

Rápida Obtención del Apareo de Imágenes para Visión Robótica

» » **Jorge Kamlofsky**

CAETI – Universidad Abierta Interamericana – Facultad de Tecnología Informática

Resumen

La visión robótica le otorga a los robots la capacidad de interpretar la escena emulando la visión humana. Esto puede lograrse con un par de cámaras que obtienen imágenes del entorno desde dos puntos de vista. Se obtiene un modelo tridimensional, en unidades de longitud mediante una técnica estereoscopia. Para su cálculo, se requiere conocer las características físicas del sistema de adquisición de las imágenes, las cuales generalmente se obtienen por única vez cuando se calibra el sistema, al inicio. Luego, para la reconstrucción tridimensional es fundamental tener las coordenadas de puntos en ambas imágenes que se corresponden con cada punto de la escena. Este proceso es llamado reconocimiento de puntos homólogos, puesta en correspondencia o apareo y posee muy alta complejidad de cálculo. Este trabajo se enfoca sobre esta tarea. Es necesario reconocer automáticamente objetos que están presente en ambas imágenes para lograr la asociación de puntos de la escena. Para ello, el análisis de bordes de objetos en imágenes es un enfoque común. Un problema que surge inmediatamente es que las formas se encuentran en cualquier posición, tamaño y orientación dentro de las imágenes. Se presenta un patrón descriptor de las curvas de borde que resulta invariante frente a rotaciones, traslaciones y escalado, consistente en una curva discreta. Lograr el reconocimiento de un objeto en una imagen consiste entonces, en medir la distancia entre el patrón descriptor del objeto a buscar y los patrones de cada objeto de la imagen y comparar su diferencia con un umbral de similitud predefinido. Dicho patrón resulta dependiente de la determinación de un punto de inicio. Sin embargo, tras la obtención de la orientación de la forma y dado que en estereoscopia, normalmente, ambas imágenes son similares, lo son también los objetos allí contenidos. Entonces, la identificación del punto de inicio se logra fácilmente. Con ello, realizar el apareo de varios puntos de las curvas de borde de un objeto puede lograrse con menor dificultad, lo cual permite lograr una representación tridimensional aproximada que puede ser compatible con requerimientos de tiempo real. Se incluyen datos experimentales.

PALABRAS CLAVE: VISIÓN ROBÓTICA, PUNTOS HOMÓLOGOS, VISIÓN 3D, ESTEREOSCOPÍA, RECONSTRUCCIÓN TRIDIMENSIONAL.

Rapid Obtaining of Image Matching for Robotic Vision

Abstract

Robotic vision gives robots the ability to interpret the scene by emulating human vision. This can be achieved with a pair of cameras that obtain images of the environment from two points of view. A three-dimensional model is obtained, in units of length, by means of a stereoscopic technique. For its calculation, it is necessary to know the physical characteristics of the image acquisition system, which are generally obtained only once when the system is calibrated, at the beginning. Then, for the three-dimensional reconstruction it is essential to have the coordinates of points in both images that correspond to each point in the scene. This process is called recognition of homologous points, correspondence or aspect and has a very high computational complexity. This work focuses on this task. It is necessary to automatically recognize that they are present in both images to achieve the association of points in the scene. To do this, edge analysis of objects in images is a common approach. One problem that immediately arises is that the shapes are in any position, size, and orientation within the images. A descriptor pattern of edge curves is presented that is invariant against rotations, translations and scaling, consisting of a discrete curve. Achieving the recognition of an object in an image then consists in measuring the distance between the descriptor pattern of the object to be searched for and the patterns of each object in the image and comparing their difference with a predefined threshold of similarity. This pattern is dependent on the determination of a starting point. However, after obtaining the orientation of the shape and given that in stereoscopy, both images are normally similar, so are the objects contained therein. Then the identification of the starting point is easily achieved. With this, making the multi-point appearance of an object's edge curves can be achieved with less difficulty, allowing for an approximate three-dimensional representation that can be compatible with real-time requirements. Experimental data is included.

KEYWORDS: ROBOTIC VISION, HOMOLOGOUS POINTS, 3D VISION, STEREOSCOPY, THREE-DIMENSIONAL RECONSTRUCTION.

1 Introducción

Este artículo es un resumen del trabajo originalmente presentado en la XIX edición del Congreso Internacional de Innovación en Tecnología Informática (Kamlofsky, 2021). Dicho trabajo obtuvo el 1º puesto del Certamen de Trabajos Estudiantiles, en la categoría: “Trabajos de Post-grado”.

La visión robótica busca emular a la visión humana para darle a los robots la capacidad de comprender el entorno en el que se desenvuelven y así, poder actuar en consecuencia. Para ello, se equipa a los robots con cámaras que actuando a modo de ojos, obtienen imágenes cuyo procesamiento permite obtener datos acerca de la escena. El abanico de aplicaciones de visión robótica es muy amplio. Gran parte de ellas, las más simples, tienen condiciones acotadas y poco cambiantes. En otras, el reconocimiento debe lograrse en ambientes muy cambiantes y en muy poco tiempo. A veces, incluso, se requiere que esto se logre en tiempo real.

1.1 Trabajos relacionados

El sistema de visión robótica se caracteriza según el método que usa para la identificación de objetos en las imágenes. Las estrategias conocidas para reconocer objetos en imágenes se basan en el análisis de los bordes o el estudio integral de la imagen o de una ventana parcial, donde es probable que el objeto esté (Zhang y Lu, 2004). Las estrategias basadas en el análisis de bordes utilizan técnicas para detectar líneas y bordes mediante técnicas de umbralizado (Davis, 1975; Rosenfeld y Kak, 2014), o mediante el uso de diferentes filtros (Gonzalez y Woods, 1993) para luego extraer las características de la forma utilizando Topología Digital (Eckhardt y Latecki, 1995; Rosenfeld, 1979) o reconociendo patrones obtenidos desde el borde. Las estrategias basadas en el análisis del área completa de la imagen obtienen características del objeto contenido en la imagen mediante técnicas de entrenamiento y reconocimiento de patrones utilizando algoritmos de inteligencia artificial (iniciales: IA) como ser: redes neuronales donde el algoritmo aprende a partir de un conjunto de imágenes, algoritmos genéticos (Rowley et al., 1998) o bien, mediante el uso de la transformada wavelets (Sarria, 2007).

La digitalización de las imágenes, la resolución de la cámara, la falta o el exceso de brillo, la ofuscación, la pérdida de claridad o el ruido en la imagen, la variedad de objetos a identificar, son aspectos a tener en cuenta, ya que en muchos casos, esto requiere un tratamiento adicional (Weaver et al., 1991; Rudin et al., 1992; Nguyen y Debled-Rennesson, 2007), lo que compromete el cumplimiento de requerimientos de tiempo real. Por el contrario, en condiciones controladas, sencillas y acotadas, el tratamiento es más sencillo, lo que facilita la identificación de objetos. Entonces, en aplicaciones donde el reconocimiento de objetos debe hacerse en tiempo real, es muy conveniente que las condiciones se simplifiquen de modo de lograr un equilibrio que permita un procesamiento correcto en el tiempo requerido.

Al basarse en imágenes digitales, la visión robótica requiere convertir la información bidimensional de la escena en un modelo tridimensional para reconocer objetos, de modo de ejecutar adecuadamente las tareas asignadas. La estereoscopia es una técnica que, a partir de dos (o más) imágenes de una escena logra una reconstrucción tridimensional de la misma (Moravec, 1996). El cálculo se realiza luego de haber finalizado el proceso de reconocimiento de puntos homólogos o puesta en correspondencia o apareo en ambas imágenes. Esto es: la obtención de los pares de coordenadas de puntos en ambas imágenes que se corresponden al mismo punto del objeto. Es conveniente, que previo a ello los objetos en cada imagen hayan sido reconocidos. Sin embargo, esta tarea requiere probar la coincidencia de millones de píxeles en millones de combinaciones posibles (Zelasco et al., 2000), lo que compromete los requisitos de reconocimiento en tiempo real.

El reconocimiento de los objetos en imágenes es un aspecto central en este trabajo. Las técnicas que usan algoritmia de IA (especialmente: Redes neuronales) son vastamente aceptadas por su velocidad y robustez. Existen múltiples aplicaciones que logran el reconocimiento de objetos y realizan tareas mediante algoritmia basada en IA. Entre otras se pueden mencionar: (Burgard et al., 1999; Monar et al., 2014; Garcia y Garcia, 2013; Peña Cabrera et al., 2004; Malpartida y Sobrado, 2003; Egmont Petersen et al., 2002). En aplicaciones donde el robot se desempeña en ambientes muy cambiantes, de gran complejidad, o inexplorados, como por ejemplo en robots de rescate para zonas de catástrofe (Greer et al., 2002; Naidoo et al., 2015; Murphy y Stover, 2008; Kamlofsky et al., 2018), es conveniente otro enfoque.

En Kamlofsky et al. (2018) se presentó, un esquema de visión robótica con reconocimiento de objetos basado en un enfoque topológico (basado análisis de bordes). En el mismo se identifican

tres instancias: calibración de las cámaras, apareo y estereoscopia. En este trabajo se presenta proceso sencillo de reconocimiento de puntos homólogos o apareo, lo cual permitiría poder lograr la reconstrucción tridimensional, que promete ser compatible con requerimientos de tiempo real.

1.2. Objetivos de este trabajo

El objetivo principal de este trabajo es presentar un algoritmo que permita lograr en forma rápida el apareo de puntos entre pares de imágenes estéreo de una misma escena, con muy baja complejidad computacional, de modo de aproximarse a exigentes requerimientos de tiempo real.

Los objetivos intermedios son: presentación de algoritmos para la aproximación poligonal de las curvas de borde, presentación del patrón descriptor de la curva y con él, mostrar una forma de hallar objetos en imágenes. Con ello, el proceso de apareo se logra muy fácilmente, sin elevada complejidad de cálculo.

Un objetivo subyacente de este trabajo es mostrar que el enfoque topológico de análisis de bordes para la identificación de objetos puede ser efectivo, más en aplicaciones con ambientes muy cambiantes, y/o con requerimientos de procesamiento en tiempo real.

1.3. Relevancia del Tema

Las técnicas de visión robótica tienen un abanico cada vez más amplio de aplicaciones. El eficiente modelado tridimensional de escenas en tiempo real y en ambientes muy cambiantes permitiría su implementación en sistemas autónomos en una gran variedad de ambientes: vehículos de transporte, robots humanoides, drones, etc. Visión robótica en tiempo real es un tema abierto, de gran relevancia y de vanguardia tecnológica.

1.4. Estructura del Trabajo

En el capítulo 2 se detalla la metodología de la investigación utilizada para realizar este trabajo. En el capítulo 3 se presenta el marco teórico, donde se definen conceptos básicos obtenidos a partir de literatura clásica del tema en cuestión. El capítulo 4 expone el desarrollo técnico del enfoque abordado para el proceso de visión robótica, detallando cada una de las etapas. En especial, este desarrollo se enfoca en la obtención de puntos homólogos en pares de imágenes estéreo. En el capítulo 5 se presenta una aplicación experimental conteniendo datos experimentales del proceso propuesto y su análisis. Finaliza con la conclusión y trabajos futuros.

2 Metodología de la investigación

2.1. Marco de investigación

Este trabajo se desarrolló a partir de los resultados de las investigaciones iniciadas en el proyecto denominado: *Herramientas de análisis de imágenes digitales para la visión artificial* dirigido por la Dra. María Lorena Bergamini, iniciado en 2012 y radicado en el Caeti.¹

¹ CAETI: Centro de Altos Estudios en Tecnología informática, dependiente de la Facultad de Tecnología informática de la UAI.
Sitio Web: <http://caeti.uai.edu.ar>.

En 2015 estas investigaciones se incorporaron y se continuaron en marco del proyecto binacional llamado “Semi-Autonomous robots for search and rescue operations”. Este proyecto fue seleccionado para ser ejecutado en el trienio 2014-2016, en el marco del Programa de Cooperación Científico-Tecnológica entre el Ministerio de Ciencia, Tecnología e Innovación Productiva de la República Argentina (MINCYT) y el Departamento de Ciencia y Tecnología de la República de Sudáfrica (DST), bajo el código SA – 13/13. El responsable de la parte sudafricana fue el Dr. Glen Bright, del Departamento de Mecánica de la Universidad Kwazulu-Natal, mientras que por la parte de Argentina, el responsable fue el Dr. José Zelasco, del Departamento de Mecánica de la Facultad de Ingeniería de la Universidad de Buenos Aires. Los estudiantes de posgrado Nicol Naidoo (Sudáfrica) y Jorge Kamlofsky (Argentina) han progresado en sus investigaciones. Estudiantes de ingeniería de ambos países han participado en los desarrollos y asistieron a las presentaciones del progreso del proyecto.

2.2. Enfoque metodológico

El plan de desarrollo de esta investigación se orientó hacia el cumplimiento del objetivo principal: presentar algoritmia que permita lograr un rápido esquema de visión 3D que pueda aproximarse a los requerimientos de tiempo real. Para ello se usó un enfoque metodológico cuantitativo (Hernandez et al., 2014) a partir de datos experimentales. Se subdividió al proceso en diferentes tareas alineadas con los objetivos intermedios, para así poder analizar y evaluar avances parciales.

