

ROBOT MÓVIL MECX1 PARA LA DETECCIÓN DE PERSONAS EMPLEANDO MEMORIAS ASOCIATIVAS ALFA-BETA

Viridiana G. Hernández Herrera

Instituto Politécnico Nacional/Centro de Investigación e Innovación Tecnológica
vhernandezhe@ipn.mx

Moisés V. Márquez Olivera

Instituto Politécnico Nacional/Centro de Investigación e Innovación Tecnológica
mvmarquez@ipn.mx

Antonio Gustavo Juárez Gracia

Instituto Politécnico Nacional/Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada
cicata81@yahoo.es

Octavio Sánchez García

Instituto Politécnico Nacional/Centro de Investigación en Ciencia Aplicada y Tecnología Avanzada
osanchez0112@gmail.com

Resumen

En presente trabajo de investigación propone emplear a las Memorias Asociativas Alpha-Beta ($AM\alpha\beta$) en la detección automática del cuerpo humano a partir de imágenes RGB-3D capturadas por el robot MECX1, las $AM\alpha\beta$ son entrenadas con vectores característicos extraídos de dos tipos de imágenes, las imágenes positivas contienen personas bajo diferentes poses, distancias e iluminación, mientras que las imágenes negativas contienen objetos que el robot puede encontrar en su entorno de navegación. El rendimiento de las $AM\alpha\beta$ es evaluado en dos pruebas, en la primera se determina la capacidad para recordar los vectores previamente aprendidos, los resultados muestran que la memoria fue capaz de recordar al 100% las formas de cuerpos humanos, así como de los

objetos con los que fue entrenada, en la segunda prueba se evalúa su capacidad para clasificar vectores que no aprendió anteriormente, obteniéndose una tasa de precisión promedio de 95.1%. Para la validación de los resultados y separación de los conjuntos de entrenamiento y prueba se empleó el método de K-fold-cross-validation.

Palabra(s) Clave: cuerpo humano, detección, forma humana, memorias asociativas, reconocimiento.

Abstract

In the present work, we propose to use Alpha-Beta Associative Memories (AM $\alpha\beta$) in the automatic detection of the human body from RGB-3D images captured by the robot MECX1, the AM $\alpha\beta$ are trained with characteristic vectors extracted from two types of images, positive images contain people under different poses, distances and illumination, while negative images contain objects that the robot can find in its navigation environment. The performance of the AM $\alpha\beta$ is evaluated in two tests, the first one determines the ability to remember previously learned vectors, the results show that memory was able to remember 100% human body forms as well as objects with the ones that were trained, in the second test we evaluated the memory capacity to classify vectors that were not previously learned, obtaining an average accuracy rate of 95.1%, K-fold-cross-validation method was used for the validation of the results and separation of the training and test sets.

Keywords: associative memories, detection, human body, human shape.

1. Introducción

Los continuos avances tecnológicos en electrónica y cómputo han permitido que campos de investigación como la robótica móvil presenten altas expectativas en aplicaciones reales, razón por la cual se ha incrementado el interés de los estudios enfocados en el logro de la autonomía de robots móviles, en donde el robot debe ser capaz de tomar decisiones adecuadamente en respuesta a las diferentes circunstancias que se presenten mientras navega dentro de su entorno.

Como se puede observar en la figura 1 los problemas que enfrenta la robótica móvil son diversos, por lo que es indispensable dividirlos para brindar soluciones eficientes que en conjunto den como resultado un robot capaz de realizar tareas específicas.

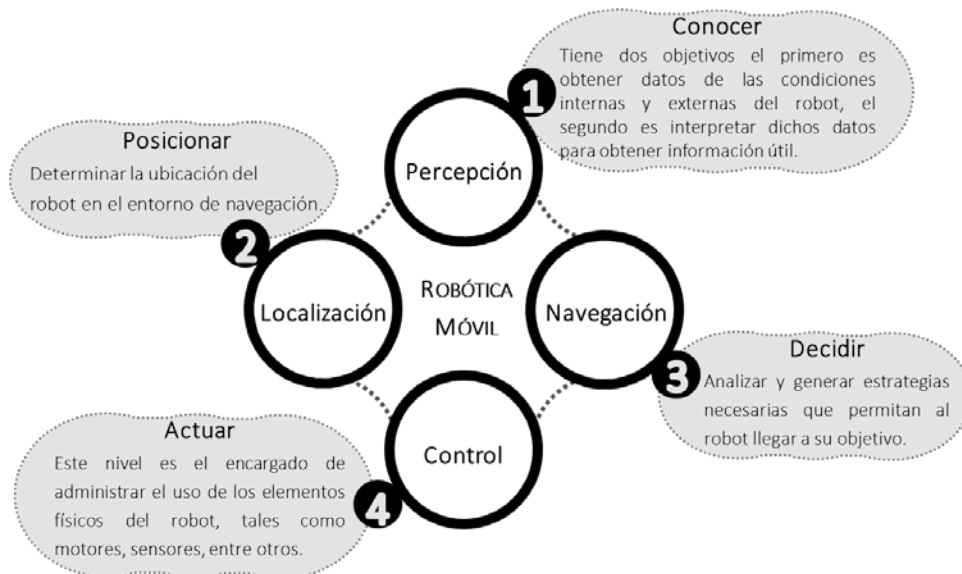


Figura 1 Principales problemas de la robótica móvil.

En un futuro cercano se espera que una gran variedad de robots móviles llamados de servicio brinden su apoyo al ser humano en la realización de quehaceres en el hogar, por lo que para el cumplimiento de dichas tareas es indispensable la interacción hombre-robot, lo cual implica que uno de los primeros problemas a los que deben enfrentarse esta clase de robots es distinguir entre un humano y todo aquello que no lo es.

En la figura 1 se presentan cuatro problemas generales localización, navegación, control y percepción, siendo este último el encargado de obtener datos y convertirlos en información útil para la toma de decisiones del robot, por tanto la eficiente detección del cuerpo humano representa uno de los retos enmarcado en el problema de percepción, el cual debe ser resuelto para lograr la interacción del robot con los seres humanos.

Duan et al. [1] Proponen etiquetar automáticamente a las personas en una secuencia de imágenes capturadas con una cámara fija, para ello se realiza una

fusión de información proveniente de los acelerómetros integrados en un teléfono inteligente portado por la persona a detectar, el objetivo es comparar una secuencia de características de movimiento recogidas de los sensores y cuando se detecta una fuerte correlación la información es enviada a la cámara para enmarcar el cuerpo humano. Lillo et al. [2] Emplean un nuevo modelo composicional jerárquico y videos RGB-D capturados con un sensor Microsoft Kinect para el reconocimiento del cuerpo humano bajo diferentes poses generadas cuando se realizan actividades cotidianas. Jalal et al. [3] también hace uso de del sensor Kinect, con la diferencia de que proponen un sistema para reconocer la actividad que un cuerpo humano está realizando, para ello se procesa la profundidad secuencias de vídeo fusionando las propiedades espaciotemporales, las características invariantes y la normalización del tamaño humano.

Otro de los trabajos basados en información extraídos por Kinect es el de Wang et al. [4] que proponen un nuevo descriptor de características local basado en información de energía para el reconocimiento de actividad humana, no obstante en lugar de detectar los puntos de interés espacio-temporales, combinan la energía cinética y potencial gestual de las articulaciones del esqueleto, durante el desarrollo de su proyecto emplean una SVM (Máquina de Soporte Vectorial) basada en la función del kernel para realizar el reconocimiento de la actividad humana. Li et al. [5] presentan un modelo para la detección automática de cuerpos humanos articulados en imágenes de alta resolución, empleando una SVM lineal que permite realizar una búsqueda del objeto a detectar de entre miles de objetos diferentes.

