

## **APROXIMACIÓN A LA NAVEGACIÓN AUTÓNOMA DE UNA PLATAFORMA MÓVIL, MEDIANTE VISIÓN ESTEREOSCÓPICA ARTIFICIAL**

AN APPROACH TO AUTONOMOUS NAVIGATION OF A MOBILE PLATFORM USING A  
STEREOSCOPIC VISION

Adriana Riveros Guevara  
Ing. Mecatrónico, Asistente de Investigación Grupo GIDAM.  
Universidad Militar Nueva Granada, Bogotá, Colombia  
u1801128@unimilitar.edu.co

Cindy Natalia Salas López  
Ing. Mecatrónico, Ingeniera LID  
Elastómeros PVM, Bogotá, Colombia  
u1801115@unimilitar.edu.co

Leonardo Solaque Guzmán  
Ing. Electrónico, Ph.D., Profesor asociado, Facultad de Ingeniería,  
Investigador grupo GIDAM. Universidad Militar Nueva Granada, Bogotá, Colombia  
leonardo.solaque@unimilitar.edu.co

**Fecha de recepción:** 9 de julio de 2012

**Fecha de aprobación:** 3 de diciembre 2012

### **RESUMEN**

La visión artificial intenta capturar información relevante del medio ambiente, utilizando cámaras como sensores de ciertas características (formas, colores, texturas, etc.), para el funcionamiento adecuado de algunos mecanismos. Con el fin de obtener una imagen idéntica al entorno real, es necesario generar imágenes estereoscópicas que nos permita obtener la profundidad y así conseguir una representación en 3D. En este documento se plasma cómo se realizó la integración de un sistema de visión estereoscópica artificial a un robot móvil, con el objetivo de reconocer y seguir el centro de un camino. Dicho sistema se encarga de la adquisición, procesamiento y caracterización de imágenes, utilizando procedimientos fuera de línea para calibrar las cámaras y los métodos en línea para generar el mapa de disparidad, aplicación del operador Canny y la Transformada de Hough, empleando las librerías de OpenCV en un proyecto de Consola Win32 en C++. Es importante mencionar que los algoritmos desarrollados en este trabajo fundamentalmente son para ambientes estructurados. Los ambientes pueden clasificarse en: estructurados, no estructurados y semi-estructurados.

**Palabras clave:** visión estereoscópica, mapa de disparidad, OpenCV, Operador Canny, Transformada de Hough.

## ABSTRACT

An artificial vision tries to capture relevant information from environment using cameras as sensors of certain characteristics (shapes, colors, textures, etc.) for a proper functioning of some mechanisms. In order to get an identical image to real environment, it is required to generate stereoscopic images that allow us to get the depth and thus a 3D representation. This paper shows an artificial stereoscopic vision system incorporated to a mobile robot to recognize and follow the center of a path. Such a system handles the capture, processing and characterization of images using offline procedures such to standardize cameras and online methods for a disparity mapping generation, application of the Canny edge detector and the Hough transform, using the OpenCV libraries in Win32 Console project in C++ . It`s important to mention that algorithms developed in this work are fundamentally for structured environments. Environments can classified as follows: structured, non-structured and semi-structured.

**Keywords:** stereovision, disparity mapping, Open CV, Canny edge detector, Hough transform.

## INTRODUCCIÓN

En la actualidad, los sistemas de visión artificial se implementan en su gran mayoría, en aplicaciones estáticas para reconocer formas, colores y control de calidad; el acople de dichos sistemas a robots móviles, viene siendo tratado, enfrentando grandes complicaciones como cambios repentinos del entorno (luminosidad y obstáculos), y las diversas condiciones ambientales y de trabajo; sin embargo, se han hecho desarrollos importantes en este campo, en el ámbito local e internacional.