2.3. Algunos resultados que alimentan a este trabajo

El proyecto se inició luego de la presentación de la tesis de grado de Kamlofsky (2011) en la que se expuso acerca de la *Topología digital* y al *análisis de bordes* como un enfoque que podría tratar eficientemente al problema de visión artificial. Esto sentó las bases para el inicio del proyecto. El primer resultado parcial se logró tras la presentación y aceptación en las Jornadas Argentinas de Robótica de un primer avance (Kamlofsky y Bergamini, 2012): la simplificación de las curvas de borde mediante la poligonalización. En Kamlofsky y Bergamini (2013b) se presentó un patrón descriptor de curvas cerradas simplificadas basado en la evolución de la curvatura a lo largo del perímetro. En Kamlofsky y Bergamini (2013a) se incluyó el análisis de la orientación de la forma. En Kamlofsky y Bergamini (2014a y 2014b) y en Bergamini y Kamlofsky (2015) se mostraron aplicaciones.

En 2015 se comenzó a estudiar a la visión en tres dimensiones en el marco del proyecto binacional entre Argentina y Sudáfrica. En Kamlofsky y Bergamini (2015) se presentaron ventajas en el uso de cuaterniones para el cálculo de rotaciones en el espacio. En Bergamini et al. (2016a y 2016b) se lograron publicaciones conjuntas entre el equipo argentino y el sudafricano relacionadas con el proceso de calibración del sistema de visión robótica. En 2017, en Bergamini y Kamlofsky (2017) se presentó otro enfoque para la poligonalización de curvas digitales. En Kamlofsky y Bergamini (2017) se presentó la idea preliminar del rápido hallazgo de puntos homólogos. Esto se incorporó al trabajo de Naidoo et al. (2017) donde se integraron los avances de las investigaciones de ambos equipos. En 2018 se ampliaron estas ideas y se las incorporó en otro artículo conjunto (Kamlofsky et al., 2018). Se presentan mejoras en los tiempos para la obtención de puntos homólogos. En Junio de 2020 el artículo ha sido indexado en Scopus²

2 Vínculo al sitio de Scopus con los detalles del documento: <https://www.scopus.com/inward/record.uri?eid=2-s2.0-85061819621&partnerID=40&md5=567acf7c58e5d96274510690e0a79cbf>.

. En 2021, la totalidad de estas investigaciones permitieron la elaboración de la Tesis presentada para obtener el título de Magister en Tecnología Informática, la cual fue defendida el 9 de Junio de 2021.

3 Marco teórico

3.1. La imagen digital

3.1.1. La imagen analógica (o real)

Cotidianamente se pueden observar innumerables imágenes bidimensionales. Por ejemplo: la pintura hecha por un artista, una imagen natural capturada por una cámara, un telescopio o un microscopio. Representan una variación continua de sombras y tonos. Por esta razón, las imágenes de este tipo reciben el nombre de imágenes continuas o analógicas (Rosenfeld, 1979).

3.1.2. Digitalización: El píxel

Para almacenar estas imágenes en una computadora es necesario digitalizarlas. Se inicia en una primera etapa denominada Adquisición: ciertos dispositivos con sensores ópticos sensibles son capaces de obtener variaciones de tonos y sombras. En Rosenfeld y Kak (2014) se brindan más detalles acerca del proceso de adquisición de imágenes. Luego se genera una imagen bidimensional y se la almacenan electrónicamente. A este proceso se lo denomina *digitalización*. Algunos dispositivos de adquisición o digitalización de imágenes son: scanners, cámaras fotográficas digitales, tomógrafos. Matemáticamente, la digitalización es una transformación de un subconjunto del dominio real infinito (objeto o escenario analógico) al codominio del plano digital, discreto y finito: la imagen digital. Durante este proceso, se divide a la imagen analógica obtenida en celdas a las que se les asigna un determinado color. A cada una de estas celdas se las denomina píxel (contracción del inglés de las palabras *picture element*).

3.1.3. La imagen digital

Una imagen digital es una función $f: A \subseteq \mathbb{Z} \times \mathbb{Z} \rightarrow \mathbb{Z}^n$, $n \in \mathbb{N}$ que asigna a cada píxel $(x,y) \in A$ un valor $f(x,y) \in \mathbb{Z}^n$ que representa el color de dicho píxel, con $\mathbb{Z}$: el conjunto de los números enteros. Mientras que (x,y) brinda las coordenadas espaciales del píxel, el valor $f(x,y)$ representa su nivel de brillo. El valor n queda determinado según el modo de confección de la imagen digital. Indica en cuántas componentes se descompone al brillo. Si $n = 1$, el brillo se descompone en su escala de grises mientras que si $n > 1$ al brillo se lo descompone en n bandas del espectro de colores (Rosenfeld y Kak, 2014).

3.1.4. Matriz asociada a una imagen digital

Una imagen digital puede considerarse como una matriz cuyos índices de filas y columnas identifican el punto en la imagen, y su valor correspondiente identifica el nivel de gris en ese punto. A dicha matriz se la llama la matriz asociada a la imagen digital. Entonces, una

imagen de $N \times M$ píxeles tiene asociada una matriz $A \in M \times N$. Y la matriz A se conformará de la siguiente manera:

$$\forall (x, y) \text{ con: } 0 \leq x < M \wedge 0 \leq y < N \wedge x \in \mathbb{Z} \wedge y \in \mathbb{Z}: \quad (1a)$$

$$A = f(x, y) = \begin{pmatrix} f(0,0) & f(0,1) & \dots & f(0,N-1) \\ f(1,0) & f(1,1) & \dots & f(1,N-1) \\ \vdots & \vdots & \ddots & \vdots \\ f(M-1,0) & f(M-1,1) & \dots & f(M-1,N-1) \end{pmatrix} \quad (1b)$$

3.2. El procesamiento de imágenes

3.2.1. Algunas causas por las que se requiera el mejoramiento de imágenes

Imágenes con bajo contraste

Entender el contenido de una imagen digital parte de identificar en la misma sus componentes. Para ello, es conveniente que sus límites sean claros. En imágenes digitales de bajo contraste, los límites en las componentes no son fáciles de identificar.

El problema del ruido

La digitalización es una transformación de un objeto analógico del dominio real infinito al codominio digital, discreto y finito. Esta transformación logra que objetos reales con infinitos detalles puedan almacenarse en un espacio finito, lo que supone una gran pérdida de información. Esa gran pérdida de información se refleja en la aparición de ruido: la resolución de un dispositivo de adquisición de imágenes define cuántos píxeles por unidad de longitud presentará en la imagen, del objeto analógico original, con infinitos píxeles por unidad de longitud. El ruido se evidencia notoriamente en las fronteras de los objetos donde pueden observarse formas de zig-zag o dientes de serrucho en partes que en la realidad son bordes rectos o suaves. Sin embargo, el ruido, tiene también otras fuentes: problemas del dispositivo de captura, mala iluminación, entre otras.

3.2.2. Técnicas básicas para el tratamiento de imágenes digitales

Segmentación

Cuando se pretende obtener información de una imagen es común referirse a esta etapa como análisis de la imagen. La primera parte en el proceso de análisis de imágenes es la Segmentación que consiste en subdividir a la imagen en partes: fondo y objetos. La segmentación automática es un problema de difícil tratamiento. El nivel y la forma en que se realiza la segmentación depende del problema en sí (Gonzalez & Woods, 1993). Por eso, es muy común incluir en los equipos de adquisición sensores y filtros, como ser cámaras térmicas, cámaras con filtros infrarrojos, etc.

Los algoritmos de segmentación para imágenes en escalas de grises generalmente se basan en dos propiedades básicas: la similaridad y la discontinuidad. La primera permite agrupar píxeles vecinos similares en un mismo objeto. Discontinuidades bruscas pueden indicar bordes o fronteras del objeto. Un enfoque muy sencillo llamado *segmentación por umbralizado* se presentó en Rosenfeld (1979). El objetivo consiste en separar a los objetos de interés del fondo de la imagen. Se determina un valor de gris umbral (dependiente del problema).

Luego se recorre la imagen píxel a píxel. Si el valor de gris del píxel es menor que el umbral, se dice que el píxel pertenece al objeto (y viceversa), y se lo marca como tal.

Sea U : valor de gris umbral. Sea $f(x,y)$ el valor de gris de la imagen en el píxel (x,y) , el umbralizado en el píxel (x,y) , puede definirse:

$$u(x, y) = \begin{cases} 0 & \text{Si } f(x, y) < U \\ 255 & \text{Si } f(x, y) \geq U \end{cases} \quad (2)$$

La función de umbralizado genera en modo RGB una imagen blanco y negro, donde con negro se destacan los píxeles de interés. Esta idea se puede extender a varios umbrales: En imágenes que poseen diferentes valores de gris, se desea determinar zonas que poseen colores similares. Para ello, se determinan varios valores de color umbral según las características de las imágenes. A esto se lo denomina *umbralizado múltiple*. El umbralizado múltiple se utiliza en el tratamiento de imágenes adquiridas con cámaras y/o dispositivos de adquisición especiales, como ser: en imágenes adquiridas mediante cámaras térmicas, donde a cada rango de temperatura se le asigna un color específico. Se lo utiliza también, en imágenes de tomografías computadas, resonancias magnéticas con o sin contraste, entre otras.

Estiramiento de contraste

Es un proceso que facilita la detección de discontinuidades y similitudes ya que en las zonas de frontera o de cambios, los valores de colores suelen ser muy similares, lo que promueve a la formación de ruido. El objetivo de esta técnica consiste en obtener una imagen con mayor contraste que la original. Y esto se logra haciendo más oscuros a los píxeles que tienen valor de gris menor que un cierto valor de gris frontera y más claros a los mayores haciendo más nítidas las zonas de frontera y evitando ruidos no deseados, unificando los colores similares. Resulta en una imagen nítida, con delimitaciones claras en las fronteras, y de fácil tratamiento.

Algunas operaciones con imágenes

Sustracción de imágenes: A una imagen obtenida se le puede restar una imagen previamente adquirida. Puede usarse para detección de movimientos (Gonzalez & Woods, 1993).

Promediado de imágenes: Si se considera una imagen ruidosa, el ruido puede atenuarse mediante la conformación de una nueva imagen lograda tras el cálculo del promedio de n imágenes ruidosas.

Uso de máscaras especiales: Se usan algunas máscaras para suavizado o difuminado, para acentuar detalles, para detectar puntos aislados, líneas o bordes (Gonzalez & Woods, 1993).

3.3. Transformaciones lineales aplicadas a imágenes digitales

Una transformación lineal (TL) se puede aplicar a un vector posición o a un conjunto de vectores posición. Por ser TL una transformación lineal, existirá una matriz asociada a la TL u operadora de la TL. Eso significa que esa matriz pre-multiplica a una matriz que contiene a los vectores posición, de modo que con una sola operación matricial se logra la transformación lineal de un conjunto de vectores. Dado que un polígono puede definirse a partir de sus vértices ordenados, al polígono se lo puede definir como una matriz donde cada columna corresponde con el vector

posición de cada vértice. De este modo, puede aplicarse una TL sobre todo un polígono con una sola operación.

Ejemplo 1: Si se conoce que una cámara está rotada 5° en comparación con otra cámara que apunta al mismo objeto y luego se desean comparar sus imágenes, generalmente se comenzaría con alinear la orientación de la imágenes, rotando la primer imagen 5° en sentido opuesto.

Ejemplo 2: En aplicaciones que usan scanners automáticos de alimentación múltiple, suele suceder que el mecanismo de transporte de las hojas posea algún desbalance mecánico. Todas las imágenes aparecen algo rotadas. Se corrige fácilmente aplicando la rotación inversa correspondiente.

3.4. Reconocimiento de Objetos en Imágenes

En esta sección se presentan dos enfoques distintos comúnmente usados cuyo objetivo es el mismo: obtener información de la imagen digital para poder reconocer objetos dentro de éstas: un enfoque basado en la Topología Digital y otro basado en el uso de técnicas de Inteligencia Artificial (IA). El enfoque de la topología digital permite extraer propiedades topológicas de un objeto. Es posible determinar sus bordes, los cuales pueden analizarse tanto topológicamente como geoméricamente. La algoritmia es muy sencilla y de baja complejidad computacional. El enfoque de IA hace uso de diferentes técnicas. En particular, en el uso de redes neuronales para su uso en el reconocimiento de objetos imágenes está ampliamente difundido y aceptado. Dado que este trabajo se basa en el primero de los enfoques mencionados, se lo presenta con mayor profundidad.

3.4.1. Enfoque Basado en Topología Digital

En este apartado se hace una introducción a los conceptos de Topología Digital, los cuales se aplican luego de realizada la segmentación de la imagen. Permiten atribuirle a los objetos características topológicas: se permite tratar a ciertos píxeles de una imagen como un conjunto conexo, con borde o esqueleto, con o sin agujeros, y demás características topológicas. Se pretende aplicar propiedades topológicas del plano real a un subconjunto del plano digital: la imagen digital.

La Topología de una imagen digital

El Plano digital: Z^2 es el conjunto formado por los pares ordenados de números enteros.

Vecindades: Dado un punto $P(x,y)$ del plano digital Z^2 , pueden definirse a sus vecinos como:

$$4\text{-vecinos} = \{(x \pm 1, y), (x, y \pm 1)\} \quad (3a)$$

$$8\text{-vecinos} = \{(x \pm 1, y), (x, y \pm 1), (x+1, y+1), (x-1, y+1)\} \quad (3b)$$

Entonces, la 4-vecindad es el conjunto de 4-vecinos y la 8-vecindad es el conjunto de 8-vecinos.