El presente artículo se encuentra organizado como sigue, en la sección 2 se provee la descripción general de la construcción del robot MECX1, así como la fundamentación teórica de los modelos y técnicas empleados en este trabajo de investigación.

2. Métodos

Debido a que el objetivo del presente trabajo de investigación propone un robot móvil capaz de determinar si existe un cuerpo humano dentro del espectro de

visión del sensor, es posible dividir en dos partes el problema general, el primer reto está relacionado con la construcción del robot móvil, mientras que en una segunda instancia se encuentra el diseño e implementación de los modelos matemáticos de Inteligencia Artificial (IA) propuestos para la detección de cuerpos humanos bajo diferentes condiciones de iluminación, pose, y ángulo de captura.

Construcción del robot móvil

El diseño del robot móvil al que se le ha dado el nombre de MECX1 está inspirado en los robots turtlebots, los cuales presentan una estructura en pisos apilados verticalmente. La base del MECX1 es un iRobot Create 2 mostrado en la figura 2, el cual es un robot móvil remanufacturado a partir de un Roomba 600, que ha sido liberado por el fabricante para ser reprogramado y controlar su hardware libremente. Create 2 incluye una interfaz abierta Roomba (OI), con la que es posible controlar los diferentes sensores y actuadores integrados en el robot a través de una serie de comandos, el set de instrucciones incluye el accionamiento de actuadores, sonido, limpieza y sensores, la información es leída o modificada a través de una PC o microcontrolador externos por medio de una conexión serial-USB cableada con una velocidad de comunicación de 19,200 hasta 115,200 baudios, otras características de interés es que dispone de 6 sensores infrarrojos de corto alcance situados en el parachoques para detección de obstáculos sin contacto, el control de motores se realiza con PWM y cada motor tiene un encoder que brinda información sobre su posición, también integra una batería recargable y se incluyen las funciones básicas divididas en 58 diferentes paquetes de datos, cuya descripción es posible encontrar en [6].

En el presente trabajo de investigación se propone emplear el Create 2 controlado con una MiniPC que se ensambló con el kit de desarrollo de Nvidia Jetson TK1, el cual puede verse en la figura 3 y una carcasa que fue impresa en una máquina 3D de inyección de plástico. La tarjeta madre cuenta con un procesador ARM Cortex A-15 de cuatro núcleos, que integra tecnología GPU Kepler (Graphics Processing Unit) con 192 núcleos, lo cual da como resultado un alto rendimiento a bajo consumo de energía, recomendada en tareas que necesitan cálculo intensivo, lo

que la hace ideal para la incorporación de los modelos de Inteligencia y Visión Artificial necesarios en el presente trabajo de investigación para la detección de cuerpos humanos.



Figura 1 Robot móvil iRobot Create 2 programado por puerto serial con la interfaz proporcionada por el fabricante.



Figura 2 Kit de desarrollo NVIDIA Jetson TK1.

Finalmente, al MECX1 se le ha integrado un Asus Xtion pro-live para obtener imágenes RGB-3D, por lo que incluye un sensor de infrarrojos para la detección adaptativa de la profundidad, imágenes en modo de color RGB con una máxima resolución de 1280 x 1024 píxeles e incluye también un par de micrófonos que pueden ser usados para comandos de voz sonido, lo que la hace una opción ideal al ofrecer altas prestaciones en la captura de los movimientos del cuerpo humano en tiempo real.

La Jetson TK1 es alimentada con una batería LiPo (Litio y Polímero) de 11.1V y 2200 mA, que es colocada en el primer piso del robot, mientras que en el segundo piso se encuentra la miniPC Jetson, el sensor asus Xtion es colocado con una

extensión vertical en la parte superior del MECX1, tal como se puede ver en la figura 4. Cabe mencionar que en la Jetson TK1 se encuentran implementados los algoritmos de IA, el procesamiento de imágenes en tiempo real, así como las instrucciones de control para accionar los diferentes componentes del iRobot Create 2.

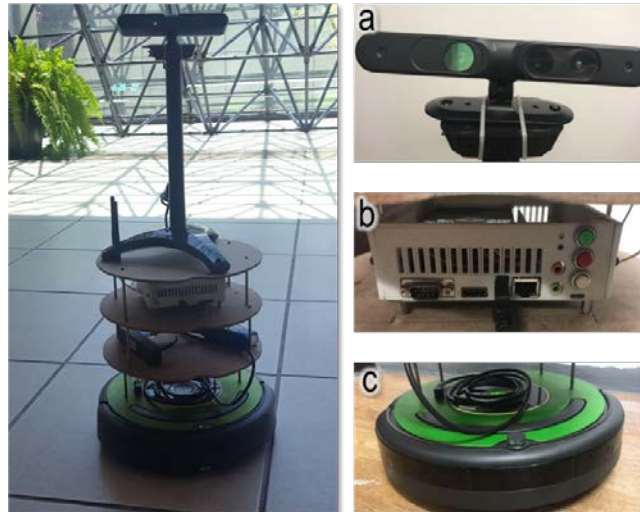


Figura 4 Robot MECX1 a) sensor Asus Xtion pro-live, b) kit de desarrollo de Nvidia Jetson TK1, c) iRobot Create 2.

Detección del cuerpo humano

Para describir cómo se realizó la detección del cuerpo humano se propone dividir en tres pasos el proceso general, tal como se ilustra en la figura 5, el primer paso está enfocado en la captura de las imágenes dinámicas a partir del sensor Asus Xtion pro-live montado en el robot, posteriormente se convierte a escala de grises y se separa el área de interés del resto de la imagen extrayendo el objeto de estudio. El paso dos está dedicado al proceso de detección de puntos de interés para posteriormente generar vectores descriptivos que caractericen la fisonomía del cuerpo humano en diferentes poses. Para el desarrollo del paso tres es necesario obtener una base de datos con vectores de cada imagen ya que serán aprendidos por las Memorias Asociativas Alfa-Beta ($MA\alpha\beta$), el cual es el modelo de Inteligencia Artificial (IA) propuesto en este estudio para determinar si una forma es o no un cuerpo humano.

Cabe denotar que se extrajeron dos tipos de imágenes del Asus Xtion pro-live, el primer tipo se denominan RGB-3D, que fusionan en una imagen RGB la información del sensor infrarrojo, las cuales son capaces de brindar información representada en color de la distancia de los elementos con respecto al sensor montado en el robot MECX1, mientras que el segundo tipo de imágenes son capturadas con una cámara tradicional, las cuales también están en modo de color RGB.

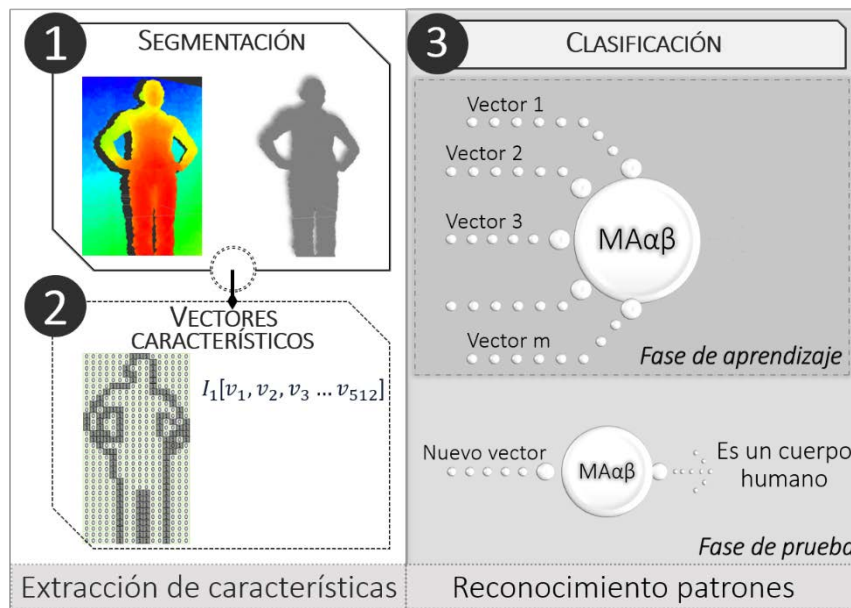


Figura 3 Proceso general para la detección de la forma del cuerpo humano.

