

# Implementación de reconocimiento de objetos por color y forma en un robot móvil

Efraín Ernesto Arévalo-Vázquez, Arturo Zúñiga-López, Juan Villegas-Cortez,  
Carlos Avilés-Cruz

Universidad Autónoma Metropolitana, Azcapotzalco, Departamento de Electrónica,  
México, D.F.

al206200579@alumnos.azc.uam.mx, {azl, juanvc, caviles}@correo.azc.uam.mx

**Resumen.** La visión artificial la componen un conjunto de procesos destinados a realizar el análisis de imágenes. Estos procesos son: captación de imágenes, memorización de la información, procesado e interpretación de los resultados. En este artículo se presenta un avance en el aprovechamiento del poder de procesamiento de los nuevos sistemas embebidos para controlar un robot móvil por medio de un algoritmo de visión más que por sensores electrónicos. En este caso particular se buscó desde un robot móvil analizar un objeto con base a su color y forma, tal que el dispositivo pueda moverse de manera autónoma hacia el objeto para analizarlo, y que realice movimientos para buscar y encontrar un objeto idéntico a él, obtenido del resultado del procesamiento de imágenes.

**Palabras clave:** visión por computadora, sistemas embebidos, reconocimiento de objetos, robots móviles.

## 1. Introducción

Los sistemas de visión artificial han evolucionado poco a poco a medida que la tecnología ha evolucionado y permitido la creación de cámaras, computadoras y algoritmos de procesamiento más potentes. De forma básica un sistema de visión artificial está conformado por varios subsistemas capaces de realizar dos funciones fundamentales: captar la información de la escena real mediante la proyección en una imagen y analizar las imágenes para extraer la información que contienen [7] [5].

El mayor peso del análisis de las imágenes captadas se encuentra a nivel de software, lo que proporciona una mayor flexibilidad en cuanto a lo que se puede implementar con base a los resultados obtenidos [8]. Esto lo podemos aprender de los resultados usados en la aplicación de algoritmos de visión para el reconocimiento de instrumental médico [6], el reconocimiento de marcas de localización para rutas en robots móviles [3], y dándole libertad de movimiento a los robots móviles, usando toma de decisión con base en sistemas difusos [4], y también en la aplicación para desplazamiento en 3D con visión [7]. En los trabajos mencionados se muestran técnicas diversas para el análisis del entorno

de desplazamiento, y en nuestro caso nos enfocamos en el análisis espacial del espacio de color de las imágenes a usar del entorno del robot móvil [9], como una etapa importante, pero también considerando la implementación de rasgos característicos del objeto de estudio, como lo son los momentos de Hu [11]. Existen aplicaciones robustas de visión para posicionamiento con base únicamente a la visión [10], pero para nuestro propósito no se considera el posicionamiento, sino la detección y reconocimiento de un objeto específico en el campo de visión de una cámara montada en un robot móvil diferencial, todo desde el paradigma de la imagen digital y el procesamiento digital de imágenes.

En éste artículo se muestra la realización del diseño e implementación de un sistema embebido que permite controlar un robot móvil que analiza, por color y forma un objeto muestra, mediante una cámara digital, se dé la vuelta y avance mientras identifica diferentes objetos con colores y formas distintas, hasta detenerse enfrente del objeto buscado. El algoritmo de reconocimiento usa los momentos invariantes de Hu [11] [5], para poder tener invarianza suficiente para lograr nuestro objetivo. En todo momento se puede monitorear parte del procesamiento de las imágenes a través de una computadora por medio de una conexión inalámbrica a una red LAN, gracias a un servidor HTTP para este propósito.

En la sección 2 presentamos el desarrollo de esta propuesta, en la sección 3 mostramos los resultados de los experimentos realizados, y finalmente las conclusiones con las perspectivas del trabajo futuro se muestran en la sección 4.