Topología del plano digital:

$$U_{(p)} = \begin{cases} \{P\} & \text{si } x+y \text{ es impar} \\ \{Q: Q \text{ es } 4\text{-vecino de } P\} & \text{si } x+y \text{ es par} \end{cases} \quad (4)$$

Sea $B=\{U_{(p)}: \in \mathbb{Z}^2\}$, B es base para una topología en $\mathbb{Z}^2$ (Kamlofsky, 2011; Rosenfeld, 1979). Así, al plano digital se le asigna una topología a partir de definida la base B . La topología digital τ de $\mathbb{Z}^2$ es la topología generada por la base B .

Topología de una imagen digital: Una imagen digital π de $M \times N$ píxeles es un subconjunto finito de $\mathbb{Z}^2$ tal que

$$\pi = \{(x,y) \in \mathbb{Z}^2 / 0 \leq x \leq N-1 \wedge 0 \leq y \leq M-1\} \quad (5)$$

El borde o frontera de π es:

$$Fr(\pi) = \{(x,y) \in \pi / x=0 \vee x=N-1 \vee y=0 \vee y=M-1\} \quad (6)$$

Sea:

$$B_{\pi} = \{U_{(p)} \cap \pi, P \in \mathbb{Z}^2\} \quad (7)$$

B_{π} es base. La topología τ_{π} de la imagen digital π es aquella generada por la base B_{π} .

Los objetos de una imagen digital:

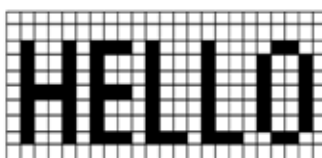
La segmentación (o binarización) divide a la imagen digital en dos: el subconjunto de los objetos y el subconjunto del fondo de la imagen. Una vez que la imagen ha sido segmentada en subconjuntos, el paso siguiente consiste en establecer sus propiedades y relaciones entre ellos.

Ejemplo 3: la ilustración 1 muestra una imagen de un texto: luego de separar las letras del fondo, interesa identificar a cada letra. Algunas propiedades pueden obtenerse fácilmente como el área del subconjunto objeto. Pero como otras propiedades son netamente topológicas, se necesitan conceptos topológicos. Estas propiedades son útiles ya que, una vez que el subconjunto fue identificado (texto), se necesita separarlo en regiones conexas que se correspondan con cada letra.

Se considera que una imagen digital π está segmentada en dos conjuntos: el conjunto S que contiene a los objetos de la imagen en sí, y su complemento S^c que representa el fondo. Se asume que $S \cap Fr(\pi) = \emptyset$. La frontera de S es el conjunto de puntos que tienen 4-vecinos en S . Con $S = \{S_1, S_2, \dots, S_j\}, j \in \mathbb{N}$, cada S_i es un objeto dentro de la imagen π con $i \in [1, 2, \dots, j]$.



(a)



(b)

Conectividad

Caminos: Sean P y Q dos puntos de la imagen π . Un camino de P a Q es una sucesión finita de puntos $P = P_0, P_1, \dots, P_n = Q$ pertenecientes a π : P_i es vecino de P_{i-1} para $1 \leq i \leq n$. Se presentan dos definiciones en una: si tratamos 4-vecinos u 8-vecinos tendremos 4-caminos u 8-caminos.

Conjunto conexo: Un conjunto S_i es conexo, si para todo par de puntos de S_i existe un camino que los una conformado por puntos dentro de S_i . Cada S_i es considerado como un objeto en la imagen. Para trasladar adecuadamente las propiedades topológicas del plano real al plano digital, si se considera 4-conectividad para S (con 4-caminos), debe considerarse 8-conectividad (con 8-caminos) para S^C u 8-conectividad para S y 4-conectividad para S^C (Rosenfeld, 1979).

Componente conexa: A cada S_i se la llama también “*componente conexa de S* ” o simplemente “*componente*”.

Agujeros: Sea $S \in \pi$, $S \neq \emptyset$. El fondo de π es la única componente conexa de S^C que contiene a $Fr(\pi)$. Las demás componentes de S^C , si existen, reciben el nombre de “*agujeros de S* ”.

Arcos: $S \in \pi$, $S \neq \emptyset$ es un arco si es conexo y todos, salvo dos de sus puntos (sus extremos) tienen exactamente dos vecinos en S , mientras que los dos extremos tienen exactamente uno. Más precisamente, un arco S es un camino $P_0, P_1, \dots, P_n$ formado por puntos distintos y tal que P_i es vecino de P_j sí y solo sí $i = j \pm 1$. Un arco es también un camino abierto que no se autointersecta.

Curvas: $S \in \pi$, $S \neq \emptyset$ es una curva si es conexo y todos sus puntos tienen exactamente dos vecinos en S . Más precisamente, una curva S es un camino $P_0, P_1, \dots, P_n$ formado por puntos distintos tal que P_i es vecino de P_j y son iguales al inicio y al final. Es decir: $P_0 = P_n$. Puede considerarse a una curva como un camino abierto que se autointersecta al inicio y fin.

El teorema de la curva de Jordan: En la topología usual el teorema de la curva de Jordan afirma que una curva simple cerrada C en el plano divide a éste en dos regiones: una acotada, llamada “el interior de C ” y otra no acotada, llamada “el exterior de C ”, siendo C la frontera común. Se pretende trasladar este resultado al plano digital. Rosenfeld (1979) presentó en el siguiente teorema:

Sea S una 4-curva en el plano digital. Entonces S^C tiene exactamente dos componentes 8-conexas. Una de ellas es acotada, llamada “el interior de S ” y la otra es no acotada, llamada “el exterior de S ”. Análogamente para 8-curvas.

Este resultado se encuentra probado en Kong et al. (1991). Una consecuencia del teorema es que una curva S tiene exactamente un agujero. Otra consecuencia es que si S es una 8-curva, entonces el agujero y el fondo son topológicamente conexos. Hay otro resultado obvio pero útil: Todo punto P de una curva $S \in \pi$ es adyacente (en el sentido de conectividad de S) a las dos componentes de S^C .

Extrayendo características de un objeto tras analizar arcos y curvas

Dada una componente, se desea simplificarla sin perder sus características básicas, de modo que sea posible describirla eficientemente. Puede pensarse en dos enfoques: la obtención de sus bordes para su análisis, o la simplificación de la componente a su esqueleto. En ambos casos, se obtienen representaciones codificadas más simples que el objeto en sí, manteniendo invariantes sus propiedades topológicas. Esto resulta muy útil para identificar objetos dentro de imágenes digitales.

Borde de una componente respecto a una componente de su complemento: Sea C una componente conexa de S y D una componente conexa de S^C . El borde de C respecto a D es el conjunto: $C_D = \{P \in C / P \text{ tiene al menos un 4-vecino en } D\}$.

Afinamiento de una componente: Consiste en reducir una componente a un arco o a una curva mediante la eliminación sucesiva de puntos simples (Rosenfeld, 1979). Si la componente posee un agujero, el proceso de afinamiento simplifica la componente a una curva. Si no posee agujeros, la componente se reduce a un arco.

3.4.2. *Uso de Técnicas de Inteligencia Artificial con imágenes.*

En la industria es común el uso indistinto de los términos IA (Inteligencia Artificial) y ML (iniciales en inglés de Machine Learning). IA se define como la teoría y el desarrollo de sistemas informáticos que pueden realizar tareas que normalmente requieren inteligencia humana, como la percepción visual, el reconocimiento del habla o la toma de decisiones (entre otras). El aprendizaje automático (ML) es una aplicación de IA que proporciona a los sistemas la capacidad de aprender y mejorar automáticamente a partir de la experiencia, sin ser explícitamente programados para ello (Gil y Liska, 2019). En general, una solución de IA pretende resolver en tiempo de ejecución, problemas que normalmente no podrían resolverse en corto plazo. Pueden distinguirse, entonces, dos instancias: entrenamiento del modelo y utilización del modelo. En la etapa de entrenamiento, se ingresa a un algoritmo de IA un conjunto de datos adecuadamente formateados y etiquetados. Con estos datos, se entrena al modelo, que se usa para predecir nuevas instancias.

Algunas técnicas y algoritmos de IA

Los algoritmos de ML más usados son (Velogig, 2020; Hernandez Orallo et al, 2004):

Reducción de Dimensionalidad: dado un conjunto de datos con muchas variables, se trata de reducir la cantidad de variables sin perder gran calidad predictora sobre la variable objetivo.

Cluster Analysis: Segmenta elementos que son similares en algún sentido. Se aplica ampliamente en Marketing y Empresas, que desean segmentar el comportamiento de sus clientes y productos.

Regresión: Consiste en aprender una función lineal o polinomial que asigna a cada instancia un valor real. Son modelos predictivos usan para predecir valores numéricos.

Reglas de asociación: Son métodos para descubrir nuevas relaciones no explícitas entre variables de una gran base de datos. Son muy utilizadas en marketing y en publicidad.

Árboles de Decisión: Se construyen diagramas lógicos en formas de árbol, leyéndose de arriba hacia abajo, hasta llegar a una decisión (hojas).

Redes Neuronales: Son modelos que imitan el funcionamiento del cerebro humano. Permite modelar problemas complejos en los que puede haber interacciones no lineal entre las variables.

Aplicaciones de redes neuronales en el procesamiento de imágenes

Se mencionan algunas instancias de aplicación de redes neuronales, desarrolladas para resolver diferentes problemas del procesamiento de imágenes (Egmont-Petersen et al., 2002):

Restauración de imagen: En general, se quiere restaurar una imagen distorsionada. El sistema de adquisición puede introducir ruido, desenfoque de movimiento, distorsión, etc.

Mejoramiento de imagen: Se logra amplificando las características perceptivas.

Detección de bordes: Aplicaciones de redes neuronales, actúan como detectores de bordes.

Compresión de imagen: Se presentan diferentes enfoques para la compresión de imágenes.

Extracción de características: Hay un amplio rango de aplicaciones que usan redes neuronales para la extracción de características.

Segmentación: El propósito de la segmentación es asignar etiquetas a píxeles individuales. Al tratarse de un problema de clasificación, los enfoques basados en redes neuronales son apropiados.

Comprensión de la imagen: Combina técnicas de segmentación o reconocimiento de objetos con el conocimiento del contenido de imagen esperado.

Reconocimiento de objetos: En la mayoría de las aplicaciones, las redes neuronales han sido entrenadas para localizar objetos individuales basados directamente en datos de píxeles.

Deep Learning, aprendizaje profundo o reforzado.

Deep Learning hacen uso de la disponibilidad de grandes volúmenes de datos de múltiples orígenes (Big Data) para que los sistemas puedan realizar un aprendizaje mucho más profundo y detallado. En las últimas décadas, las redes sociales como Flickr³, Facebook⁴, Twitter⁵ y diversos servicios digitales “gratuitos” como ser Google⁶, Microsoft⁷ Azure⁸, Amazon⁹, han obtenido a diario, y en forma gratuita y voluntariamente, de usuarios de todo el mundo, millones de archivos conteniendo información textual o multimedial que les permite lograr modelos mucho más precisos (Zhang et al., 2018). Muchos de ellos construyeron enormes plataformas en la nube para desarrollo rápido de soluciones de alcance global.

Algunos usos de Deep Learning en imágenes

Se mencionan dos ejemplos de aplicaciones de Deep Learning en Imágenes. En particular, se eligió la plataforma de Amazon: AWS¹⁰ (Amazon Web Services). Dentro de esta plataforma se

3 Sitio de Flickr: <https://www.flickr.com/photos/tags/flickr/>

4 Sitio oficial de Facebook: <https://about.fb.com/news/>

5 Sitio de Twitter: <https://twitter.com/>

6 Sitio de Google: <https://www.google.com/>

7 Sitio de Microsoft: <https://www.microsoft.com/>

8 Sitio de Microsoft Azure: <https://azure.microsoft.com/es-es/>

9 Sitio de Amazon: <https://www.amazon.com/>

10 Sitio de AWS en español: <https://aws.amazon.com/es/>

presentan aquí, dos aplicaciones: Textract y Rekognition. La primera de ella extrae los textos presentes en una imagen, mientras que la segunda reconoce objetos en imágenes.

AWS Textract¹¹:

Es un servicio que detecta y extrae automáticamente textos y datos de documentos escaneados. A diferencia de los simples OCR (siglas del inglés: Optical Character Recognition), Textract extrae el contenido rápida y eficientemente con muy alta confianza, identificando campos y ubicación. Los resultados se pueden presentar en archivos json lo cual permite su tratamiento posterior.

AWS Rekognition:

Es una aplicación de Machine Learning de AWS que permite reconocer en una escena (presentada desde una foto) objetos y sus atributos y características en imágenes, rápidamente.

3.5. Modelado tridimensional

3.5.1. Calibración

Para realizar la reconstrucción tridimensional de una escena a partir de múltiples imágenes mediante estereoscopía, éstas deben calibrarse. Hay muchos métodos que han sido propuestos para la calibración de cámaras. En Bergamini et al. (2016b) se propuso un esquema de calibración sencillo que es fácil de implementar. Mediante el uso de la matriz esencial (Kalantary & Jung, 2008), se obtiene la base y la rotación como solución de un sistema de ecuaciones lineales (Bergamini et al., 2016b). Los parámetros de calibración son necesarios para obtener con precisión las coordenadas tridimensionales de los puntos de la escena que se obtienen mediante estereoscopía.