Se realizó una base de datos integrada con un total de 24,000 imágenes RGB-3D y 24,000 RGB tradicionales tomadas de forma paralela, es decir que por cada imagen RGB-3D se tiene su homólogo en RGB tradicional tomada en el mismo instante de tiempo. Ahora bien, de las 24,000 imágenes RGB-3D 8,000 son positivas es decir que, pertenecen a los cuerpos de 20 personas en las que se incluyen hombres y mujeres con diferentes vestuarios y características físicas, que posaron en diferentes posiciones, mientras que las 16,000 imágenes restantes fueron extraídas de 50 objetos que no son cuerpos humanos, por tanto se debe sobre entender que se tiene la misma cantidad de imágenes positivas y negativas

en RGB tradicionales. Las características tanto de las imágenes RGB-3D como de las RGB tradicionales se muestran a continuación:

- Tamaño en pixeles: 512px ancho por 1024px alto.
- Resolución de la imagen: 300 ppp.
- Modo de color: RGB
- Profundidad de pixel: 24 Bits
- Formato de imagen digital: JPEG con extensión .jpg

En la figura 6 se muestran un ejemplo de las imágenes positivas y negativas que están incluidas en una base de datos creada para este trabajo de investigación, los incisos a), b), c) y d) son ejemplos de imágenes RGB tradicionales mientras que los incisos e) y f) son homólogos de las imágenes de los incisos anteriores pero en formato RGB-3D. a) y b) pertenecen a objetos que el robot móvil podría encontrarse mientras navega en su entorno pero que no deben ser reconocidas como cuerpos, por lo que se les denomina como negativas. Los incisos c) y d) forman parte de las imágenes definidas como positivas, ya que contienen sujetos de estudio bajo diferentes poses, distancias y ángulos del robot.

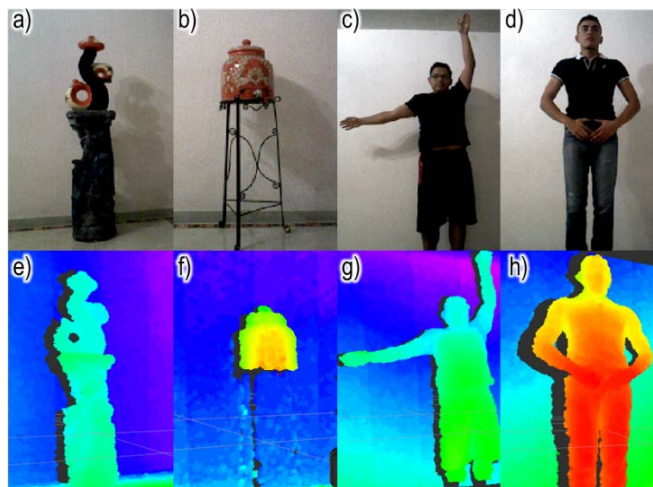


Figura 6 Ejemplos de imágenes de la base de datos, a) y b) imágenes negativas en RGB tradicional, c) y d) imágenes positivas en RGB tradicional, e) y f) imágenes negativas en RGB-3D, g) y h) imágenes positivas en RGB-3D.

Segmentación

Para la segmentación del área de interés es necesario realizar un pre-procesamiento a cada una de las imágenes dinámicas RGB-3D capturadas por el sensor Asus Xtion pro-live, las cuales combinan en cada pixel información del color en modo RGB así como de la profundidad. El primer tratamiento realizado a las imágenes es la conversión a escala de grises empleando el Algoritmo 1 de la tabla 1, no obstante, en este mismo proceso también se realiza la segmentación, en donde por medio de un umbral variante se seleccionan solo los pixeles que cumplen con dicha condición con la finalidad de separar del resto de la imagen el área de interés.

Tabla 1 Escala de grises y segmentación.

| |
|---|
| Algoritmo 1. Escala de grises y segmentación |
| Entrada: Imagen en RGB tradicional y RGB-3D, Salida: Imagen segmentada en imagen RGB y RGB-3D en escala de grises |
| <ol style="list-style-type: none"> 1. Recorrer desde (f=1) hasta (f<=alto) //recorre fil 2. Recorrer desde (c=1) hasta (c<ancho) //recorre col 3. Recorrer desde (i=0) hasta (i<=2) //para el recorrido del RGB 4. suma_canalesRGB_3D+=imagen_RGB_3D[i]; // acum del colorXpixel 5. suma_canalesRGB+=imagen_RGB[i]; // acum del colorXpixel 6. Fin de recorrer i 7. media_aritméticaRGB_3D= suma_canalesRGB_3D/3; 8. media_aritméticaRGB= suma_canalesRGB/3; 9. suma_canales=0;// limpia acumulador 10. Si umbral_inferior<=media_aritmética>umbral_superior //evaluación umbral 11. RGB-3D_grises[f][c]= media_aritméticaRGB_3D //mantener valor 12. RGB_tradicional_grises[f][c]= media_aritméticaRGB //mantener valor 13. En caso contrario 14. RGB_tradicional_grises[f][c]=255 // se convierte en blanco 15. RGB-3D_grises[f][c]= 255 // se convierte en blanco 16. Fin de Si 17. Fin de recorrer c 18. Fin de recorrer f 19. Crear y guardar la información del cuerpo de la imagen en un archivo |

El resultado de aplicar el algoritmo de segmentación y escala de grises puede verse en la imagen figura 7, en donde los primeros tres incisos forman parte de las muestras de diferentes sujetos de estudio tomadas con el sensor Asus Xtion, los incisos d), e) y f) son la imagen resultante de aplicar el algoritmo de escala de

grises mientras g), h) y i) son el resultado final de la segmentación por umbralización variable.

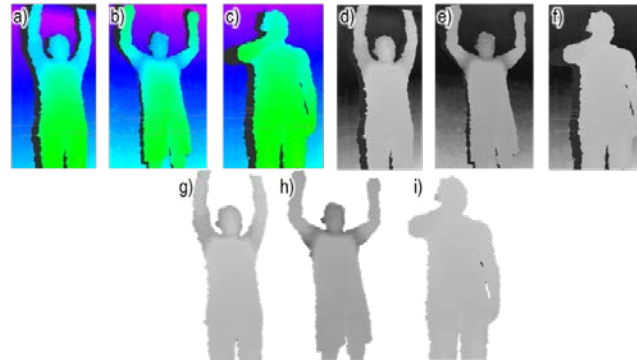


Figura 7 Resultados del proceso del preprocesamiento de imagen, a), b) y c) imágenes RGB-3D originales, d),e) y f) imágenes en escala de grises, g),h) y i) segmentación del cuerpo humano de la imagen original.