## 2. Desarrollo

El desarrollo del sistema se realiza en tres etapas: *(i)* diseño e implementación de una metodología de detección e identificación del objeto de interés usando Procesamiento Digital de Imágenes, *(ii)* construcción del robot móvil diferencial con la cámara, y *(iii)* implementación del sistema de monitoreo en tiempo real. A continuación se describe cada una de estas etapas.

### 2.1. Metodología de detección e identificación del objeto de interés usando Procesamiento Digital de Imágenes.

Aquí se muestra cómo se analizan las imágenes, capturadas con una cámara web ordinaria, con el fin de buscar y detectar marca, color y forma de un objeto, para posteriormente estar enviando las imágenes para su visualización en el sistema de monitoreo. Las imágenes se tomaron con resolución de 320 x 240 píxeles a 32 bits de color. Las etapas del procesamiento digital se muestran en la figura 1, ahí se puede seguir con detalle cada uno de los bloques en su propósito y funcionamiento.

#### Inicio

Al encender el dispositivo: se carga el sistema operativo, se conecta automáticamente al modem por la conexión inalámbrica, inicio automático al servicio de monitoreo a través del servidor web y el servicio de conexión remota. El usuario

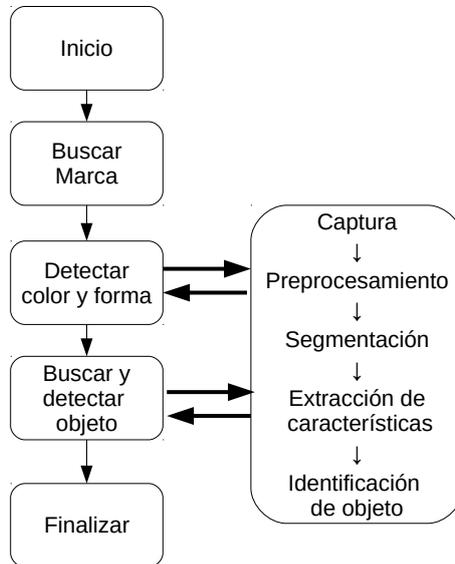


Fig. 1: Metodología propuesta para la detección y reconocimiento de objetos en un robot móvil diferencial.

se conecta vía SSH Secure Shell Client y se ejecuta el comando para iniciar el programa. Al iniciar el programa se verifica la conexión y el funcionamiento de la cámara web, en caso de error el programa se interrumpe y manda un aviso.

### Buscar marca

Se utiliza una marca distintiva para la colocación del objeto y para que el sistema identifique su ubicación, la cual se coloca frente al objeto y este es analizado. Consta de una hoja carta de color blanco colocada de manera horizontal con cuatro cuadrados huecos de contorno negro dibujados cerca de las esquinas de la hoja. El tamaño de los cuadrados es de  $3\text{cm}$  por lado y el grosor de la línea es de  $3\text{mm}$ . El tamaño puede variar, pero no deben quedar cubiertos al colocar el objeto a analizar. El objeto se coloca cerca de la hoja y en la parte media de los dos cuadrados inferiores. Si no se coloca en esta posición, el sistema no podrá analizarlo correctamente.

Al iniciar el programa, el robot busca la marca de los cuatro cuadrados, una vez encontrada toma las coordenadas de dos cuadrados inferiores para centrar la imagen y acercarse lo suficiente para analizar el color y la forma del objeto. La muestra se toma de un segmento interno del objeto para poder crear la muestra correcta; si no se logra establecer una muestra correcta, el robot retrocede y vuelve a acercarse.

En ésta etapa el procesamiento de imágenes se utiliza para reconocer la marca de los cuatro cuadrados, y obtener las coordenadas de los mismos para enviar las instrucciones de movimiento y centrar la marca en cuadro de visión de la

cámara. Las coordenadas nos servirán para ubicar un rectángulo en el centro de los cuadros inferiores, que es donde se encuentra el objeto.