3.5.2. Obtención de puntos homólogos

La identificación automática de puntos homólogos obtenida a partir de dos o más imágenes digitales es un problema esencial para lograr un modelo tridimensional de la escena mediante estereoscopía. Estos puntos corresponden a las proyecciones en dos o más imágenes de un mismo punto físico de la escena. A este problema se lo llama *apareo estéreo* (o simplemente *apareo*) o *puesta en correspondencia* (Donadio y Mendez, 1997). Es la tarea más compleja en el proceso de visión 3D.

Algunas condiciones para lograr el apareo

Epipolaridad: Los puntos homólogos deben estar sobre la misma línea epipolar, es decir, los puntos principales de cada imagen deben estar en la misma línea (imágenes roto-rectificadas).

Similitud: Este es el principio básico para resolver el apareo. Establece que los elementos homólogos deben poseer características similares en ambas imágenes. Es decir, ambas

¹¹ AWS Textract en español: <https://aws.amazon.com/es/textract/>

imágenes deben ser “parecidas” y los ángulos entre los planos de las cámaras deben ser pequeños.

Continuidad figural: Se espera que los objetos no posean oclusiones en ambas imágenes.

Tipos de algoritmos de apareo

Basados en áreas: Emplean como primitiva la intensidad de los píxeles y su estrategia de apareo se aplica en forma local sobre los píxeles de la imagen.

Basados en características: Seleccionan un conjunto de primitivas abstractas y aplican el apareo sobre ese conjunto asumiendo que las propiedades geométricas de la escena son invariantes.

3.5.3. Estereoscopia

La fotogrametría o metrología con imágenes estereoscópicas o simplemente estereoscopia es un conjunto de técnicas que mediante una cámara fotográfica, permiten deducir una proyección cónica de la imagen. Es posible proveer con precisión conocida: dimensiones, orientación y ubicación de objetos en el espacio 3D utilizando medidas hechas en dos o más imágenes. Partiendo de dos imágenes de una misma escena, conociendo sus puntos homólogos, mediante el uso de la estereoscopia es posible realizar su reconstrucción tridimensional. Cuantos más puntos homólogos de la escena se conozcan, mejor resultará su modelo tridimensional (Zelasco et al., 2000).

4 Desarrollo técnico de la propuesta

4.1. Enfoque y abordaje del desarrollo técnico

Se presenta un proceso que permite obtener rápidamente los puntos homólogos de un objeto presente en dos imágenes de una misma escena resumido en la Ilustración 2. Las condiciones de epipolaridad, similaridad y continuidad figural son necesarias para lograr el apareo entre ambas imágenes. El tipo de algoritmo desarrollado en esta propuesta es incremental, basado en características, las cuales se van hallando mientras se recorren las curvas de borde.

Generalmente, *búsqueda de puntos homólogos*, *apareo* y *puesta en correspondencia* se denomina al mismo proceso que entrega los pares de correspondencia de puntos de un objeto entre las imágenes. En este trabajo, al proceso global aquí se lo llama *búsqueda de puntos homólogos*, al proceso que parte de las imágenes analizadas y finaliza del mismo modo que el proceso global se lo denomina *apareo*, mientras que la tarea que solo relaciona la correspondencia entre puntos de las imágenes de objetos ya reconocidos, se la denomina *puesta en correspondencia*.

4.2. El Sub-proceso de interpretación de imagen

El sub-proceso denominado *interpretación de imagen* se inicia con la captura de la imagen y finaliza con la presentación de una colección de patrones descriptores de los objetos hallados.

4.2.1. Captura, pre-procesamiento y binarización de la imagen

El tipo de cámara adecuado depende del problema. Para esta propuesta, se supone que las imágenes se reciben sin necesidad de ser mejoradas, y en condiciones para realizar la binarización. La binarización o segmentación se realiza mediante umbralizado simple, tal como fue definido en Rosenfeld (1979). Luego, los píxeles que tienen valor 255 se los agrupa conformando el fondo, mientras que los píxeles con valor 0 se agrupan conformando al conjunto de objetos de interés.

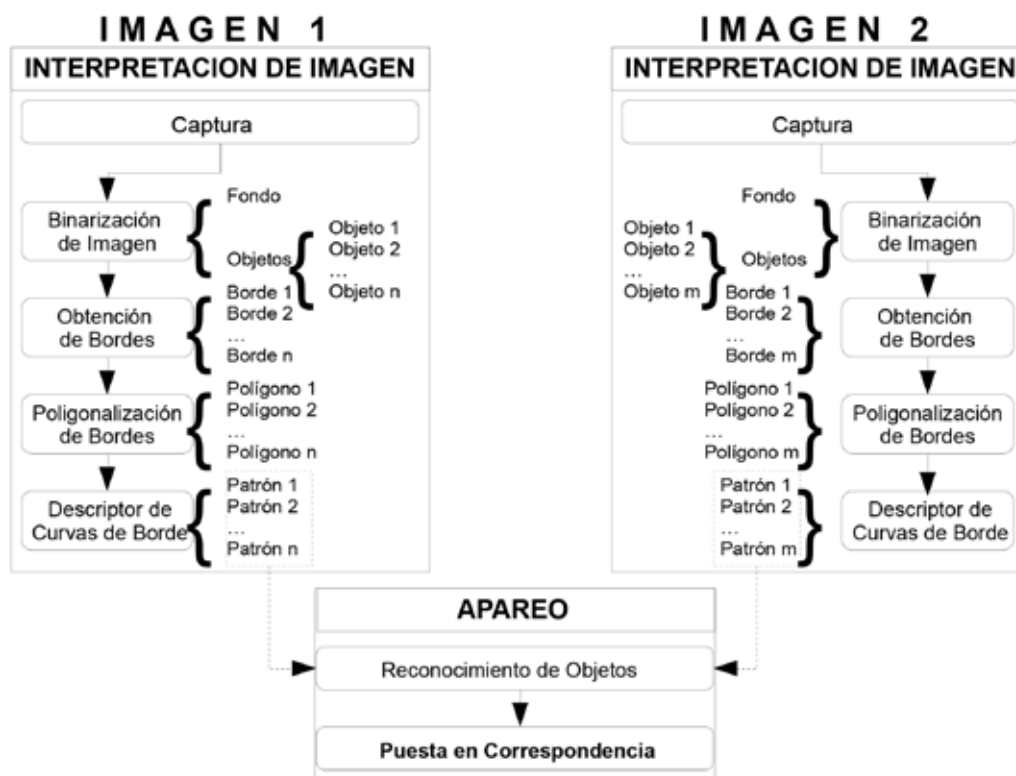


Ilustración 2: Esquema del proceso de obtención de puntos homólogos

4.2.2. Obtención de puntos de borde: el algoritmo BF4/8

Obtener los bordes de un objeto permite extraer muchas de sus características con una cantidad reducida de puntos sin perder muchos atributos. Con el algoritmo BF4/8 se obtienen los puntos de borde de los objetos en una imagen binarizada.

Sea C una componente conexa de S (es decir, un objeto) y D una componente conexa de S^c (es decir, una componente del fondo), BF4/8 permite obtener el borde de C respecto de D . El algoritmo fue presentado originalmente por Rosenfeld (1979).

Resumen del Algoritmo BF8: Dado un par de puntos $(P_i; Q_i)$ con $P_i \in C$ y $Q_i \in D$ con P_i 4-vecino de Q_i , el algoritmo determina un nuevo par de puntos $(P_{i+1}; Q_{i+1})$ con $P_{i+1} \in C$ y $Q_{i+1} \in D$ con P_{i+1} 4-vecino de Q_{i+1} y P_{i+1} 8-vecino de P_i . Así se visitarán todos los puntos de C_D . Inicia con el par de puntos $(P_0; Q_0)$. Sea R_0 la 8-vecindad de P_0 ordenada en sentido horario, partiendo desde Q_0 . $R_0 = Q_0; R_{02}; R_{03}; \dots; R_{08}$. Sea R_{0j} el primer elemento de R_0 perteneciente a C , y 8-vecino de P_0 . Como $R_{0j-1} \in D$: $(P_1; Q_1) = (R_{0j}; R_{0j-1})$. El proceso se repite sobre los restantes puntos. Finaliza cuando: $(P_i; Q_i) = (P_0; Q_0)$.

Una explicación de BF4, con ejemplos y una explicación más amplia de ambos algoritmos puede hallarse en el trabajo de Rosenfeld (1979) y en Kamlofsky (2021).

4.2.3. Simplificación de las Curvas de Borde

El análisis de bordes de un objeto permite estudiar eficientemente la forma de un objeto en una imagen digital: sólo se analiza una pequeña cantidad de puntos del mismo, de modo que ese conjunto de puntos obtenidos mantiene ciertas propiedades topológicas respecto del objeto original. Las curvas de bordes de los objetos pueden simplificarse mediante una transformación de dichas curvas a polígonos aproximantes: se logran aproximaciones adecuadas de las curvas que nos aseguran robustez algorítmica y baja complejidad computacional (De Berg, 1997). Además se permite reducir efectos de ruido causados por la digitalización. El grado de aproximación se controla con un parámetro ϵ , que permite manejar el balance entre el error de la aproximación, y la complejidad del polígono resultante (Kamlofsky y Bergamini, 2012).

Pre-poligonalización: La pre-poligonalización es una simplificación de la curva de borde, eliminando puntos que están alineados. Equivale a una aproximación con tolerancia $\epsilon = 0$.

Ejemplo 4: En la ilustración 3 se presenta la Pre-poligonalización de una forma en una imagen sintética correspondiente al dibujo de un automóvil.

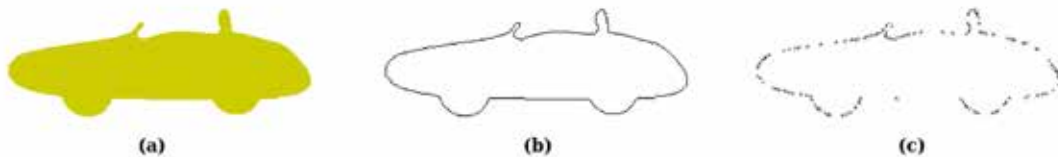


Ilustración 3: Pre-poligonalización de una forma. (a) Forma tratada. (b) Sus bordes obtenidos mediante BF8. (c) Puntos restantes tras el procedimiento.

En la ilustración 3 (a) se presenta el objeto a analizar, el dibujo de un automóvil con 14971 píxeles. En (b) se muestra el borde obtenido por BF8. La cantidad de puntos del borde es de 648 píxeles. (c) Luego de la pre-poligonalización: 212 píxeles, que representa el 32% de los puntos del borde.

Poligonalización: Se recibe un conjunto de puntos que corresponden a un borde de una forma pre-poligonalizada. Dado un factor de tolerancia, se realiza la poligonalización del borde. Resulta en un conjunto de puntos mucho más reducido que el anterior: los vértices del polígono aproximante.

Funcionamiento del algoritmo: Se inicia con el primer punto P_0 . Se van agregando puntos al lado mientras el ancho de la cáscara convexa del lado sea menor que una tolerancia ϵ : método admitirPuntoEnLado. Si para un punto P_i , el ancho de la cáscara convexa de todos los puntos del lado es mayor a ϵ , el punto P_0 y el punto P_{i-1} , son los extremos del primer lado y se eliminan los puntos restantes. Se repite a partir de P_{i-1} , y finaliza cuando el último extremo sea P_0 .

La cáscara convexa: Dado un conjunto de puntos en el plano X , la cáscara convexa o envoltura convexa de X se la define como la intersección de todos los conjuntos convexos que contienen a X . Una buena analogía para entender este concepto se presenta en De Berg (1997): imagine que tiene un conjunto de clavos clavados en una tabla plana, y luego encierra los clavos con una banda elástica. El polígono que encierra a todos los clavos con la banda elástica es la cáscara convexa.

Ejemplo 5: En la ilustración 4 se presenta un ejemplo de poligonalización de una curva cerrada. En este ejemplo, se muestra el resultado de aplicar el algoritmo a la imagen original mostrada en la ilustración 24, con tolerancias $\epsilon_1 = 2$ y $\epsilon_2 = 5$ (en píxeles). Con tolerancia $\epsilon = 2$ píxeles se obtuvo una frontera simplificada consistente de 48 píxeles (a), y la forma del objeto mantiene las características relevantes de la figura original (b). Con tolerancia $\epsilon = 5$ píxeles, el borde se reduce a 22 píxeles (c). Obviamente, la calidad de la aproximación disminuye (d).



Ilustración 4: (a, b) Vértices y polígono aproximante del objeto con tolerancia de 2 píxeles. (c, d) Vértices y polígono aproximante del mismo objeto con tolerancia de 5 píxeles.

Mediante el procedimiento de poligonalización de las curvas de borde de una forma, se logra una simplificación notable de la misma. En el ejemplo presentado, con 22 píxeles se logra una representación simplificada de un objeto de 14971 píxeles, a un costo computacional muy bajo.

Conclusiones acerca del método de simplificación: El método propuesto para la simplificación del borde de un objeto digital permite representar al objeto con una cantidad reducida de puntos, manteniendo las características principales de su forma. Además, mediante la poligonalización de curvas de borde se logra eliminar efectos adversos de la digitalización: el ruido. El uso de envolturas convexas permite calcular el ancho de un conjunto de puntos de manera eficiente. El control de la tolerancia queda en manos del usuario del algoritmo, quien podrá definirla según los requerimientos de la aplicación. Así, con baja tolerancia pueden obtenerse resultados de alta calidad, mientras que con tolerancias mayores el usuario gana en velocidad de procesamiento. Esta herramienta es fundamental en el proceso de reconocimiento de objetos en movimientos, donde los requerimientos de eficiencia y flexibilidad son de importancia relevante.