Posteriormente, se propone emplear la posición de los píxeles segmentados en el proceso anterior como referencia para realizar una nueva segmentación en la imagen RGB, de modo que se realiza una coincidencia entre la imagen RGB-3D segmentada y RGB, aquellos píxeles de la imagen segmentada que tengan un valor igual a 255(blanco) se correlacionan con su homólogo en la imagen RGB convertidos a escala de grises y se sustituye el valor original por 255(blanco), de manera que con este proceso se segmenta también la imagen RGB. En la figura 8 se muestran los resultados obtenidos de la segmentación indirecta incluida en la tabla 1.



Figura 8 Resultados del proceso de la segmentación indirecta de formas humanas en las imágenes originales en escala de grises generadas a partir de las imágenes RGB-3D previamente segmentadas.

Una vez que son segmentadas las imágenes positivas y negativas RGB-3D y RGB tradicionales, se guardan en un nuevo set de entrenamiento denotado como $F_i^n | \forall n \in \{1, 2, \dots, p\}, \forall i \in \{c\}$, donde p es igual al número de muestras del set de entrenamiento y c es la clase booleana asignada, donde 1 es igual a imagen positiva, es decir que es un cuerpo, mientras que -1 significa que esta etiquetado como imagen negativa.

Extracción de vectores característicos

La extracción de vectores característicos está representado en el paso 2 de la figura 5, cuyo objetivo es obtener a partir de las imágenes segmentadas negativas y positivas un vector que integre rasgos significativos que permitan distinguir a un cuerpo humano del resto de los elementos que el robot puede encontrar dentro de su entorno de navegación. La extracción de características se realiza empleando rejillas, en donde primeramente se debe dividir la imagen RGB-3D en celdas de igual tamaño, de modo que considerando que las dimensiones de las imágenes son de 1024*512 pixeles se propone que cada celda tenga una dimensión de 32*32 pixeles, con lo cual se obtiene una matriz de 16*32 celdas, tal como se muestra en la figura 9, ahora bien, en el inciso c) se sobrepone la rejilla en la imagen RGB-3D previamente segmentada.

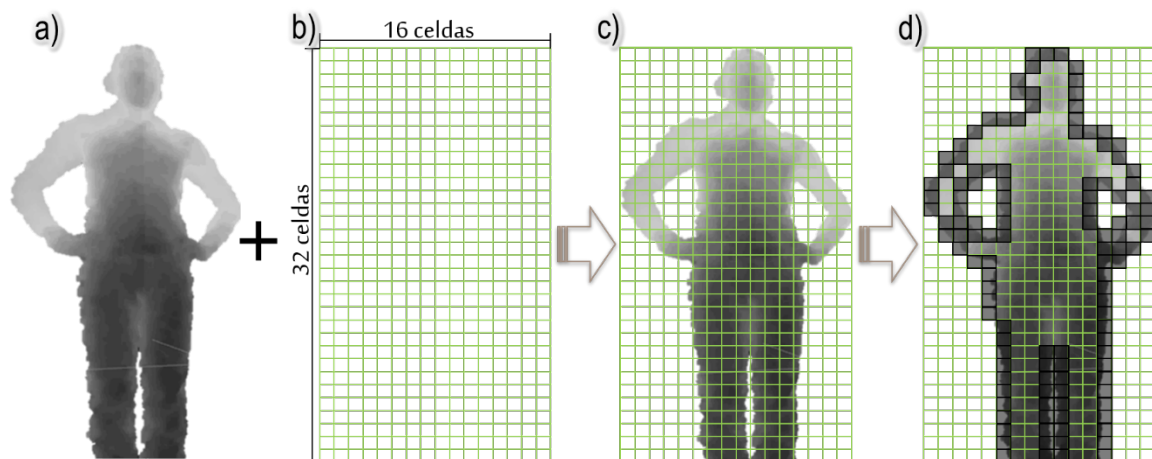


Figura 4 Proceso de extracción de características empleando contorno por rejillas.

Finalmente, la rejilla se representa como una matriz bidimensional, lo que implica que es posible asignarle un valor booleano a cada celda, las celdas que conforman el contorno de la figura se les asigna el valor de 1, mientras que aquellas que no forman parte de la silueta contendrán un 0 (cero). Para ello se recorre la imagen por sectores de 32×32 píxeles, en decir que se procesa celda a celda para evaluar el valor que le será asignado, de modo que a todas aquellas casillas que solo contengan píxeles con niveles de gris igual a 255 se le asigna el valor de 0 (cero), a las casillas que en donde todos los píxeles sean inferiores a 255 también se les asigna el valor de 0 (cero) y las únicas casillas a las que se les establece el valor de 1 son a las que presenten píxeles mixtos, lo que indica que contienen valores iguales a 255 y también inferiores a ese nivel, en figura 9 es posible observar que las casillas de la rejilla que han sido marcadas en color negro representan las celdas seleccionadas como siluetas.

Una vez que se tiene la matriz de cada una de las imágenes 24,000 RGB-3D segmentadas tanto positivas como negativas, es posible generar los vectores característicos para ello se concatena el valor booleano de cada celda de la rejilla $V(x_n, y_m)$, siendo (x_n, y_m) las coordenadas de cada casilla y V el valor booleano asignado.

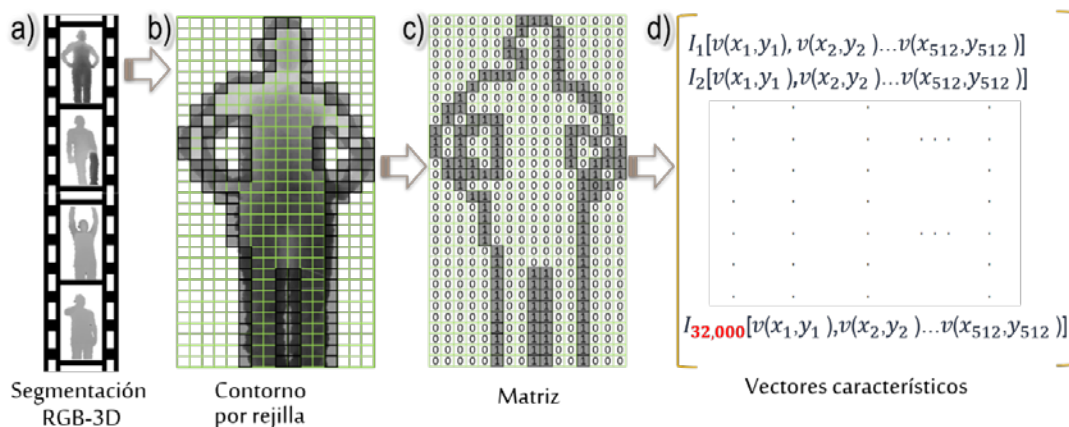


Figura 10 Proceso para generar vectores característicos a) base de datos con imágenes segmentadas, b) sobre posición de rejillas para determinar contorno, c) asignación de valores de las celdas de la matriz, d) generación de vectores característicos de cada imagen positiva y negativa de la base de datos.

En la figura 10 se muestra el proceso de la extracción de vectores característicos de todas las imágenes previamente segmentadas, para ello se emplean rejillas para determinar aquellas que forman parte de la silueta de la figura segmentada, en el inciso c) se muestra el valor que fue asignado a cada celda dependiendo si forma parte o no del contorno de la figura. En el inciso d) se muestra como se genera la base de datos con los valores concatenados en una cadena a partir de los valores asignados a cada celda de la matriz, en total se obtiene una base de datos de 32 patrones con 512 valores booleanos (0-1) concatenados cada uno.

