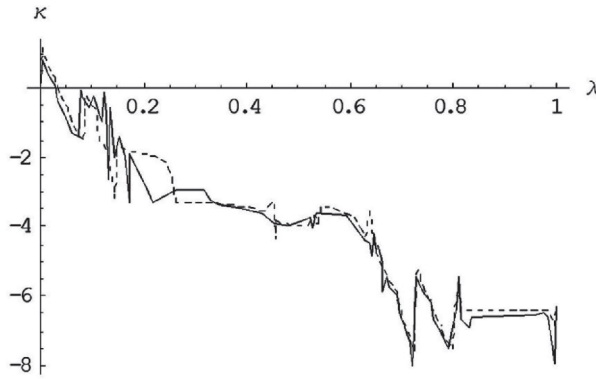


Fig. 8. Patrones de dos vehículos: furgoneta (línea punteada) y tricuerpo (línea llena).

Otro ejemplo puede verse en la Fig. 9. Allí se muestran en conjunto los patrones de un vehículo bicuerpo y un vehículo tricuerpo. Se destaca la marcada diferencia en el tramo inicial de los patrones, nuevamente correspondientes a la parte trasera de los vehículos.

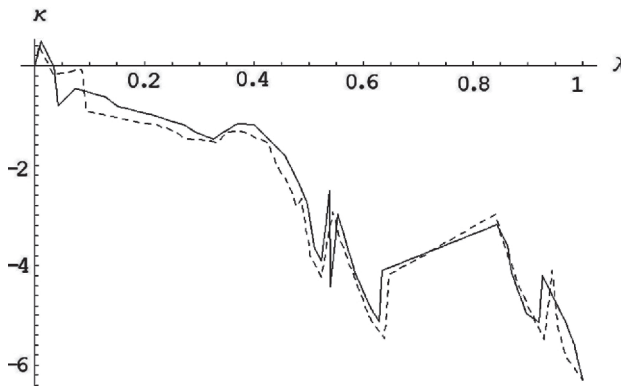


Fig. 9. Patrones de dos vehículos: bicuerpo (línea punteada) y tricuerpo (línea llena).

Por otro lado, en la Fig. 10 se pueden observar los patrones de dos vehículos tipo furgoneta. La distancia entre ellos es 0.204, y esta distancia se distribuye regularmente a lo largo de todo el patrón.

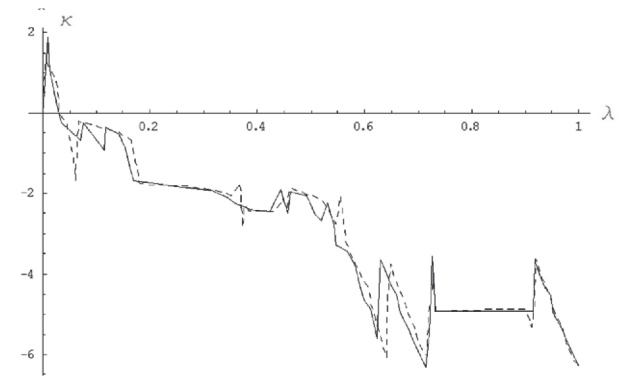


Fig. 10. Patrones de dos vehículos tipo furgoneta.

4. Conclusiones

El patrón de curvatura definido es una herramienta útil en la comparación de formas en imágenes digitales. Al ser una aproximación de la curvatura del borde del objeto, permite determinar puntos de inflexión y usarlos como referencia para dividir la curva en partes significativas, a fin de lograr la invariancia del patrón frente a rotaciones, es decir, que el patrón sea independiente de la posición del objeto en la imagen digital.

Aunque en la mayoría de los casos de experimentación la localización de los puntos de referencia correspondientes fue exitosa, hubo casos en que esto falló, debido a la aparición de puntos de referencia ficticios, originados por ruido de alta profundidad. El proceso de poligonalización tiende a ignorar el efecto del ruido dependiendo de la tolerancia usada. A mayor tolerancia, menor efecto del ruido, pero la aproximación se aleja de la forma real. Existe entonces un compromiso entre la tolerancia usada y la incidencia del ruido en la eficacia del método.