4.2.4. Patrón descriptor de las curvas de borde

A partir de disponer de la representación simplificada de la curva de borde mediante su poligonalización, se espera extraer sus características distintivas, para poder identificar al objeto. Se presenta un patrón descriptor de curvas cerradas. El mismo consiste en una función lineal a trozos de la evolución de la curvatura en función de la evolución del perímetro del polígono. Cada trozo es lineal debido a que la curvatura en los lados de un polígono es cero. El ángulo en cada uno de sus vértices se considera un estimador de la curvatura. Este estimador permite identificar características geométricas del objeto, como ser concavidad, convexidad, puntos angulosos, etc, mediante el análisis de la evolución de dicho parámetro a

lo largo de la curva. Así, se obtiene una representación simplificada de la forma en el espacio longitud-curvatura, y es poco sensible al ruido.

El Patrón de evolución del ángulo de giro

Dado el polígono obtenido como aproximación de una curva digital, se propone una representación del mismo basado en el patrón de evolución de ángulo de giro, que se explica a continuación. Sea C una curva regular parametrizada en el plano xy con: $\alpha(s) = (x(s); y(s))$ donde s es el parámetro longitud de arco, $s \in [0, L]$ con L : Longitud de la curva. La curvatura en el punto $\alpha(s)$ es $\kappa(s)$, el cambio instantáneo de la inclinación del vector tangente a C en $\alpha(s)$. Sea $l(s)$ la longitud de la curva C desde $\alpha(0)$ hasta $\alpha(s)$. Sea: $\lambda(s) = l(s) / L$ la longitud normalizada de la curva C . Así, $\lambda(0) = 0, \lambda(L) = 1$. La curvatura acumulada en s es:

$$\kappa_{acum}(s) = \int_0^s \kappa(r) dr \quad (9)$$

Entonces, $\kappa_{acum}(0) = 0$ y $\kappa_{acum}(L) = 2\pi$ para curvas simples cerradas. Sea $\beta(s)$ la curva en el plano longitud-curvatura y $\kappa(s)$ la curvatura acumulada en función de la longitud normalizada:

$$\beta(s) = \kappa_{acum}(\lambda(s)) \quad (10)$$

Extensión a polígonos: Se extiende estas definiciones para el caso en el que C sea un polígono con vértices $v_1, v_2, \dots, v_n$. En un polígono solo tiene sentido definir la curvatura en sus vértices (ya que en los lados es cero). La curvatura en el vértice v_i es el ángulo interno al polígono en dicho vértice (o ángulo de giro) formado por los lados definidos por los vértices adyacentes: v_{i-1} y v_{i+1} . La curvatura acumulada en el vértice v_i es:

$$\kappa_{acum}(i) = \sum_{j=1}^{i-1} \kappa(j) \quad (11)$$

es decir, es el ángulo de giro acumulado desde el primer vértice hasta el vértice v_i . Claramente:

$$\kappa_{acum}(n) = 2\pi \quad (12)$$

Sea $\lambda(i)$ la suma de las longitudes de los segmentos del polígono desde v_1 hasta v_{i+1} (tomando los índices módulo n) y L la longitud total del polígono. Entonces: $\lambda(i) = l(i) / L$. La evolución de giro de un polígono con n vértices es la curva lineal por tramos en el plano λ/κ :

$$\beta(i) = \kappa_{acum}(\lambda(i)) \quad (13)$$

A esta función se la llama el patrón de evolución del ángulo de giro (o más simplemente patrón de giro). Esta función caracteriza a la forma del polígono, independientemente de la posición o escala del mismo en el plano. Es decir, es invariante frente a traslaciones, rotaciones y escala. Es un patrón normalizado. Además, brinda información sobre las características geométricas de la curva. Los intervalos de λ donde $\lambda(s)$ es creciente, determinan las partes convexas del polígono y viceversa. Este patrón fue presentado en Kamlofsky y Bergamini (2013b).

El patrón de giro encierra toda la información de la silueta de la forma analizada. Las zonas crecientes del patrón representan las partes cóncavas de la curva, y las decrecientes las zonas convexas. De esta forma, los mínimos relativos del patrón representan puntos de inflexión. Estos puntos son puntos de referencia para segmentar la forma e identificar en el objeto analizado los tramos característicos.

4.3. El sub-proceso del apareo

Se puede lograr la reconstrucción tridimensional de una escena mediante estereoscopia, a partir de dos imágenes obtenidas de la misma, desde dos puntos de origen diferentes, manteniendo la similaridad entre ellas. Hallar los puntos homólogos en dos imágenes similares de una misma escena implica definir pares de correspondencia de puntos entre la primer imagen con la segunda. En esta propuesta, para lograr ello, se analiza la primer imagen, se obtienen los objetos de interés junto con sus patrones descriptores, y los mismos se los busca en la segunda imagen. A esta etapa se la denomina *reconocimiento de objetos*. Luego de reconocidos los objetos, se tratará de presentar una correspondencia entre los puntos de un objeto de la primer imagen con su reconocido en la otra imagen. A esta etapa se la denomina *puesta en correspondencia*.

4.3.2. Reconocimiento de objetos

Se describe al uso del patrón de evolución de ángulo de giro para hallar un objeto en una imagen.

Distancia entre patrones de giro: Sean κ_1 y κ_2 dos patrones de giro. La distancia entre los patrones κ_1 y κ_2 es el valor:

$$D(\kappa_1, \kappa_2) = \int_0^j (\kappa_1(\lambda) - \kappa_2(\lambda))^2 d\lambda \quad (14)$$

Reconocimiento de objetos mediante la distancia entre patrones: Dos representaciones κ_1 y κ_2 de la misma forma (o con gran similaridad), obtenidas de imágenes distintas, posiblemente a distinta escala y orientación, tienen distancia entre ellas pequeña. Análogamente, si dos representaciones son cercanas (en distancia), las curvas que las originan son similares (Kamlofsky y Bergamini, 2013a). Entonces, un objeto de una imagen es similar a otro objeto de otra imagen, independiente de su rotación, ubicación o escala si la distancia entre sus patrones descriptores es menor a un valor umbral predefinido por la experiencia.

Sea $\kappa(\lambda)$ el patrón de giro de una forma S que se quiere identificar. Se realiza por comparación con los patrones conocidos: sean $\alpha_j(\lambda)$ patrones de formas candidatas a empatarse con la forma S . Se identifica a la forma S con aquella cuyo patrón diste de en menos de un valor umbral μ , es decir:

$$D(\kappa, \alpha_j) \leq \mu \quad (15)$$

Dependencia del punto inicial: El algoritmo de obtención de bordes define al punto inicial como el que se encuentra más arriba y más a la izquierda en la imagen. Entonces, el punto inicial de la curva analizada, depende de la orientación del objeto. Dadas dos representaciones mediante el patrón de giro del mismo objeto iniciadas desde distintos vértices, la distribución de la distancia entre ambos patrones será uniforme (es decir, sus representaciones son paralelas), pero no pequeña, impidiendo el cumplimiento de la ecuación (15). Y eso dependerá de la distancia entre los vértices de inicio: si los vértices de inicio coinciden, la distribución de la distancia será uniforme y pequeña.

Si una de las poligonalizaciones posee n vértices y el punto de inicio es desconocido, son necesarias $n-1$ comparaciones hasta hallar una distancia pequeña entre patrones. La determinación del punto de inicio puede lograrse fácilmente si ambos objetos poseen el mismo

ángulo de rotación (condición de roto-rectificación). O bien, si los ángulos son relativamente distintos, detectada la orientación relativa, es simple obtener puntos de inicio correspondientes y lograr un punto homólogo. La detección de la orientación o ángulo de inclinación se puede lograr a partir de identificar al menor rectángulo que encierra a la curva. El ángulo que forman sus lados con el sistema de coordenadas cartesianas determina cuatro diferentes ángulos de inclinación u orientación:

$$\alpha + k \cdot \frac{\pi}{2}, k=0,1,2,3. \quad (16)$$

Existen cuatro (o tres) puntos de cada curva que resultan de intersecarlas con el rectángulo minimal. Y esos puntos son mayormente correspondientes u homólogos en ambas curvas. El algoritmo iniciará el establecimiento del patrón descriptor en solo esos cuatro puntos (o a lo sumo tres puntos) posibles de la curva, independientemente de su orientación (Kamlofsky y Bergamini, 2013b). Esto significa que en lugar de necesitar realizar $n-1$ comparaciones de patrones, es suficiente con solo realizar cuatro (o tres) comparaciones. Y más aún: si se conoce que las orientaciones son similares, las comparaciones se reducen a una sola. Se muestra en la ilustración 5.

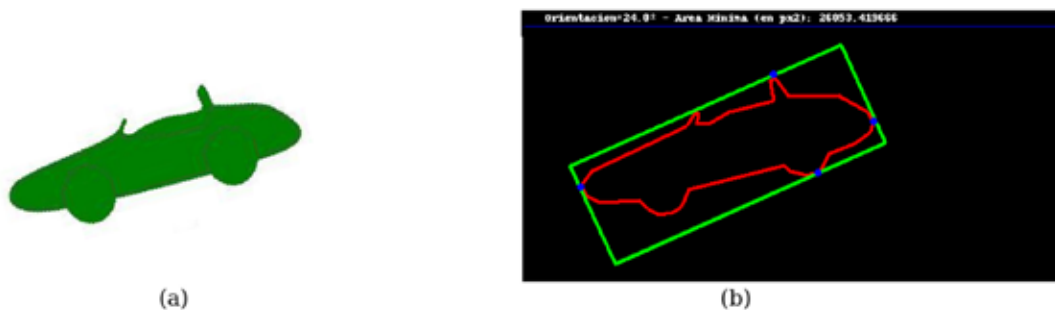


Ilustración 5: La Orientación de un objeto. (a) Objeto analizado. (b) Orientación detectada

En la ilustración 5 (a) se muestra un objeto que se desea hallar en otra imagen. En 5 (b) se muestra un objeto candidato encerrado por el rectángulo minimal. Puede observarse que el rectángulo interseca al borde del objeto en cuatro puntos: los cuatro puntos candidatos a punto de inicio. Además, el lado mayor indica la orientación de la forma tal como aquí se definió: 24°.

El reconocimiento: Luego que el rectángulo minimal haya permitido obtener la orientación de los objetos candidatos, se presentan (a lo sumo) cuatro puntos de inicio candidatos. Con ello, se establecen los patrones descriptores y se calcula las respectivas distancia.

Ejemplo 6: En la ilustración 6 se presentan dos imágenes (a) y (b). En la primera (a) se presenta el objeto que se desea buscar. Puede observarse que en la segunda imagen (b) se encuentra un dibujo similar al objeto de la primer imagen (a), con diferente orientación, ubicación y escala.

La Tabla 1 muestra los valores de distancia de los cuatro patrones de la forma que se desea buscar ilustración 7 (a) con cada uno de los cuatro patrones de las tres formas (b). Se observa que el cuarto patrón del Objeto 2 (Hoja verde) presenta una distancia de 0,364 con el primer patrón del modelo, lo cual se encuentra destacado con un sombreado verde.

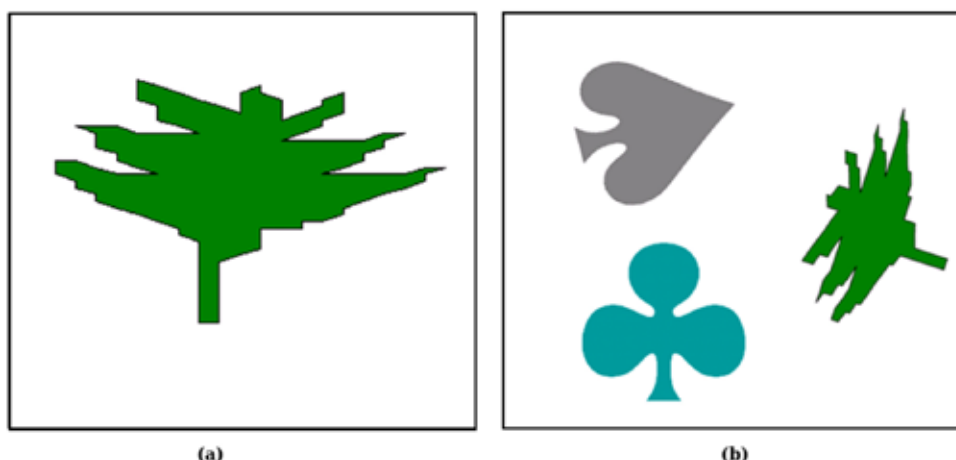


Ilustración 7: (a) Imagen conteniendo el objeto que se desea buscar (Imagen Modelo). (b) Imagen que contiene varios objetos (Imagen de Búsqueda).