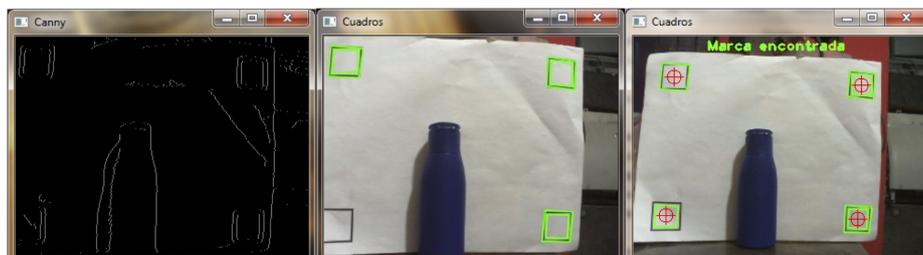


Fig. 2: Ejemplo del funcionamiento del uso de la plantilla usada y las marcas sobre la imagen capturada, para ubicar la posición del objeto de interés.

#### *Captura*

Se obtiene una imagen de la cámara en la configuración establecida la cual se utiliza para buscar la marca de los cuatro cuadrados.

#### *Pre-procesamiento*

Se aplican filtros, el primero es convertir la imagen a escala de grises, el segundo reduce la resolución de la imagen a la mitad y se vuelve a ampliar a la resolución original para eliminar ruido, y el tercero aplica el filtro Gaussiano [5] para suavizar la imagen y eliminar el mayor ruido posible.

#### *Segmentación*

La imagen filtrada es segmentada mediante el algoritmo de Canny [5], buscando reducir los datos de la imagen de interés con la información que nos interesa, para nuestro caso lograr una imagen binaria que muestra todos los bordes encontrados en color blanco y el fondo negro, preservando la información estructural importante en la imagen. Este algoritmo halla bordes con base a buscar máximos locales del gradiente de la imagen, calculándolo usando la derivada de un filtro Gaussiano, usa dos valores de umbral para localizar bordes fuerte o débilmente marcados en la imagen, y en esto está la ventaja de usar Canny en lugar de otros métodos ya que tiene buena detección ante ruido.

#### *Extracción de características*

En la imagen binaria se buscan contornos, ya que los cuadrados de la marca son figuras cerradas. De éste proceso se obtiene un conjunto de coordenadas que representan cada uno de los contornos encontrados en la imagen. Éste arreglo se procesa para encontrar figuras geométricas, que en este caso son cuadrados.

#### *Identificación de objetos*

Los contornos encontrados se analizan para detectar si son cuadrados o no, y almacena las coordenadas de cada uno de los cuadrados encontrados. Se dibuja un recuadro verde sobre la imagen original, resaltando cada cuadro encontrado. Cuando se detecta los cuatro cuadrados, se informa que se ha encontrado la

marca, y se guardan las coordenadas de los centros de cada uno de ellos. En la imagen original se dibuja la frase “Marca encontrada” y se resalta en rojo el interior de cada cuadro.

Una vez obtenida la marca, se identifican los dos cuadrados inferiores y se calculan dos datos: la distancia entre los centros y la coordenada  $x$  del punto medio de dicha distancia. El punto  $x$  nos sirve para centrar la imagen con respecto a la horizontal del campo de visión de la cámara. La distancia obtenida es utilizada para calcular la relación con respecto al ancho de la imagen y poder acercar el robot a la marca.

Los movimientos laterales que puede realizar el robot son determinados por la posición del punto  $x$  con respecto al centro de la imagen, moviéndose primero a la izquierda o a la derecha hasta centrarlo en un margen de error de 16 píxeles. Una vez centrado, avanza hacia adelante hasta que la relación calculada no sea menor a 0.73. Una vez centrado el robot y cerca de la marca se detendrá el sistema de movimiento y el proceso de ‘Búsqueda de Marca’, y se iniciará el análisis del objeto en la etapa ‘Detectar color y forma’.