El algoritmo propuesto es simple, de baja complejidad, y la tolerancia está determinada por el implementador, por lo que este procedimiento permite lograr la identificación rápida de objetos dentro de las imágenes, y la clasificación dentro de un conjunto de objetos de la misma especie. La posibilidad de descomponer el patrón en tramos significativos permite medir distancias parciales en determinadas zonas del borde de objetos, focalizando la identificación de diferencias y similitudes.

Referencias

- [1] Zhang D., Lu G. *Review of shape representation and description techniques*, Patt. Recogn. 37 (1), 1-19 (2004).
- [2] Hermann S., Klette R. *A comparative study on 2D curvature estimators*. Int. Conference on Computing: Theory and Applications (ICCTA'07), 584-589 (2007).
- [3] Kerautret B., Lachaud J.-O. *Curvature Estimation along Noisy Digital Contours by Approximate Global Optimization*. Patt. Recogn. 42(10), 2265–2278 (2008).
- [4] Roussillon T., Lachaud J.-O. *Accurate Curvature Estimation Along Digital Contours With Maximal Digital Circular Arcs*. In: Aggarwal J., Barneva R., Brimkov V. Koroutchev K., Korutcheva E. (eds.) *Combinatorial Image Analysis*, LNCS 6636, 43-55, Springer, Heidelberg (2011).
- [5] Vieilleville , Lachaud J.-O. *Revisiting digital straight segment recognition*. In: Kuba A., Nyúl L., Palágyi K. (eds.) *Discrete Geometry for Computer Imagery*, LNCS 4245, 355-366 (2006).
- [6] Debled-Renesson, I., Reveillès, J.P. *A linear algorithm for segmentation of digital curves*. Int. J. Pattern Recogn. Artif. Intell. 9, 635–662 (1995).
- [7] Rosenfeld A. *Digital straight lines segments*, IEEE Trans. Comput. 23, 1264–1369 (1974).
- [8] Dorst L., Smeulders A.W.M. *Discrete representation of straight lines*. IEEE Trans. Pattern Anal. Mach. Intell. 6, 450–463 (1984).
- [9] Berstel J., De Luca A. *Sturmian words, lyndon words and trees*. Theoret. Comput. Sci. 178(1-2), 171–203 (1997).
- [10] Nguyen T.P., Debled-Renesson I. *Curvature Estimation in Noisy Curves*. In: Kropatsch W., Kampel M., Hanbury A. (eds.) *Computer Analysis of Images and Patterns*, LNCS 4673, 474-481, Springer, Heidelberg (2007).
- [11] Kettle R., Rosenfeld A. *Digital Straightness – a review*. Discrete Appl. Math. 139, 197-230 (2004).
- [12] Buzer L. *An Elementary Algorithm for Digital Line Recognition in the General Case*. In: Andres E., Damiand G., Lienhardt P. (eds.)

- Discrete Geometry for Computer Imagery, LNCS 3429,299-310 (2005).
- [13] Carmona-Poyato A., Madrid-Cuevas F.J., Medina Carnicer R., Muñoz-Salinas R. *Polygonal approximation of digital planar curves thought break point suppression*. Pattern Recognition 43(1), 14-25 (2010).
 - [14] Masood, A. *Dominant point detection by reverse polygonization of digital curves*. Image and Vision Computing 26(5), 702-715 (2008).
 - [15] Bhowmick, P., Bhattacharya, B. B. *Fast polygonal approximation of digital curves using relaxed straightness properties*. IEEE Trans. on Pattern Anal. Mach. Intell. 29(9), 1590-1602 (2007).
 - [16] Debled-Rennesson, I., Rémy, J. L., Rouyer-Degli, J. *Linear segmentation of discrete curves into blurred segments*. Discrete Applied Mathematics, 151(1-3), 122-137 (2005).
 - [17] Wang, B., Shu, H., Shi, C., Luo, L. *A novel stochastic search method for polygonal approximation problem*. Neurocomputing 71(16-18), 3216-3223 (2008).
 - [18] Alvarado-Velazco P., Ayala-Ramirez V. *Polygonal approximation of digital curves using genetic algorithm*, IEEE International Conference on Industrial Technology (ICIT) 254-259, (2012).
 - [19] Kamlofsky, J.A., Bergamini, M. L. *Patrón de Evolución Discreta de Curvatura y Concavidad para Reconocimiento de Formas*. CONAIISI, (2013).
 - [20] Kallay M. *The complexity of incremental convex hull algorithms in R^d* , Inform. Process. Lett. 19(4), 197 (1984).