4.3.3. Puesta en correspondencia

La puesta en correspondencia de los puntos de un objeto en ambas imágenes consiste hallar pares de puntos donde el primer punto pertenece al borde poligonalizado de primer imagen, el segundo corresponde al primer punto de la segunda imagen. Es decir, dados ciertos puntos del objeto real, la puesta en correspondencia presenta los pares de puntos de ambas imágenes que muestran a los puntos reales del objeto. Si las figuras en ambas imágenes están roto-rectificadas (condición de epipolaridad), la tarea es sencilla. En cambio, si las imágenes no están roto-rectificadas, puede realizarse la puesta en correspondencia entre ambas figuras, si se mantienen las otras condiciones: similaridad y continuidad figural. Durante el reconocimiento de las formas, su roto-rectificación se logra luego de hallar la orientación. En este proceso, el rectángulo mínimo que encierra a la forma en ambas imágenes, se interseca con 4 (o tres) puntos de ella, que son puntos homólogos. Pueden determinarse así, cuatro pares de puntos correspondientes (o tres). Gracias a que las curvas están parametrizadas por longitud de arco, y teniendo en cuenta que las imágenes son similares, pueden obtenerse más puntos homólogos: se recorren los vértices de una de las curvas, se busca su homólogo en la evolución del perímetro normalizado en la segunda curva, dentro de un radio pequeño de tolerancia ρ . Por ejemplo, con tolerancia $\rho = 5\%$ para un vértice, y si un vértice v_i tiene coordenadas en el patrón de la primer imagen $(\lambda_{1i}; \kappa_{1i})$, se buscará un vértice v_j en el patrón de la segunda imagen en $(\lambda_{2j}; \kappa_{2j})$ de modo que: $D((\lambda_{1i}; \kappa_{1i}); (\lambda_{2j}; \kappa_{2j})) < 0,05 = \rho$, con $D(x; y)$: distancia usual o euclídea, y se lo agregará al conjunto de puntos homólogos.

4.3.4. Restricciones para lograr mejoras en el Desempeño

En Kamlofsky et al. (2018) se mostró que contemplando ciertas restricciones, pueden obtenerse notables mejoras en la performance del apareo. La primer restricción se refiere a que la imagen se recibe ya binarizada (blanco y negro). Entonces, como en RGB el gris se representa con el promedio de las tres componentes de la imagen en color en cada una de las componentes, en el proceso de lectura, solo se lee una de las tres componentes RGB de la imagen binarizada. La segunda restricción se refiere al tamaño mínimo de un objeto, definido en 10px de ancho y alto. Es decir, todo elemento de menos de 10px de ancho y 10px de alto no se tendrá en cuenta y será considerado ruido (valor parametrizable). De este modo, la recorrida de la imagen se realizará con dos ciclos for anidados a saltos de 10px tanto en ancho como en

alto, reduciendo al 1% en promedio el proceso de lectura de los bits de la imagen. La tercer restricción establece que en imágenes proveniente de videos, la velocidad del movimiento del objeto es constante. Gracias a ello, se la realizará en solo una porción de la misma. En (Kamlofsky, 2021) se incluyó una nueva mejora: tal como se muestra en la ilustración 20, el procesamiento de lectura de imágenes, binarización, obtención de puntos de borde, poligonalización y obtención de los patrones de giro en cada imagen pueden realizarse en paralelo. Y dado que el equipamiento usado incluye cuatro procesadores, las mencionadas tareas se ejecutan en hilos paralelos, lo cual permitió mejorar aún más la performance del proceso.

Tabla 1: Distancias entre los diferentes patrones de objetos en una imagen con un patrón modelo.

Distancia entre patrones de Objetos		Objeto en Imagen Modelo			
Objeto de la Imagen de Búsqueda		Patrón 1	Patrón 2	Patrón 3	Patrón 4
Objeto 1: Pica gris	Patrón 1	2,297	2,448	2,529	1,993
	Patrón 2	4,994	8,529	2,222	3,893
	Patrón 3	3,263	6,667	2,173	2,699
	Patrón 4	1,752	2,545	2,752	2,244
Objeto 2: Hoja verde	Patrón 1	2,654	0,695	5,485	3,485
	Patrón 2	2,081	5,143	1,959	2,29
	Patrón 3	2,641	3,957	2,63	0,798
	Patrón 4	0,364	2,684	2,3	2,318
Objeto 3: Trébol Turquesa	Patrón 1	4,149	7,297	1,698	3,718
	Patrón 2	3,402	6,155	1,482	3,361
	Patrón 3	1,724	3,107	3,034	1,547
	Patrón 4	1,912	3,587	2,855	1,693

4.4. La complejidad computacional del programa

El cálculo de la complejidad computacional de un algoritmo permite presentar un indicador que resulta adecuado para comparar la performance entre distintos algoritmos. Así, puede compararse cómo un par de algoritmos se comportan frente a un problema (Guillermo, 2018). Se llama $O(f(n))$ al Orden de la complejidad de la función $f(n)$. Se suele identificarse la complejidad de ciertas funciones de referencia. La Tabla 2 muestra a las funciones de referencia con su correspondiente orden de complejidad.

En general, con conocer los órdenes de complejidad de estas funciones es suficiente (Mañas, 2017). Puede evaluarse la calidad de un algoritmo realizando una comparación con otros algoritmos o realizando una comparación de la complejidad del problema con alguna cota de complejidad conocida (Schaeffer, 2020). Con esta idea, se presenta un enfoque empírico simple que permite determinar la complejidad de un programa basado en el trabajo de Mañas (2017). Este enfoque consiste en calcular los tiempos de resolución de un problema con algoritmos cuyo orden de complejidad es conocido, y compararlo con los tiempos de resolución de nuestro programa. Luego, el orden de complejidad de nuestro programa estará determinada por el orden de complejidad que se encuentra entre una función de complejidad mayor y una función de orden menor (basados en la Tabla 2). A este enfoque se lo denomina Análisis asintótico (Schaeffer, 2020).

Tabla 2: Ordenes de complejidad de las funciones de referencia

Función	Orden de complejidad	Notación
$f(n)=1$	$O(f(n))$: Orden Constante	$O(f(n))=O(1)$
$f(n)=\log(n)$	$O(f(n))$: Orden Logarítmico	$O(f(n))=O(\log n)$
$f(n)=n$	$O(f(n))$: Orden Lineal	$O(f(n))=O(n)$
$f(n)=n \cdot \log(n)$	$O(f(n))$: Orden Cuasi-lineal	$O(f(n))=O(n \cdot \log n)$
$f(n)=n^2$	$O(f(n))$: Orden Cuadrático	$O(f(n))=O(n^2)$
$f(n)=n^a$	$O(f(n))$: Orden Polinomial ($a \geq 2$)	$O(f(n))=O(n^a)$
$f(n)=a^n$	$O(f(n))$: Orden Exponencial ($a > 1$)	$O(f(n))=O(a^n)$
$f(n)=n!$	$O(f(n))$: Orden Factorial	$O(f(n))=O(n!)$

5. Implementación de la propuesta

5.1. Presentación de la aplicación experimental

Se obtuvieron dos imágenes de una misma escena con dos cámaras separadas, con orientación similar. Luego, se segmentaron las imágenes con la intención de separar los objetos del fondo. Se les obtuvo su borde, luego sus patrones de giro, se realizó el reconocimiento de un objeto de una imagen en la otra imagen y finalmente, la puesta en correspondencia. La aplicación experimental se realizó siguiendo las tareas presentadas en la propuesta técnica. Los datos experimentales consisten en: mediciones de acierto en el apareo en ambos objetos para distintos valores de ρ . Se presentan mediciones que permiten clasificar a la solución según su complejidad computacional.

5.2. Herramientas y equipamiento usado

5.2.1. Hardware y Software

Para la adquisición de las imágenes se usaron dos cámaras USB marca Logitech modelo C170¹² separadas 230mm una de otra y apuntando ambas hacia adelante. Las imágenes obtenidas son de 640x480 píxeles. El computador usado es una notebook HP Pavillion TS-15 con pantalla táctil de 15 pulgadas, 4 cores AMD A10-5745m, 12GB de memoria RAM. El sistema operativo instalado en el computador es Ubuntu 20.04 LTS, una distribución con núcleo Linux de 2020. Los algoritmos se programaron en Python¹³ 2.7. Las principales librerías usadas son: sys y os para manejo de archivos, numpy y math para formatos numéricos y funciones matemáticas, PIL y OpenCV para el manejo de imágenes y matplotlib para la realización de gráficos.

5.2.2. Características de las Imágenes y parámetros usados

Las imágenes: Se trabajó sobre dos pares de imágenes estéreo de 640 x 480 píxeles: un Martillo y un zapato, colocados sobre una silla, con fondo blanco, importante, para simplificar la segmentación (el problema de la segmentación no forma parte de este trabajo).

¹² Hoja de datos disponible en: <http://www.cartimex.com/v2/pdf/960-000880.pdf> Fecha de Consulta: 1/8/19

¹³ Sitio Oficial de Python: <https://www.python.org/>

Parámetros usados: ancho de la cáscara convexa = 4 (píxeles), la tolerancia en el hallazgo de puntos homólogos ρ se la seteó entre 1% y 5% a intervalos del 1% y ventanas de búsqueda iguales de 320 x 240 píxeles en ambos pares de imágenes estéreo (o bien: “pares estéreo”).

5.3. Resultados obtenidos

5.3.1. Proceso de interpretación de imágenes

Lectura y Binarización. Cada una de las imágenes posee 307200 píxeles. Para el par estéreo del martillo, el tamaño del mismo es de 8165 en la imagen izquierda y 8403 en la imagen derecha. Mientras que para el par estéreo del zapato, el zapato ocupa 16838 píxeles en la imagen izquierda y 17346 píxeles en la imagen derecha.

Puntos de borde. Para el par estéreo martillo, el objeto martillo de la imagen izquierda posee 792 puntos de borde mientras que el de la imagen derecha posee 790 píxeles. Mientras que para el par estéreo del zapato, el objeto zapato de la imagen izquierda posee 636 puntos de borde mientras que el de la imagen derecha posee 648 píxeles.

Poligonalización: Para el par estéreo del martillo, el martillo de la imagen izquierda poligonalizado posee 16 vértices mientras que el de la imagen derecha posee 15 vértices. Mientras que para el par estéreo del zapato, el objeto zapato de poligonalizado posee 14 vértices en ambas imágenes.

Patrón de giro: En la ilustración 8 se muestra una gráfica donde se presentan los dos patrones.

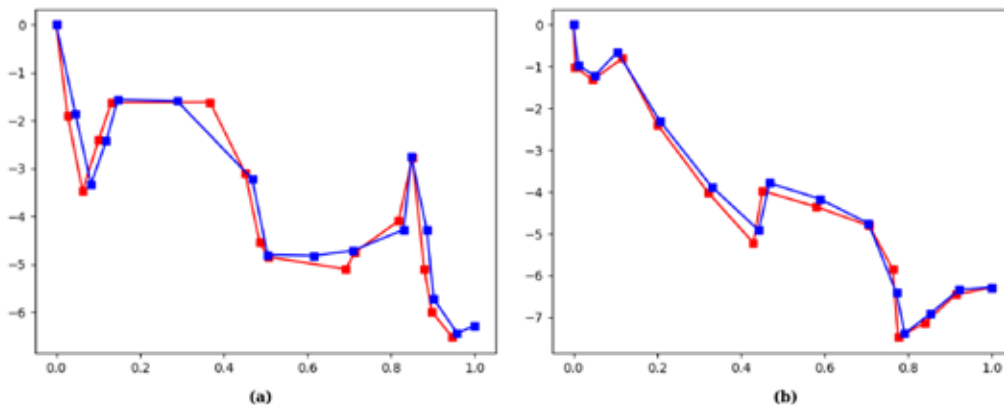


Ilustración 8 Comparación entre los patrones de dos pares estéreo. (a) Martillo, (b) zapato.

5.3.2. Apareo

Reconocimiento de objetos: Se define el valor del umbral: $\mu=0,1000$. Para el par estéreo de martillo, la distancia entre los patrones: $D1 = 0,044$, con $D1 < \mu$, mientras que para el par estéreo zapato, la distancia entre los patrones: $D2 = 0,080$, con $D2 < \mu$. Ello significa que ambos objeto presentes en las imágenes de la derecha fueron hallado en las imágenes de la izquierda.

Puesta en correspondencia: Tras el reconocimiento de un objeto, el punto de inicio de ambos patrones queda definido. El resto de los puntos pueden hallarse recorriendo cada punto del

perímetro de uno de los objetos, hallando la distancia entre los valores del patrón en ambas imágenes en ese punto, y comparándolo con un valor ρ . Se recopilaban los resultados obtenidos.