Clasificación

Las Memorias Asociativas (AM) tienen su inspiración en la memoria humana, en consecuencia su ventaja competitiva reside en almacenar información que puede ser recuperada de forma eficiente, con el tiempo han surgido nuevas líneas de investigación enfocadas en mejorar los modelos, lo cual dio como resultado un mejor rendimiento al recordar patrones previamente aprendidos, pero al mismo tiempo les permitió asociar patrones alterados por un ruido de tipo aditivo, sustractivo o combinado y que no fueron presentados a la memoria durante su aprendizaje. Ritter and Sussaner [7] presentan las memorias morfológicas las cuales retoman las cualidades de los modelos de las memorias clásicas integrando los conceptos de dilatación y erosión de morfología matemática, por lo que las memorias morfológicas hacen uso de los operadores de máximos (\vee) y mínimos (\wedge), lo cual da como resultado que en el tipo auto-asociativo sean capaces de almacenar y recuperar de manera perfecta cualquier cantidad de patrones aprendidos.

Memorias Asociativas Alfa-Beta

Las Memorias Asociativas Alpha-Beta ($AM\alpha\beta$) fueron propuestas en 2002 Cornelio [8], las cuales retoman los principios fundamentales de los operadores de \max (\vee) y \min (\wedge) de las Memorias Morfológicas, lo que les permite recuperar todos los patrones del set de entrenamiento, no obstante, su aportación radica en

la inclusión de dos nuevos operadores Alpha (α) y el operador Beta (β) usado para la recuperación y clasificación de patrones. La incorporación al modelo de los operadores α y β hace posible que la memoria sea capaz de clasificar también patrones que no han sido previamente aprendidos, lo que la convierte en un modelo de aprendizaje supervisado empleado en problemas de reconocimiento y clasificación [9, 10, 11, 12, 13, 14, 15].

Operadores $\alpha\beta$

Las AM $\alpha\beta$ se caracterizan por incluir a su modelo los operadores binarios α y β , el operador α es utilizado para la fase de aprendizaje y el operador β durante la fase de recuperación. Para la definición de los operadores binarios se tienen los conjuntos A y B, los cuales contienen los elementos:

$$A = \{0,1\} \text{ y } B = \{0,1,2\}$$

Con base en los conjuntos **A y B** el operador binario α se define como $\alpha = AxA \rightarrow B$, así mismo, el operador β está dado por $\beta = BxA \rightarrow A$. La demostración matemática y propiedades algebraicas de ambos operadores se pueden encontrar en [8].

Tabla 2 Operadores binarios α y β .

| $\alpha = AxA \rightarrow B$ | | | $\beta = BxA \rightarrow A$ | | |
|------------------------------|---|----------------|-----------------------------|---|---------------|
| | | $\alpha(x, y)$ | | | $\beta(x, y)$ |
| | 1 | 1 | | 0 | 0 |
| | 0 | 0 | | 0 | 0 |
| | 2 | 2 | | 0 | 1 |
| | 1 | 1 | | 1 | 1 |
| | 0 | 0 | | 1 | 1 |

Fases de la AMαβ

Las AMαβ forman parte de los modelos de aprendizaje supervisado, por tanto hacen uso de una base de datos con patrones que representan las características más significativas que definen los objetos que se desean reconocer, adicionalmente los patrones están debidamente catalogados en clases. Los métodos de validación como k-fold-cross validation, bootstrap o leave one out son empleados para segmentar en dos partes la base de datos, la primera parte de los patrones es llamada set de entrenamiento y es usada durante la fase de aprendizaje del modelo, mientras que a la segunda parte se le da el nombre de set de prueba y es mediante este que determina la capacidad del modelo para asociar un patrón desconocido con algún patrón que haya sido previamente aprendido. Las AMαβ al igual que otros métodos supervisados presentan tres fases: fase de aprendizaje, fase de recuperación y fase de prueba, de modo que, dependiendo de la fase se hace uso del set de entrenamiento o el de prueba.

Fase de aprendizaje

La finalidad de esta fase es construir la memoria asociativa a partir de los patrones del set de entrenamiento. Las técnicas para codificar los patrones en una matriz de aprendizaje M dependen del modelo matemático de cada tipo de memoria.

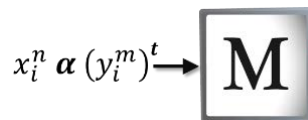


Figura 5 Fase de aprendizaje de las AMαβ.

En la figura 11 se ejemplifica como se crea M en las AMαβ, la cual se construye a partir de cada uno de los patrones de entrada que pertenecen al set de entrenamiento, los patrones deben ser vectores binarios denotados por $x_i^n | \forall n \in \{1, 2, \dots, p\}, \forall i \in \{1, 2, \dots, l\}$, donde p = número de patrones de entrada and l = tamaño del patrón. Se debe tener en cuenta que el vector x_i^n se conforma a

partir de las características representativas del objeto a reconocer, por lo tanto, el vector y_c^m también es binario y está definido como $y_i^m | \forall m \in \{1, 2, \dots, c\}, \forall i \in \{1, 2, \dots, l\}$, donde c = número de clases y l = tamaño de patrón, por tanto para cada patrón de entrada x_i^n se le asocia una salida deseada y_i^m , lo que da como resultado la pareja ordenada (x_i^n, y_i^m) .

$$x_i^n = | : | \quad y_i^m = | : | \quad (1)$$

Para construir M se siguen los siguientes pasos:

- a) Se toma el par ordenado (x_i^n, y_i^m) y el operador α definido es aplicado mediante la Tabla 1, por tanto para cada (x_i^n, y_i^m) una matriz primaria es generada la cual se define como $[M'_{ij}]^n | \forall n \in \{1, 2, \dots, p\}, \forall i, j \in \{1, 2, \dots, l\}$.

$$[M']^n = [x^n \ \alpha \ (y^m)^t]_{l \times l} = \begin{bmatrix} x_1^n \\ x_2^n \\ \vdots \\ x_i^n \end{bmatrix} \alpha [y_1^m \ y_2^m \ \dots \ y_l^m] \quad (2)$$

- b) Se aplica el operador $\max V$, para ello se determina el valor máximo de cada casilla $\alpha (x_i^n, y_i^m)$ de las matrices $[M']^n$ generadas en el paso a).

$$v_{ij} = \bigvee_{n=1}^p [\alpha(x_i^n, (y_i^m)^t)]^n \quad (3)$$

El resultado obtenido es una matriz de aprendizaje M .

$$M = \begin{bmatrix} v_{11} & v_{12} & \dots & v_{1l} \\ v_{21} & v_{22} & \dots & v_{2l} \\ \vdots & \vdots & \dots & \vdots \\ v_{l1} & v_{l2} & \dots & v_{ll} \end{bmatrix} \quad (4)$$

Se debe considerar que las $AM\alpha\beta$ son capaces de funcionar en dos modos autoasociativo y heteroasociativo, la diferencia entre ambas radica en la definición del vector y_i^m , de modo en una memoria $\alpha\beta$ heteroasociativa se espera que el

patrón x_i^n este catalogado en una clase, por lo que $x_i^n \neq y_i^m$ ya que x_i^n representa las características de un objeto, mientras y_i^m representa la clase a la que pertenece el objeto. En las AM $\alpha\beta$ que son autoasociativas $x_i^n = y_i^m$, lo que significa que la memoria trata de asociar las características del objeto que se desean clasificar con las características de los objetos que la AM $\alpha\beta$ haya aprendido previamente, de modo que en las memorias autoasociativas $p = c$.

Fase de recuperación

Una vez que M ha sido construida, se busca determinar la eficiencia de la AM $\alpha\beta$ para recuperar o recordar los patrones previamente aprendidos. En la figura 12, se muestra un esquema de la fase de recuperación, donde cada uno de los patrones los patrones x_i^n del set de entrenamiento se le presentan a M , durante esta fase se evalúa el rendimiento de la AM $\alpha\beta$ para asociar el patrón x_i^n con alguno de los patrones que ha aprendido previamente, por tanto a la salida se obtiene un vector binario representado por y_i^m . De modo que sí se trata de una memoria heteroasociativa, la salida esperada y_i^m será igual a la clase asociada del patrón correspondiente, no obstante, si se trata de una memoria autoasociativa entonces se espera que $y_i^m = x_i^n$.