#### **Detectar color y forma**

En esta etapa se dibuja un rectángulo en el centro de los cuadrados inferiores de 25 píxeles de lado el cual debe cubrir una sección interna de la imagen del objeto a analizar. De esta sección se toma la muestra de los colores de cada píxel y se usa para crear el histograma de colores y la máscara que ayuda a separar la imagen del objeto y el fondo de la imagen. Una vez separados se analiza la forma del objeto y se guardan sus valores característicos que sirven para identificar el objeto.

*Captura.* Se obtiene una imagen de la cámara y se utiliza las coordenadas de la sección para crear una segunda imagen que solo contenga la sección recortada.

*Pre-procesamiento.* Se reduce la resolución de la imagen a la mitad y se vuelve a ampliar a la resolución original para eliminar ruido. Creando una imagen llamada ‘corte’, de la imagen original a partir de la sección del rectángulo central. De la imagen ‘corte’ se obtienen los mínimos y máximos de los valores de tono, saturación y brillo (HSV) de cada píxel. Estos valores se usan para crear el histograma de colores y el filtro para la imagen original. La imagen original se transforma de color RGB al modelo HSV usando los valores mínimos y máximos obtenidos del corte, y se obtiene una imagen en color HSV. De la imagen HSV se crea una imagen binaria que es la primera máscara en la que los píxeles que se encuentran dentro del rango de mínimos y máximos que se representan en blanco, y los demás en negro. Se crea una imagen llamada hue (tinte) mezclando los canales tomados del corte de la imagen original, para realzar la tonalidad de colores del objeto y ayude a los cambios leves de iluminación. Se aplica un filtro a la imagen hue en base a los valores del histograma para eliminar el fondo, obteniendo una imagen en escala de grises que representa la máscara del color predominante del objeto. La imagen se convierte a binaria. Esta imagen se combina con la primera máscara y se obtiene otra con un mejor detalle del objeto. Ésta última imagen contiene un mayor detalle de la forma del objeto y los filtros aplicados ayudan a eliminar del fondo todo lo que no coincida con

la tonalidad del color. En la figura 3 se muestra un ejemplo del procesamiento digital aplicado al cuadro de imagen.

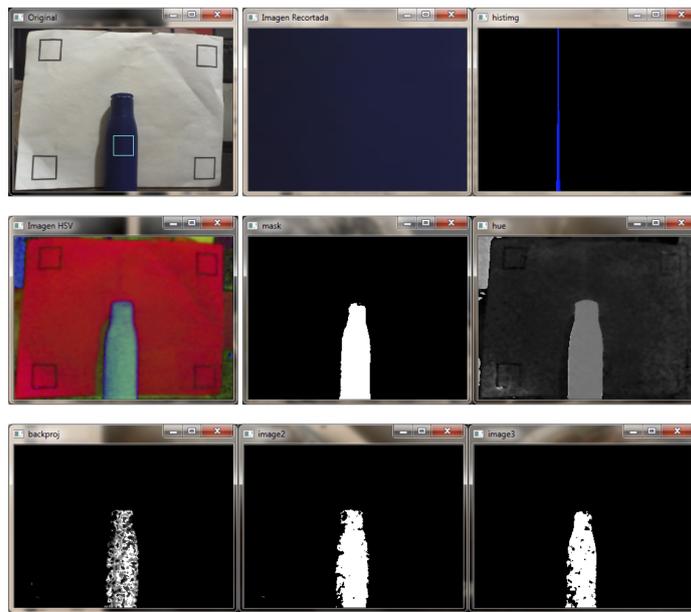


Fig. 3: Metodología propuesta para la detección y reconocimiento de objetos en un robot móvil diferencial.

*Segmentación.* En la imagen del paso anterior se buscan todos los contornos, obteniendo como resultado solo uno, el del objeto. Si no se encuentra ningún contorno quiere decir que hubo una mala muestra y se debe volver a tomar.