5.3.3. Análisis de confianza del proceso

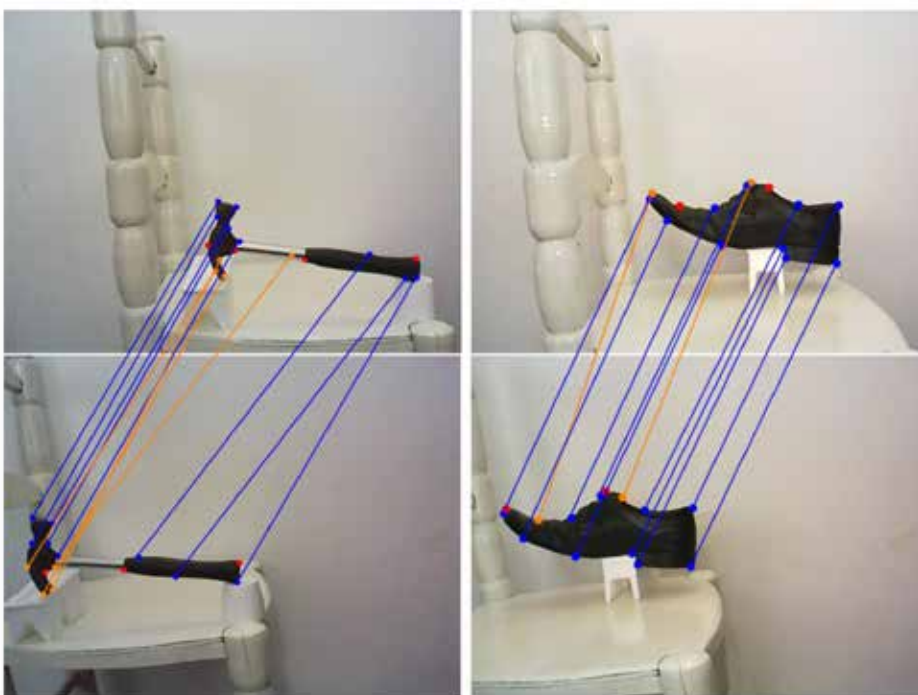
En la ilustración 9 se presenta una puesta en correspondencia entre pares estereos de imágenes de los objetos: martillo y zapato. Con un pequeño círculo de color se representa a cada uno de los vértices obtenidos en el proceso de poligonalización. Si los vértices que fueron apareados correctamente, los círculos se pintaron en azul y se unieron con una línea continua del mismo color. Si los vértices que fueron apareados incorrectamente se pintaron de amarillo y se unieron con una línea amarilla. Los vértices que no fueron apareados, se pintaron de rojo. Esto se realizó para ambos pares estereos para valores enteros desde $\rho = 1$ hasta $\rho = 5$. Con ello, se elaboró la Tabla 3 que contiene datos experimentales que permiten analizar la confiabilidad del proceso.

Se presentan dos indicadores que pretenden describir la confiabilidad del proceso: el Índice de apareos y el índice de confianza de los apareos. El índice de apareos (IA) indica la proporción de los vértices del objeto poligonalizado que son apareados por el algoritmo.

$$I_A = \frac{\text{cantidad de vértices apareados}}{\text{cantidad de vértices}} \quad (17)$$

El índice de confianza de los apareos (CA) se presenta como un indicador que refleja la bondad en el proceso de apareo. Del conjunto de vértices que han sido apareados, CA indica la proporción de ellos que resultaron correctos tras la comprobación visual.

$$C_A = \frac{\text{cantidad de vértices apareados}}{\text{cantidad de vértices apareados correctamente}} \quad (18)$$



En el total del conjunto de datos, el IA no presenta datos homogéneos, por lo que su promedio no es un indicador de tendencia central adecuado. Presenta valores bajos para ρ pequeños. Pero a partir de $\rho = 0,03$ sus valores mantienen homogeneidad (con coeficiente de variación de Pearson de 0,04) alrededor del promedio cuyo valor es del 84%. Esto es: a partir de $\rho = 0,03$ el algoritmo logra aparear (en promedio) el 84% de los vértices. La confianza en los apareos (CA) en cambio, presenta datos homogéneos (con coeficiente de variación de Pearson de 0,10). Por ello, puede decirse que el promedio de la confianza en los apareos (CA) es un indicador de tendencia central adecuado.

Conclusión: Para $\rho > 0,02$ el algoritmo logra aparear correctamente (en promedio) aproximadamente el 70% de todos los vértices del polígono que simplifican a las curvas de bordes de los objetos presentes en ambas imágenes estéreo.

Tabla 3: Análisis de la confiabilidad del proceso de apareo

Objeto	ρ	Vértices totales	Vértices apareados	Apareos Correctos	Índice de Apareos	Confianza en Apareos
Zapato	0,01	14	5	5	0,36	1,00
Martillo	0,01	15	6	5	0,40	0,83
Zapato	0,02	14	12	10	0,86	0,83
Martillo	0,02	15	9	7	0,60	0,78
Zapato	0,03	14	12	10	0,86	0,83
Martillo	0,03	15	13	12	0,87	0,92
Zapato	0,04	14	12	10	0,86	0,83
Martillo	0,04	15	12	10	0,80	0,83
Zapato	0,05	14	12	10	0,86	0,83
Martillo	0,05	15	12	8	0,80	0,67
Promedio					0,73	0,84
Desvío Estándar					0,20	0,09
Coef. Var. de Pearson					0,27	0,10

5.3.4. Análisis del desempeño

Se analizan aquí, tiempos de cómputo del algoritmo. Se realizaron 5 mediciones por cada par de imágenes estéreo del tiempo para la obtención de puntos homólogos. Cada experimento se lo subdividió en las etapas y se cronometró cada tarea. La Tabla 4 presenta el análisis de las mediciones tomadas. Se tomó nota de los tiempos para la realización de cada una de las tareas.

Tabla 4 Tiempos de cómputo del proceso

Número de Prueba	Objeto	Tiempos de Cómputo (en segundos)										Tiempo Total
		Segmentación	Obtención de Bordes	Poligonización	Cálculo de Patrón	Segmentación	Obtención de Bordes	Poligonización	Cálculo de Patrón	Reconocimiento	Puesta en Correspond.	
1	Martillo	0,191	0,074	0,083	0,134	0,169	0,069	0,051	0,153	0,047	0,039	0,570
2	Martillo	0,197	0,072	0,081	0,134	0,149	0,071	0,050	0,149	0,038	0,036	0,558
3	Martillo	0,183	0,070	0,083	0,125	0,157	0,071	0,050	0,150	0,039	0,031	0,533
4	Martillo	0,179	0,071	0,086	0,125	0,161	0,070	0,048	0,096	0,038	0,031	0,530
5	Martillo	0,187	0,076	0,086	0,126	0,151	0,069	0,050	0,150	0,042	0,033	0,530
6	Zapato	0,233	0,088	0,067	0,101	0,232	0,068	0,068	0,088	0,043	0,044	0,580
7	Zapato	0,268	0,069	0,063	0,103	0,217	0,079	0,071	0,082	0,043	0,045	0,593
8	Zapato	0,285	0,065	0,067	0,101	0,218	0,077	0,072	0,083	0,046	0,044	0,608
9	Zapato	0,143	0,076	0,069	0,104	0,221	0,071	0,069	0,086	0,046	0,045	0,538
10	Zapato	0,234	0,068	0,068	0,100	0,223	0,068	0,079	0,078	0,043	0,046	0,579
Promedio:		0,212	0,073	0,076	0,115	0,191	0,071	0,061	0,112	0,043	0,039	0,564
Desvío estándar:		0,046	0,006	0,009	0,015	0,036	0,004	0,012	0,034	0,003	0,006	0,026
Coef. Var. Pearson		0,215	0,087	0,123	0,127	0,190	0,052	0,197	0,304	0,080	0,157	0,047

El proceso global presenta homogeneidad de datos (coeficiente de variación de Pearson $CV = 0,047$), por lo que puede considerarse a su promedio como un indicador de tendencia central adecuado.

Conclusión: El algoritmo logra la obtención de puntos homólogos en un par de imágenes estéreo de un objeto en un promedio de 0,564 segundos en las condiciones presentadas.

5.3.5. Limitaciones de la esta implementación

El análisis de desempeño o performance anteriormente presentado podría mostrar resultados notoriamente mejores sin alguna de las limitaciones que se presentan a continuación. La primera de ellas se refiere al lenguaje de prototipado usado: Python es un lenguaje que resulta indefectiblemente más lento que C, aunque con este lenguaje resulta fácil programar e implementar prototipos. La segunda se refiere a que la programación de los algoritmos no ha sido optimizada. Entonces, la implementación de cualquier técnica de optimización mejoraría su tiempo de cómputo. La tercera hace referencia a que alguna de las tareas como ser la poligonalización o el reconocimiento podrían realizarse de una manera aún más eficiente. Y la cuarta limitación, a que la puesta en correspondencia se realiza sobre puntos de borde: si se analizaran los esqueletos, quizás los resultados fueran mejores.

5.4. Análisis asintótico de la complejidad computacional

Para realizar empíricamente el cálculo del orden de complejidad de la solución aquí presentada (análisis asintótico), se necesita calcular los tiempos de cómputo para funciones de referencia y compararlos con los tiempos de cómputo del programa que ejecuta la solución presentada. Los programas de las funciones de referencia se obtuvieron de una consulta en el sitio Stack Overflow en Español (Abulafia, 2020), aquí llamados EM1 y EM2 con complejidad lineal y cuadrática respectivamente, apropiados para realizar el análisis asintótico de complejidad (Schaeffer, 2020). El programa que ejecuta la solución aquí presentada es llamado P. Su orden de complejidad es $O(P(n))$.

5.4.1. Datos experimentales

Respecto del programa P, su variable de análisis es n : cantidad de píxeles que tiene la imagen a analizar. La experiencia se realizó con la imagen “Zapato”: se le realizaron escalados varios para presentar un conjunto de imágenes a escala (cuasi) lineal de valores de n . Se varía, el valor de la tolerancia en la poligonalización ε : Para los valores de $\varepsilon=1$, $\varepsilon=3$, $\varepsilon=5$, llamaremos a las variantes del programa: $PZ1(n)$, $PZ3(n)$ y $PZ5(n)$. La Tabla 5 muestra los datos recopilados.

La Ilustración 10 presenta la comparación entre los tiempos de procesamiento de las funciones $PZ1(n)$, $PZ3(n)$ y $PZ5(n)$ en comparación con funciones de referencia $EM1(n)$ y $EM2(n)$. En la Ilustración 40 puede observarse que para todos los valores de n analizados, las funciones $PZ1(n)$, $PZ3(n)$ y $PZ5(n)$ están acotadas por las funciones $EM1(n)$ y $EM2(n)$.

5.4.2. La complejidad computacional de la solución

Al analizar la Ilustración 10 y los datos de la Tabla 5 puede observarse que:

$$\begin{cases} \forall n \in \mathbb{N}: EM2(n) > PZ1(n) > EM1(n) \Rightarrow O(PZ1(n)) = n \log n \\ \forall n \in \mathbb{N}: EM2(n) > PZ3(n) > EM1(n) \Rightarrow O(PZ3(n)) = n \log n \\ \forall n \in \mathbb{N}: EM2(n) > PZ5(n) > EM1(n) \Rightarrow O(PZ5(n)) = n \log n \end{cases}$$

Conclusión: la complejidad computacional de la solución presentada es cuasi-lineal.

Tabla 5 Tiempos de ejecución de las diferentes funciones en función de n .

Tamaño Imagen	n (píxeles)	Tiempos de procesamiento (en s)				
		EM2(n)	EZ1(n)	EZ3(n)	EZ5(n)	EM1(n)
106x80	8480	4,462	0,044	0,027	0,026	0,009
144x108	15552	14,406	0,085	0,042	0,041	0,019
176x132	23232	31,901	0,070	0,046	0,044	0,031
200x150	30000	53,911	0,066	0,051	0,057	0,033
228x171	38988	—	0,065	0,062	0,059	0,039
250x187	46750	—	0,083	0,071	0,069	0,048
266x202	53732	—	0,082	0,075	0,074	0,059
288x216	62208	—	0,088	0,087	0,087	0,062
304x228	69312	—	0,094	0,088	0,900	0,072
320x240	76800	—	0,125	0,100	0,103	0,079

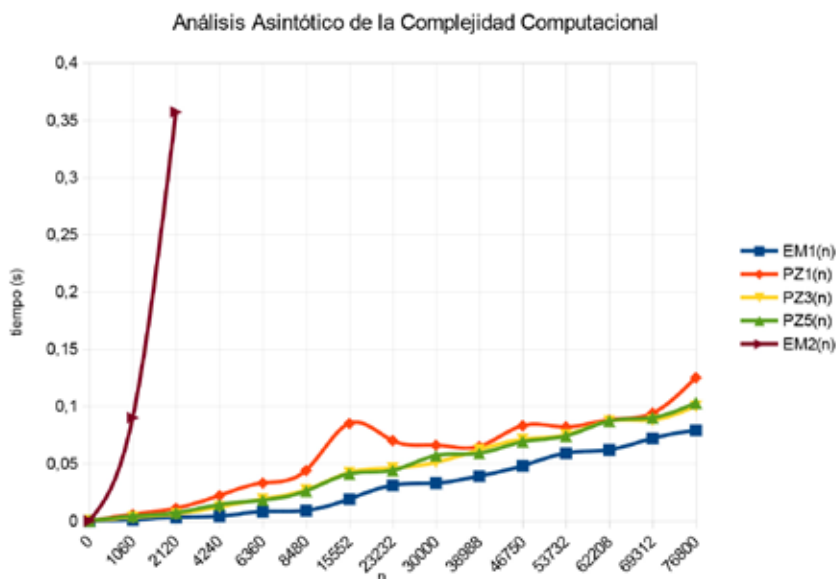


Ilustración 10: Comparación de los tiempos de ejecución de las diferentes funciones.

Nota: Si bien el análisis asintótico muestra que la complejidad de la solución propuesta es “cuasi-lineal”, es fácil observar que las funciones se encuentran mucho más próximas a la función de referencia de complejidad lineal que a la cuadrática.