Figura 6 Fase de recuperación de las Am $\alpha\beta$.

Los pasos que se siguen durante la fase de recuperación son:

- a) Se le presentan los patrones x_i^n a M , y se le aplica el operador β mostrado en la tabla 1, esto da como resultado n matrices como la que se mostrada a continuación.

$$y_i^m = \begin{bmatrix} \beta(v_{11}, x_1^n) & \beta(v_{12}, x_1^n) & \dots & \beta(v_{1l}, x_1^n) \\ \beta(v_{21}, x_2^n) & \beta(v_{22}, x_2^n) & \dots & \beta(v_{2l}, x_2^n) \\ \vdots & \vdots & \dots & \vdots \\ \vdots & \vdots & \dots & \vdots \\ \beta(v_{l1}, x_l^n) & \beta(v_{l2}, x_l^n) & \dots & \beta(v_{ll}, x_l^n) \end{bmatrix} \quad (5)$$

b) Finalmente, para obtener el vector de salida y_i^m , es necesario aplicar el operador \wedge en cada fila de la matriz obtenida en el paso c.

$$y_i^m = \begin{bmatrix} \bigwedge_{i=1}^l \beta(v_{1i}, x_i^n) \\ \bigwedge_{i=1}^l \beta(v_{2i}, x_i^n) \\ \vdots \\ \bigwedge_{i=1}^l \beta(v_{li}, x_i^n) \end{bmatrix} \quad (6)$$

Los pasos a, b, c y d, pueden ser resumidos bajo la expresión:

$$y_i^m = \bigwedge_{i=1}^l \beta \left\{ \left[\bigvee_{n=1}^p [\alpha(x_i^n, (y_i^m)^t)]^n \right], x_i^n \right\} \quad (7)$$

Fase de prueba

En esta fase se busca determinar el desempeño de la $AM\alpha\beta$ cuando clasifican patrones que no han sido aprendidos por la memoria.

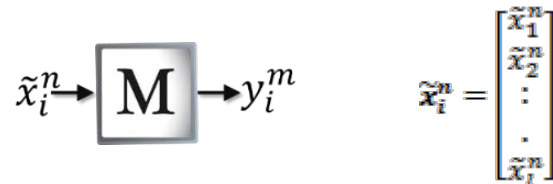


Figura 7 Fase de prueba de las $Am\alpha\beta$.

En la figura 13 se muestra la fase de prueba, en donde los patrones que no pertenecen al conjunto de entrenamiento y que por tanto contienen ruido sustractivo se le presentan a M , estos patrones son denotados por \tilde{x}_i^n , y se

encuentran almacenados en una base de datos denominada set de prueba. La fase de prueba es muy parecida a la fase de recuperación ya que se siguen los mismos pasos c y d, los cuales son resumidos en la expresión 13, no obstante, los patrones que se le presentan a la matriz de aprendizaje M son aquellos que están almacenados en el set de prueba.

$$y_i^m = \bigwedge_{i=1}^l \beta \left\{ \begin{matrix} [v_{11} & v_{12} & \dots & v_{1l}] \\ [v_{21} & v_{22} & \dots & v_{2l}] \\ \vdots & \vdots & \dots & \vdots \\ [v_{l1} & v_{l2} & \dots & v_{ll}] \end{matrix} \right\} \begin{matrix} [\tilde{x}_1^n] \\ [\tilde{x}_2^n] \\ \vdots \\ [\tilde{x}_l^n] \end{matrix} \quad (8)$$

3. Experimentos y resultados

La base de datos creada para entrenar y probar a las $AM\alpha\beta$ contiene 24,000 imágenes RGB-3D, las cuales fueron pre-procesadas para obtener imágenes en las que se segmenta el área de interés, posteriormente se extrae una base de datos con vectores característicos extraídos de cada imagen, por lo que se tienen un total de 24,000 vectores, integrados de 512 valores booleanos generados mediante el empleo de rejillas, los vectores fueron también catalogados en 8,000 vectores positivos los cuales provienen a partir de la silueta de cuerpos humanos y 16,000 vectores negativos que se generan de las siluetas de otros elementos que es posible que el robot encuentre en su entorno de navegación pero que no son un cuerpo humano. A los vectores también se le puede considerar como patrones que la $AM\alpha\beta$ debe aprender para posteriormente ser capaz de reconocerlo, los patrones son definidos como $x_i^n | \forall n \in \{1, 2, \dots, p\}, \forall i \in \{1, 2, \dots, l\}$ donde p = número de patrones o imágenes, mientras que l = es la dimensión de la cadena concatenada de cada patrón.

Para entrenar a las $AM\alpha\beta$ y posteriormente determinar su rendimiento al reconocer lo aprendido es necesario emplear un método de validación para dividir la base de datos en aquellos vectores o patrones que aprenderá la memoria (conjunto de entrenamiento) y aquellos con los que se le pondrá a prueba (conjunto de prueba), se propone hacer uso del método k-fold cross validation con $k=10$

para generar el set de entrenamiento y prueba, para ello se divide aleatoriamente la base de datos en k subconjuntos aproximadamente del mismo tamaño, en donde los subconjuntos $k-1$ constituyen el set de entrenamiento y el subconjunto restante forma el set de prueba.

Por tanto de los 24,000 patrones que integran la base de datos original, 21,600 patrones constituyen el set de entrenamiento, mientras que el conjunto de prueba queda integrado por el 10% restante, es decir 2,400 patrones. Es necesario repetir el proceso de validación cruzada k veces, de modo que se hacen 10 rondas realizando una rotación del k subconjunto que genera el set de prueba, el objetivo es que cada patrón forme parte del conjunto de prueba en alguna de las rondas permitiendo probar el rendimiento de las AM $\alpha\beta$ con diferentes patrones en los conjuntos de aprendizaje y prueba.

Una vez que se tienen los conjuntos de aprendizaje y prueba formados con patrones o vectores característicos integrados por una cadena binaria de 0 y 1, es posible comenzar la fase de aprendizaje de las AM $\alpha\beta$ descrita en la sección 2.5.3-Fase de aprendizaje, empleada en su modo autoasociativo donde $x_i^n = y_i^m$, cuyo objetivo es generar la matriz de aprendizaje M a partir de los 21,600 patrones del set de entrenamiento. Posteriormente, se propone probar la eficiencia de la AM $\alpha\beta$ cuando recuerda a cada uno de los patrones previamente aprendidos, para ello es necesario presentar a M cada uno de los patrones que forman el conjunto de entrenamiento y determinar cuántos de ellos fue capaz de recordar exitosamente, el procedimiento de encuentra descrito en la sección 2.5.3-Fase de recuperación. Ahora bien, para probar la eficiencia de las AM $\alpha\beta$ en la detección de patrones nuevos, que pueden contener variaciones con respecto a aquellos con los que fue entrenado, es necesario hacer uso del conjunto de prueba denotados como x_i^m , para ello se le presentan a la M uno a uno los 2,400 patrones, y se determina el índice de asertividad, que se calcula tomando en cuenta aquellos patrones que fue capaz de clasificar exitosamente entre las dos clases, siendo 1 un cuerpo humano y un -1 un objeto que no lo es.