*Extracción de características.* Una vez detectados los contornos, se analizan en busca del contorno con mayor área, y se guarda en un vector que contenga todos los puntos del contorno del objeto. Este vector se guarda para ser comparado con el objeto a buscar en la siguiente etapa. También se guarda el histograma y los valores máximos y mínimos HSV para usarlos como filtro de color.

*Identificación de objetos.* El contorno guardado se utiliza para identificar el objeto en la siguiente etapa.

#### **Buscar y detectar el objeto**

Una vez completada la etapa 'Detectar color y forma', el robot gira a la derecha hasta perder de vista el objeto analizado y comienza la búsqueda de otro objeto con las mismas características. El robot se mueve lentamente a la derecha hasta que aparezca un objeto similar en el campo de visión de la cámara. Se utilizan los filtros obtenidos de la etapa anterior; por lo cual solo aparecerá en la imagen binaria cualquier objeto con la misma tonalidad que el objeto a buscar. Una

vez encontrado un objeto, se compara con el contorno guardado y si su forma es similar, se moverá el robot para centrarlo en la cámara y marcarlo como encontrado.

*Captura.* Se obtiene una imagen de la cámara.

*Pre-procesamiento.* Se utilizan los mismos filtros de la etapa anterior, tanto el histograma de colores como los mínimos y máximos valores de la muestra de color. De esta manera cualquier objeto que no pertenezca al rango de colores de la muestra, no se verá en la imagen filtrada, y el robot seguirá girando a la derecha. Un objeto de color similar aparecerá en ventana dependiendo su grado de coincidencia.

El robot seguirá girando a la derecha hasta que aparezca en la imagen un objeto del mismo color, cuando esto suceda, se guarda el contorno para que en la siguiente fase del procesamiento sea analizado.

*Segmentación.* De la misma manera que en la etapa anterior, se busca el contorno de la imagen para extraer sus características.

*Extracción de características.* Del contorno obtenido se extraen los puntos que lo conforman, incluyendo las coordenadas de su centro de masa.

*Identificación de objetos.* Se compara con el contorno del objeto a buscar por medio de sus momentos invariantes Hu [9], obteniendo un valor numérico de dicha comparación. Se coloca una señal en la imagen original, utilizando las coordenadas de su ubicación y el número obtenido de la comparación de contornos. El número representa la diferencia entre la similitud de ambas figuras; sí es mayor a 40 entonces son diferentes y entre más cercano a cero mayor el parecido. El carro seguirá girando a la derecha, pero deja la señal para indicar la mínima coincidencia. En la figura 4 se muestran cómo funciona la identificación de un objeto, aún ante presencia de oclusiones.

**Finalizar** Si encuentra un objeto similar, entonces inicia movimientos laterales para centrar la imagen y colocarse frente al objeto y así dejar de moverse. En este momento se da por encontrado el objeto.

## 2.2. Construcción del robot móvil diferencial con la cámara

Existen varios diseños de ruedas para elegir cuando se quiere construir una plataforma móvil sobre ruedas, para el desarrollo se utiliza el diseño diferencial. Se compone de 2 ruedas en un eje común, cada rueda se controla independientemente, puede realizar movimientos en línea recta, en arco y sobre su propio eje de contacto de rodamiento, requiere de una o dos ruedas adicionales para balance o estabilidad. Sencillo mecánicamente, puede presentar problemas de estabilidad y su cinemática es sencilla, para lograr el movimiento en línea recta requiere que las dos ruedas de tracción giren a la misma velocidad.

La implementación del sistema embebido se realizó integrando los componentes del proyecto, para que trabajen como un solo sistema. La parte del hardware está compuesto por la tarjeta de desarrollo Raspberry PI [2], La cámara web, la tarjeta de red inalámbrica y el sistema de movimiento. En la figura 5 se muestran imágenes de la construcción del robot móvil experimental aquí explicado.