5.5. Los resultados de la experiencia

Los resultados aquí presentados y en las condiciones de esta experiencia (lograr el apareo en 0,564 segundos) permitirían lograr el seguimiento de un objeto en tiempo real. Aún así, estos resultados podrían mejorarse si se llegara a implementar alguna mejora que evite las limitaciones en el desempeño o performance presentadas. Se logró la rápida puesta en correspondencia de

algunos pocos puntos del objeto, lo que permite la reconstrucción tridimensional a partir de los bordes. Y al poseer coordenadas tridimensionales (en unidades de longitud), es más sencillo lograr el reconocimiento de algunos objetos mediante abstracciones en unidades espaciales. Por ejemplo, en una aplicación que busca personas vivas, se reconoce una forma humana erguida obtenida mediante el patrón de giro que posee 37°C (segmentado mediante una cámara térmica) y una altura de 1,70m (esto logrado tras la reconstrucción tridimensional en unidades de longitud): podría afirmarse que es un ser humano. Este tipo de abstracciones de nivel superior pueden lograrse muy rápidamente con un estudio profundo del dominio del problema y puede ser de gran utilidad por ejemplo en implementaciones como ser: robótica para búsqueda y rescate en áreas de catástrofe (Greer et al., 2002; Naidoo et al., 2015; Murphy & Stover, 2008; Kamlofsky et al., 2018).

6. Conclusion

En este trabajo se presentó un algoritmo que logra la puesta en correspondencia de objetos en un par de imágenes estéreo de una misma escena rápidamente, lo cual puede ser compatible con requerimientos de tiempo real: se logra la obtención de puntos homólogos en aproximadamente medio segundo, en un prototipo sin optimizar en las condiciones de esta experiencia. El cálculo de la complejidad computacional de esta solución permite suponer que es posible generalizar la afirmación anterior independizándola de las condiciones de esta experiencia. Se presentó un adecuado detalle técnico del algoritmo, junto con datos experimentales e indicadores que permiten medir su alcance.

Una conclusión subyacente apunta a presentar al enfoque topológico de análisis de bordes como un enfoque simple, sencillo y muy eficiente, para la identificación de objetos en imágenes digitales y así usarse en soluciones de visión robótica.

Trabajos Futuros

El esquema presentado en este trabajo posee aspectos muy novedosos en el marco de la Visión 3D, orientado a su procesamiento en tiempo real, el cual es un tema abierto en el estado del arte actual. En el capítulo 5 se presentó la implementación de la solución, con sus ventajas y limitaciones. Entre estas últimas, se presentan limitaciones en la performance. Un primer trabajo futuro puede tratar acerca de solucionar las limitaciones aquí presentadas. Otra limitación presentada se refiere a que la obtención de puntos homólogos para modelización tridimensional se realiza sobre los puntos de borde de los objetos. En el trabajo presentado por Rosenfeld (1979), se presentó el algoritmo BF4/8 para obtener los bordes de un objeto. Sin embargo, en el mismo trabajo, Rosenfeld también se presentó un algoritmo para afinamiento de formas a sus esqueletos. Con ello también se permitiría obtener la reconstrucción tridimensional aproximada de los puntos del esqueleto. Otro trabajo futuro podría ser, entonces, obtener un apareo rápido a partir de puntos de los esqueletos o también, la combinación de bordes y esqueletos.

Referencias

- » Abulafia, (2020). "Python – cuál es la complejidad de mi algoritmo". Stack Overflow en Español. [En línea]: <https://es.stackoverflow.com/questions/258283/cu%C3%A1l-es-la-complejidad-algor%C3%Admica-de-este-programa>. Consultado el 25/02/2021.
- » Bergamini María Lorena, Kamlofsky Jorge. "Aproximación Poligonal de Curvas Digitales". *Matemática Aplicada Computacional e Industrial* Vol. 6 (2017): 545-548.
- » Bergamini, María L. & Jorge A. Kamlofsky. "Representación de Formas Digitales para Reconocimiento y Clasificación de Objetos." *Revista Colombiana de Computación* 16 (2015): 28-47.
- » Bergamini M., Ansaldo F., Bright G., & Zelasco J. Fundamental Matrix: Digital Camera calibration and Essential Matrix parameters. *International Journal of Signal Processing*, 1 (2016)(a): 120-126.
- » Bergamini M, Ansaldo F, Bright G & Zelasco J. "Fundamental Matrix: Digital camera calibration and Essential Matrix parameters". 16th International Conference on Signal Processing, Computational Geometry and Artificial Vision 2016 proceedings (ISCGAV), (2016)(b).
- » Burgard et al.. "Experiences with an interactive museum tour-guide robot." *Artificial intelligence* 114.1-2 (1999): 3-55.
- » Davis, Larry S. "A survey of edge detection techniques." *Computer graphics and image processing* 4.3 (1975): 248-270.
- » De Berg M et al.. "Computational Geometry", Springer-Verlag Berlin Heidelberg New York (1997).
- » Donadio, A., Mendez, D. "Vision 3D. Tratamiento de imágenes estereoscópicas. Restitución Automática. Evaluación de resultados. Universidad Nacional del Centro de la Provincia de Buenos Aires, Facultad de Ciencias Exactas, 1997
- » Eckhardt, Ulrich; Longin Latecki. "Digital topology". *Inst. für Angewandte Mathematik* (1994).
- » Egmont-Petersen, Michael, Dick de Ridder, & Heinz Handels. "Image processing with neural networks—a review." *Pattern recognition* 35.10 (2002): 2279-2301.
- » García, Pedro Pablo García. "Reconocimiento de imágenes utilizando redes neuronales artificiales." *Facultad de Informatica, Universidad Complutense de Madrid* (2013).
- » Gil, Laurent and Liska, Allan. "Security with AI and Machine Learning." O'Reilly Media, (2019).
- » Guillermo, J. (2018) "¿Qué es la complejidad computacional y cómo se come?". Sitio: Medium.com. [En línea]: https://medium.com/@joseguillermo_/qu%C3%A9-es-la-complejidad-algor%C3%Admica-y-con-qu%C3%A9-se-come-2638e7fd9e8c. Fecha de consulta: 24/2/2021
- » Gonzalez, Rafael C, Woods, Richard E. "Digital image processing." Addison-Wesley Publishing Company (1993).
- » Greer D., McKerrow P., & Abrantes J., "Robots in Urban Search and Rescue Operations", © ARAA, 2002 Australasian Conference on Robotics and Automation, November 2002.
- » Gross, Ari; Latecki, Longin. "Digital Geometric Invariance and Shape Representation". 0-8186-7190-4/95, IEEE, (1995): 121-126.
- » Hernández, R., Fernández, C. y Baptista, P. "Metodología de la investigación (6ª ed.)". México: McGraw Hill Education, (2014).
- » Hernández Orallo, J; Ramírez Quintana, M, y Ferri Ramirez, C. "Introducción a la Minería de Datos". Editorial Pearson Educación SA, Madrid, (2004).
- » Hirschmüller, Heiko. "Accurate and efficient stereo processing by semi-global matching and mutual information.", IEEE, 2005.
- » Kalantary M. & Jung F, "Estimation Automatique de l'Orientation Relative en Imagerie Terrestre.", XYZ-

AFT, 114, (2008): 27-31.

- » Kamlofsky, Jorge. "Topología digital, base para la visión artificial". Tesis de Licenciatura, Universidad Abierta Interamericana, Buenos Aires, 2011
- » Kamlofsky J., Bergamini, M.: Aproximación de Formas de Objetos Digitales por Polígonos. VII Jornadas Argentinas de Robótica ISBN: 978-950-658-316-3, (2012).
- » Kamlofsky, Jorge Alejandro & Bergamini, María Lorena. "Patrón de Evolución Discreta de Curvatura y Concavidad para Reconocimiento de Formas." CONAISI, (2013)(b).
- » Kamlofsky, Jorge, & María Lorena Bergamini. "Patrón de Ángulo de Giro para Reconocimiento de Objetos en Imágenes Digitales." 14th Argentine Symposium on Technology, ISSN 1850-2806, (2013) (a): 115-127.
- » Kamlofsky, Jorge y Bergamini, María Lorena. "Herramientas de Análisis de Bordes para Reconocimiento y Clasificación de Objetos en Imágenes Digitales." II Congreso Nacional de Ingeniería en Sistemas de Información (Conaisi2014), 2014(a).
- » Kamlofsky, Jorge y Bergamini, María Lorena. "Un Modelo para Soporte de la Investigación Asistido por Técnicas de Visión Artificial." II Congreso Nacional de Ingeniería en Sistemas de Información (Conaisi2014), 2014(b).
- » Kamlofsky Jorge y Bergamini María Lorena. "Los Cuaterniones en Visión Robótica". Matemática Aplicada Computacional e Industrial Vol. 5 (2015): 517-520.
- » Kamlofsky. J y Bergamini, M. "Rápida Obtención de Puntos Homólogos para Vision 3D". VI Congreso de Matemática Aplicada, Computacional e Industrial, Vol. 6 (2017): 549-552.
- » Kamlofsky J., Bergamini, M. Aproximación de Formas de Objetos Digitales por Polígonos. VII Jornadas Argentinas de Robótica ISBN: 978-950-658-316-3, (2012).
- » Kamlofsky, J. Naidoo, N., Bright, G., Stopforth, R., Zelasco, J., Ansaldo, F., & Bergamini, M. "Semi-Autonomous Robot Control System with an improved 3D Vision Scheme for Search and Rescue Missions. A joint research collaboration between South Africa and Argentina." Advanced in Sciences, Technology and Engineering System Journal (Astes Journal) Vol 3 (6), 347-357 DOI: 10.25046/aj030643 (2018).
- » Kamlofsky Jorge. "Rápida obtención de puntos homólogos para visión robótica". XIX Congreso Internacional de Innovación en Tecnología Informática, 2021.
- » Kong, Yung; Kopperman, Ralph; Meyer, Paul. "A Topological Approach to Digital Topology". The American Mathematical Monthly Vol. 98, (1991): 901-917.
- » Malpartida, Eddie Angel Sobrado. "Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot." Pontificia universidad católica del Perú, Perú (2003).
- » Mañas, José. "Análisis de algoritmos – Complejidad". Universidad Politécnica de Madrid, Departamento de Ingeniería de Sistemas Telemáticos, (2017).
- » Monar, Monar, & Willan Leopoldo. Aplicación de las Redes Neuronales al Reconocimiento de Objetos en Robots Manipuladores. MS thesis. Quito, 2014.
- » Moravec, Hans P. "Robot spatial perception by stereoscopic vision and 3d evidence grids." Perception (1996).
- » Murphy R & Stover S. "Rescue Robots for Mudslides: A descriptive study of the 2005 La Conchita Mudslide Response", International Journal of Field Robotics, Wiley, 2008.
- » Naidoo, N., Bright, G., Stopforth, R., Zelasco, J., Ansaldo, F., Bergamini, M., & Kamlofsky, J. (2017, November). "Semi-autonomous robot control system and with 3D vision scheme for search and rescue missions: A joint research collaboration between South Africa and Argentina." In Mechatronics and Machine Vision in Practice (M2VIP), 24th International Conference on (pp. 1-6). IEEE, 2017.

- » Naidoo et al.. "Optimizing search and rescue missions through a cooperative mobile robot network." Pattern Recognition Association of South Africa and Robotics and Mechatronics International Conference (PRASA-RobMech), IEEE, 2015.
- » Nguyen T.P., Debled-Rennesson I.: Curvature Estimation in Noisy Curves. In: Kropatsch W., Kampel M., Hanbury A. (eds.) Computer Analysis of Images and Patterns, LNCS, vol. 4673, pp 474-481, Springer, Heidelberg (2007).
- » Peña-Cabrera, Mario, et al.. "Un Proceso de Aprendizaje para Reconocimiento de Objetos en Línea en Tareas Robotizadas." 3ª Conferencia Iberoamericana en Sistemas, Cibernética e Informática (CISCI 2004). 2004.
- » Rosenfeld, A. "Digital Topology", The American Mathematical Monthly, 86(8), (1979) pp. 621-630.
- » Rosenfeld, A, & Kak, A. "Digital picture processing". Vol. 1. Elsevier, (2014).
- » Rowley, Henry A., Shumeet Baluja, & Takeo Kanade. "Neural network-based face detection."IEEE Transactions on pattern analysis and machine intelligence 20.1 (1998): 23-38.
- » Rudin, Leonid I., Stanley Osher, & Emad Fatemi. "Nonlinear total variation based noise removal algorithms."Physica D: Nonlinear Phenomena 60.1 (1992): 259-268.
- » Sarría, Alfonso Fernández. Estudio de técnicas basadas en la transformada Wavelet y optimización de sus parámetros para la clasificación por texturas de imágenes digitales. Diss. Universitat Politècnica de València, (2007).
- » Schaeffer, E., (2020). "Complejidad computacional de problemas y el análisis y diseño de algoritmos". [En línea]: <https://elisa.dyndns-web.com/teaching/aa/pdf/aa.pdf>. Consultado el 25/02/2021
- » Velogig. Machine Learning: Algoritmos más utilizados en marketing digital (2020). En línea: <https://velogig.com/10-algoritmos-mas-usados-en-machine-learning-que-debes-conocer/> Consultado el 03/08/2020.
- » Weaver, J, et al.. "Filtering noise from images with wavelet transforms."Magnetic Resonance in Medicine 21.2 (1991).
- » Zelasco, Jose F., et al.. "Computer vision in AUVs: automatic roto-rectification of stereo images."OCEANS 2000 MTS/IEEE Conference and Exhibition. Vol. 3. IEEE, (2000).
- » Zhang D., Lu G. "Review of shape representation and description techniques". Patt. Recogn., 37 (1), (2004): 1-19.
- » Zhang, Q., Yang, L. T., Chen, Z., & Li, P. "A survey on deep learning for big data". Information Fusion, 42, (2018)