Rendimiento de las AM $\alpha\beta$ en la detección

La programación de cada uno de los algoritmos se realizó en lenguaje C++, instalado en una mini-PC ensamblada con un NVIDIA Jetson TK1 developer kit con un procesador Quad-Core ARM Cortex, 16 GBytes de memoria RAM, bajo un sistema operativo Ubuntu 14.01. Para la implementación de los algoritmos de segmentación, rejillas y las AM $\alpha\beta$ fue necesario programar todas sus fases y crear una biblioteca propia.

Las pruebas aplicadas a la AM $\alpha\beta$ fueron diseñadas para determinar su rendimiento tanto en el proceso de recordar los patrones aprendidos como para medir su capacidad en la clasificación de patrones no aprendidos, las cuales pueden ser divididas en la fase de recuperación y la fase de prueba.

En la figura 14 se muestran los resultados obtenidos durante la fase de recuperación de las 10 rondas en las que el algoritmo se ejecuta, en cada ronda se forman nuevos conjuntos de aprendizaje y prueba rotando los patrones de la base de datos original que contiene los vectores característicos binarios extraídos de los rostros de 20 sujetos de estudio.

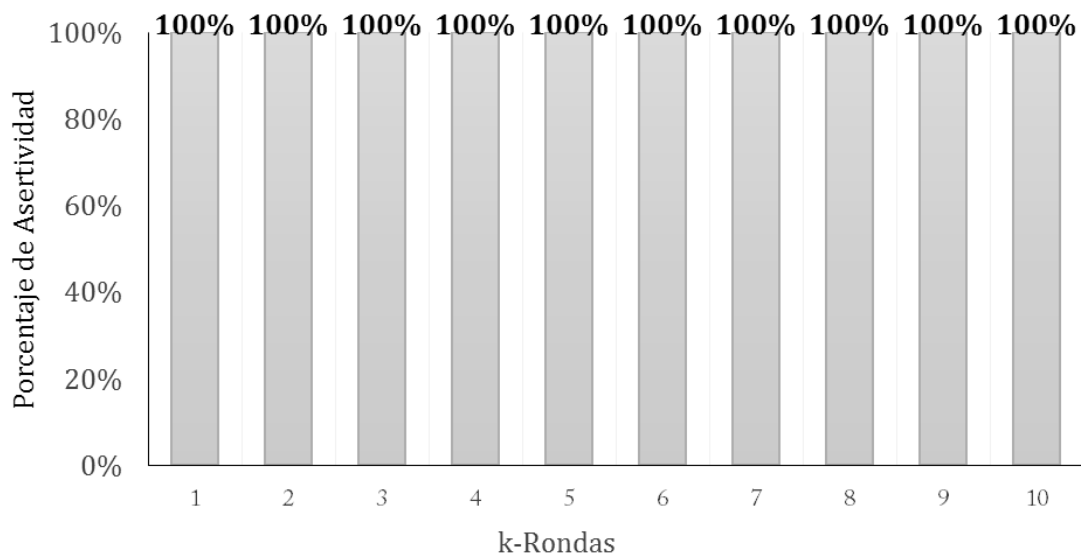


Figura 14 Resultados de la fase de recuperación obtenidos al clasificar los vectores característicos del conjunto de entrenamiento bajo el método de validación K-fold cross validation con k=10.

Es necesario denotar que durante la fase de recuperación solo se hace uso del conjunto de entrenamiento, ya que la finalidad es determinar el rendimiento de la $AM\alpha\beta$ al recordar los patrones aprendidos, no obstante, en cada ronda se generan los conjuntos de aprendizaje y prueba para poder rotar los patrones que pertenecen a cada conjunto, lo que da como resultado 10 diferentes matrices de aprendizaje M , por tanto, se debe evaluar la capacidad de recordar de cada M , para ello es necesario que en cada ronda uno a uno los patrones x_i^m del set de aprendizaje se presenten a la matriz M previamente entrenada, en donde se emplean el operador β y $\max V$ para generar un patrón y_i^m , seguidamente se determina si este patrón fue recuperado correctamente, al terminar con todos los patrones del conjunto de entrenamiento el porcentaje de los aciertos obtenidos es calculado.

La segunda evaluación que se le realizó a las $AM\alpha\beta$ está enmarcada en fase de prueba y cuyo objetivo es determinar el desempeño de la memoria al relacionar patrones que no fueron aprendidos con aquellos que han sido incluidos en la fase de entrenamiento, de modo que durante esta evaluación la memoria funciona como clasificador. El algoritmo de las $AM\alpha\beta$ fue puesto a prueba 10 veces, en donde se generaron los conjuntos de entrenamiento y prueba mediante el empleo de k-fold cross validation, en la cual se rotan en cada ronda los patrones que integran los conjuntos con la finalidad de que cada patrón en alguna de las rondas forme parte una vez del conjunto de prueba. Para probar a la $AM\alpha\beta$ en cada ronda se le van presentan uno a uno los patrones x_i^m del conjunto de prueba, a la memoria M en donde se hace uso del operador β y $\max V$ para obtener un patrón de salida y_i^m , el cual es la respuesta para indicar con que patrón de los aprendidos previamente la memoria ha encontrado un mayor parecido, posteriormente patrón a patrón del conjunto de prueba se determina si la clasificación fue correcta y al concluir con los 2,400 patrones se calcula el porcentaje de aciertos en cada ronda. En la figura 15 se muestran los resultados que se han obtenido durante la fase de prueba, en donde el índice de asertividad más alto corresponde a la segunda ronda con un 96.9% lo que equivale a que la $AM\alpha\beta$ clasificó con éxito 2,325 de los

2,400 patrones evaluados, mientras que la tasa de precisión más baja corresponde a la sexta ronda con un 93.8% lo que indica que tuvo éxito al clasificar 2,241 patrones. Si se considera la tasa de precisión promedio de las 10 rondas se obtiene un 95.1% de éxito.

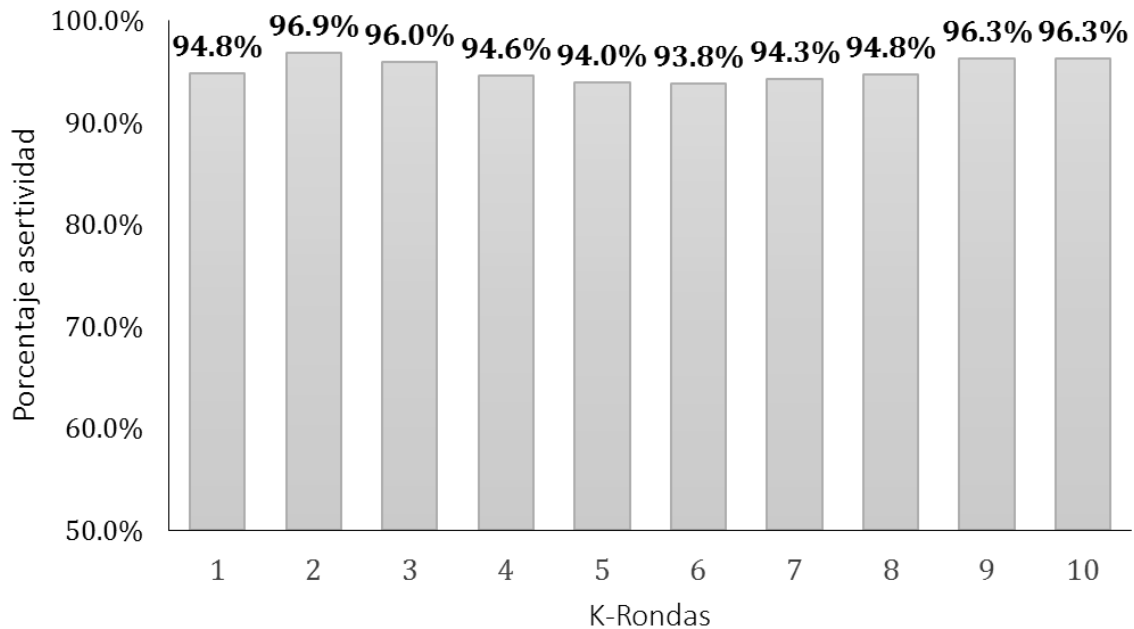


Figura 15 Resultados de la fase de prueba obtenidos al clasificar los vectores característicos bajo el método de validación K-fold cross validation con k=10.