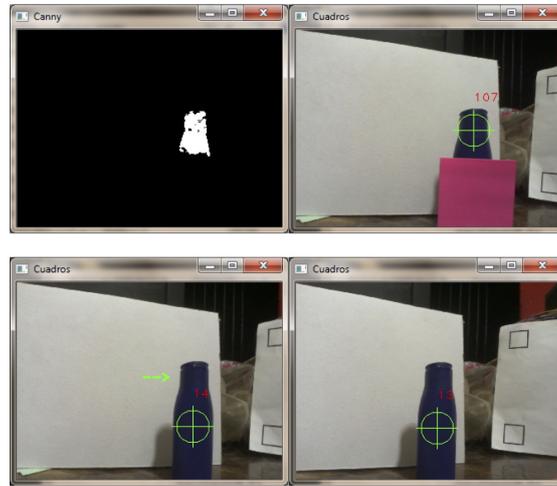


Fig. 4: Imágenes de monitoreo sobre el funcionamiento para la detección e identificación de un objeto.

La parte del software la integran el sistema operativo instalado en la tarjeta SD del Raspberry Pi, la librería de visión por computadora OpenCV [1], Servidor Web Lighttpd, el cual es parte del sistema de monitoreo, y el código fuente del programa que representa el sistema de procesamiento de imágenes.

Los pasos a seguir para la implementación son los siguientes:

- Se instala el sistema operativo Raspbian (Linux Debian Wheezy), la librería OpenCV 2.4.9, el servidor Lighttpd.
- Conectamos la cámara web y la tarjeta inalámbrica a los puertos USB de la Raspberry PI. La cámara Web y la tarjeta inalámbrica son compatibles con el sistema; por tanto no es necesario realizar una configuración adicional.
- Conectamos los puertos Gpio's al sistema de movimiento.
- Implementación del Sistema de monitoreo, se creó una página web con cuatro recuadros en los cuales se muestran las imágenes del sistema de procesamiento de imágenes. Estableciendo que cada una se actualice cada 200 ms.
- El sistema de procesamiento de imágenes se implementó dentro una carpeta, ubicada dentro de la tarjeta de desarrollo Raspberry PI, el cual se compila y ejecuta en el mismo sistema.

### 2.3. Sistema de monitoreo en tiempo real de la implementación

El sistema de monitoreo fue instalado en el dispositivo Raspberry Pi. El servidor Web Lighttpd se configuró en el puerto 80 para acceder por medio de cualquier navegador web escribiendo la dirección IP del Raspberry Pi, el cual

*Implementación de reconocimiento de objetos por color y forma en un robot móvil*



Fig. 5: Detalle de construcción del robot móvil diferencial.

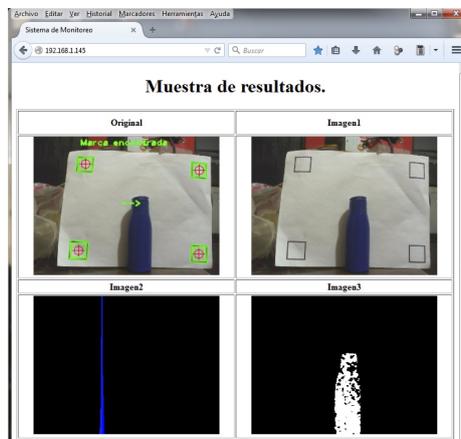


Fig. 6: Ventana del sistema de monitoreo del sistema, accesando con cualquier navegador Web en una red LAN.

representa la página web con cuatro imágenes que muestran los resultados del procesamiento que se lleva a cabo, como se muestra en la figura 6.

### 3. Resultados

En la figura 6 se muestran las pantallas de los resultados experimentales que se obtuvieron, con el reconocimiento esperado para condiciones de iluminación suficiente, a fin de lograr el nivel de detección deseado.

En la primera imagen (esquina superior izquierda) se muestra su color original con señales que destacan el resultado del procesamiento de imágenes, la segunda imagen (esquina superior derecha) muestra los cambios del proceso de filtrado, la tercera imagen (esquina inferior izquierda) muestra el histograma de colores del objeto analizado; y la cuarta (esquina inferior derecha) muestra el contorno guardado del objeto que será la base para buscar coincidencias.

Las imágenes se van actualizando cada 200 ms y son recibidas del programa principal en aproximadamente la misma cantidad de tiempo. La página solo proporciona imágenes, no interfiere en el funcionamiento del sistema.

#### **4. Conclusiones**

En este artículo presentamos una propuesta de fácil y rápida implementación, de bajo costo de la implementación de un algoritmo de visión por computadora en un sistema embebido. El sistema tiene operación autónoma, tomando las decisiones de movimiento del robot móvil con base al análisis y procesamiento de las imágenes capturadas desde su cámara web, teniendo un procesamiento fluido de 5 cuadros de imagen por segundo, logrando la detección de objetos en tiempo real. El modelo aquí propuesto se basa primero en obtener una muestra de color, y después su contorno. Los objetos que son mejores para éste sistema son aquellos que ofrecen texturas opacas.

A futuro planteamos continuar la mejora del sistema, refinando el algoritmo para diferentes propósitos. Adicionalmente el sistema puede mejorarse en su desempeño usando un sistema embebido con mayor capacidad de procesamiento.

#### **Referencias**

1. Adrian Kaehler, G.B.: Learning OpenCV. O'Reilly Media (2013)
2. Foundation, R.P.: Raspberry pi, <https://www.raspberrypi.org>
3. Gonzalez, E.: Study of behaviors subsumed with the generation of motor schemas in lego nxt 2.0. In: Engineering Applications (WEA), 2012 Workshop on. pp. 1–6 (May 2012)
4. Gonzalez, F., Guarnizo, J., Benavides, G.: Emulation system for a distribution center using mobile robot, controlled by artificial vision and fuzzy logic. Latin America Transactions, IEEE (Revista IEEE America Latina) 12(4), 557–563 (June 2014)
5. Gonzalez, R.C., Woods, R.E.: Digital Image Processing. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA (1992)
6. Humberto Sossa, Roberto A. Vazquez, R.B.: Reconocimiento y localización de instrumental médico usando análisis automatizado de imágenes. Revista mexicana de ingeniería biomédica 26, 75–85 (2005)
7. Kim, J.H., Lyou, J.: The development of an artificial vision based navigation system for helicopter using modified ndgps/ins integration. In: Control, Automation and Systems, 2007. ICCAS '07. International Conference on. pp. 2114–2118 (Oct 2007)
8. Malpartida, E.D.S.: Sistema de visión artificial para el reconocimiento y manipulación de objetos utilizando un brazo robot. Master's thesis, Escuela de Graduados, Pontificia Universidad Católica del Perú (2003)
9. Rao, A., Srihari, R., Zhang, Z.: Spatial color histograms for content-based image retrieval. In: Tools with Artificial Intelligence, 1999. Proceedings. 11th IEEE International Conference on. pp. 183–186 (1999)
10. Villanueva-Escudero, C., Villegas-Cortez, J., Zúñiga-López, A., Avilés-Cruz, C.: Monocular visual odometry based navigation for a differential mobile robot with android os. In: Gelbukh, A., Espinoza, F., Galicia-Haro, S. (eds.) Human-Inspired

- Computing and Its Applications, Lecture Notes in Computer Science, vol. 8856, pp. 281–292. Springer International Publishing (2014)
11. Zhang, R., Wang, L.: An image matching evolutionary algorithm based on hu invariant moments. In: Image Analysis and Signal Processing (IASP), 2011 International Conference on. pp. 113–117 (Oct 2011)