4. Discusión

La interacción hombre-máquina es indispensable para los robots de servicio, ya que de ello dependerá una exitosa comunicación entre ambos y de forma colateral la correcta interpretación de las instrucciones realizadas por el ser humano, no obstante el problema es multifactorial lo que implica que es necesario dividirlo, en retos puntuales, es este estudio se propone que el robot sea capaz de detectar a un humano, para ello se diseñó y construyó un robot móvil, que integra un iRobot Create 2, un Mini-PC ensamblada con el kit de desarrollo de Nvidia Jetson TK1 y un sensor Asus Xtion pro-live del cual se capturaron imágenes 24,000 RGB-3D y 24,000 tradicionales RGB mientras el robot realiza rondas de navegación, las cuales ha sido catalogadas en dos clases, de primera clase se tienen 8,000

imágenes que se denominan como positivas y corresponden a imágenes que incluyen un cuerpo humano. De la segunda clase se tiene un total de 16,000 imágenes, las cuales son llamadas negativas, ya que contienen objetos que no son un cuerpo humano, pero que el robot se puede encontrar en su entorno mientras se desplaza. Posteriormente, se pre-procesan las 24,000 imágenes RGB-3D con el objetivo de separar el área que de interés del resto de la imagen, creando así una nueva base de datos con imágenes segmentadas, en el siguiente paso se extraen características distintivas de estas últimas imágenes para ello se emplean rejillas y el resultado son 24,000 vectores característicos conformados por una cadena de 512 valores booleanos. Los vectores también llamados patrones son aprendidos por las $AM\alpha\beta$ en una fase de entrenamiento, consecuentemente se evalúa el desempeño de las memorias para recordar los patrones aprendidos, y también se determina el comportamiento de las mismas al asociar un nuevo patrón con aquellos que ha aprendido previamente.

En la primera evaluación en la que se busca que la $AM\alpha\beta$ recuperen eficientemente cada uno de los patrones incluidos en su fase de entrenamiento, el resultado obtenido en la recuperación durante las 10 rondas del k-fold cross validation fue de 100%, lo que implica que no hubo olvido ninguno de los patrones aprendidos, los resultados corresponden a que las memorias auto-asociativas en esencia fueron concebidas para guardar información y recuperarla de forma eficiente al emplear los operadores máximo (\vee) y mínimo (\wedge).

Para la segunda evaluación en donde se determina la capacidad de la $AM\alpha\beta$ para clasificar un patrón nuevo que no fue presentado a la memoria durante la fase de entrenamiento, se obtienen que durante las 10 rondas del k-fold cross validation los resultados van desde un 93.8% hasta 96.9%, los cuales muestran el rendimiento de la memoria al momento de clasificar siluetas de cuerpos humanos de aquellas que no lo son, los resultados presentan una congruencia con los fundamentos de las $AM\alpha\beta$, ya que retoman algunas de las propiedades de matemática morfológica de los modelos predecesores, los cuales presentan mejores resultados al aprender formas incluidas en imágenes que al interpretar por ejemplo dimensiones de objetos.

Finalmente, y con base en los resultados obtenidos es posible decir que las AM $\alpha\beta$ son un modelo que presentan resultados relevantes al diferenciar de forma automática entre un cuerpo humano y todo aquello que no lo es, aun cuando la persona dentro de la imagen presente diferentes ángulos, distancia, iluminación y poses, ya que el bando de datos con el que fue entrenada la memoria asociativa está integrado con imágenes capturadas mientras el robot realizaba rondas de navegación, por lo que se presupone que en trabajos futuros es posible implementar el algoritmo en tiempo real en el robot móvil MECX1, con la intención de brindar a este la posibilidad de detectar si lo que está percibiendo por medio del sensor infrarrojo es o no una persona.

5. Bibliografía y Referencias

- [1] D. Duan, G. Gao, C. H. Liu y J. Ma, «Automatic person identification in camera video by motion correlation,» *Journal of Sensors*, vol. 2014, 2014.
- [2] Lillo, J. C. Niebles y A. Soto, «Sparse composition of body poses and atomic actions for human activity recognition in RGB-D videos,» *Image and Vision Computing*, vol. 59, pp. 63-75, 2017.
- [3] Jalal, Y.-H. Kim, Y.-J. Kim, S. Kamal y D. Kim, «Robust human activity recognition from depth video using spatiotemporal multi-fused features,» *Pattern recognition*, vol. 61, pp. 295-308, 2017.
- [4] Y. Wang, Y. Shi y G. Wei, «A novel local feature descriptor based on energy information for human activity recognition,» *Neurocomputing*, vol. 228, pp. 19-28, 2017.
- [5] Y. Li, X. Liu, S. Zhang y X. Ye, «Human articulated body recognition method in high-resolution monitoring images,» *Neurocomputing*, vol. 181, pp. 116-121, 2016.
- [6] Company, «iRobot Create 2 Open Interface (OI), Specification based on the iRobot Roomba 600,» 2015.
- [7] G. Ritter, J. D. de Leon y P. Sussner, «Morphological bidirectional associative memories,» *Neural Networks*, vol. 12, n^o 6, pp. 851-867, 1999.

- [8] Yáñez-Márquez, «Associative memories based on order relations and binary operators (In Spanish),» PhD Thesis (In Spanish), Center for Computing Research, México, pp. 300-311, 2002.
- [9] L. O. López-Leyva, C. Yáñez-Márquez, R. Flores-Carapia y O. Camacho-Nieto, «Handwritten Digit Classification Based on Alpha-Beta Associative Model,» Iberoamerican Congress on Pattern Recognition, Springer, pp. 437-444, 2008.
- [10] M. E. Acevedo-Mosqueda, C. Yáñez-Márquez y I. López-Yáñez, «Alpha-Beta bidirectional associative memories: theory and applications,» *Neural Processing Letters*, vol. 26, nº 1, pp. 1-40, 2007.
- [11] R. Godínez, I. López-Yáñez y C. Yáñez-Márquez, «Classifying patterns in bioinformatics databases by using Alpha-Beta associative memories,» *Biomedical Data and Applications*, Springer, pp. 187-210, 2009.
- [12] Román-Godínez y C. Yáñez-Márquez, «Complete recall on Alpha-Beta heteroassociative memory,» *Mexican International Conference on Artificial Intelligence*, Springer, pp. 193-202, 2007.
- [13] Alarcón-Paredes, O. Pogrebnyak y A. J. Argüelles Cruz, «Image Transform based on Alpha-Beta Associative Memories,» *Computación y Sistemas*, vol. 17, nº 4, pp. 527-541, 2013.
- [14] Yáñez-Márquez, M. E. Cruz-Meza, F. A. Sánchez-Garfias y I. López-Yáñez, «Using alpha-beta associative memories to learn and recall RGB images,» *de International Symposium on Neural Networks*, 2007.
- [15] Yáñez-Márquez, E. Cruz-Meza, F. Sánchez-Garfias y I. López-Yáñez, «Using alpha-beta associative memories to learn and recall RGB images,» *International Symposium on Neural Networks*, Springer, pp. 828-833, 2007.