En la Universidad Militar Nueva Granada, se han desarrollado proyectos que aplican la robótica móvil y la visión artificial; uno de los trabajos más significativos es "Diseño y Construcción de un robot Scara con un sistema de visión de máquina, que pueda ser utilizado como herramienta didáctica en asignaturas de robótica y control" [1], que implementa un sistema de visión de máquina que consta de una cámara web (Creative Webcam Instant), acoplada a una base especialmente diseñada, con la cual se consigue una vista del área completa. Además, utiliza un marcador de posición para identificar la posición cero del espacio de trabajo.

En el ámbito regional, se encuentran proyectos como "Interceptor de trayectorias basado en visión artificial" [2], realizado en la Pontificia Universidad Javeriana utilizando la librería OpenCV, en donde se modifica la trayectoria de una pequeña bola, la información es tomada por una cámara y enviada para su posterior procesamiento a una tasa de 30 cuadros por segundo.

Los desarrollos internacionales se han dado principalmente en Europa, tras realizar tesis doctorales; un ejemplo de estas investigaciones lo vemos en "Navigation visuelle d'un robot mobile dans un environnement d'extérieur semi-structuré" [3], en donde se hace un desarrollo de la adquisición y

el procesamiento de imágenes desde el primer nivel que incluye la selección de cámaras, pasando por la corrección de las imágenes, segmentación por textura, y color, hasta llegar a los algoritmos que permiten la generación de trayectorias y movilización de un robot agrícola. Este trabajo se hizo en "l'Institut National Polytechnique de Toulouse". En "Lane Extraction and Tracking for Robot Navigation in Agricultural Applications" [4], se enfatiza el procesamiento de imágenes en la segmentación de colores; este procedimiento en muchas ocasiones, es suficiente para hacer la correcta separación del terreno por donde se va a navegar, y de los objetos del entorno que pueden tornarse como obstáculos. Además, se realiza la extracción de los caminos mediante señales de video y posicionamiento del robot móvil.

También se han realizado proyectos en Sudamérica; uno de ellos se describe en "Aplicación de Técnicas de Robótica e Inteligencia Artificial sobre un robot móvil utilizando el entorno de programación Visual Studio.Net" [5], en donde se utilizan las librerías OpenCV de Intel y el lenguaje de programación C bajo la plataforma .Net, para desarrollar un algoritmo que permita la detección de líneas y su posterior seguimiento. Se utiliza el filtro Canny para resaltar los bordes y se detectan dos pares de puntos que enmarcan la línea, lo cual permite el seguimiento de la misma.

Este artículo se desarrolla en las siguientes secciones: principio de funcionamiento de un sistema estereoscópico, elaboración de la plataforma de prueba, procesamiento de imágenes, integración del sistema de visión artificial a la plataforma móvil y finalmente, resultados experimentales y conclusiones.

## 1. PRINCIPIO DE FUNCIONAMIENTO DE UN SISTEMA ESTEREOSCÓPICO

Contextualizando lo que es el principio de función estereoscópica, a continuación se presentan algunos conceptos que conllevan a la selección de las cámaras y los algoritmos por usar.

En esta sección, se hará una breve descripción sobre qué es la visión artificial, qué ventajas tiene su utilización, se definirán algunos conceptos básicos y su representación matemática y además, se nombrarán algunos criterios que se utilizaron al escoger las cámaras que se usaron.

### 1.1. CONTEXTUALIZACIÓN TEÓRICA

A continuación, se muestra la definición de algunos conceptos que se deben tener en cuenta para realizar este proyecto:

**Visión artificial:** también llamada Visión por Computador, pretende capturar la información visual del entorno físico para extraer características visuales relevantes, utilizando procedimientos automáticos. Según Marr, "Visión es un proceso que produce a partir de imágenes del mundo exterior una descripción útil para el observador y no tiene información irrelevante" [6].

En la construcción de sus artefactos, el hombre ha imitado muchas veces, a la Naturaleza. En este caso, también ocurre. Las cámaras de vídeo con sus ópticas hacen las veces del globo ocular, mientras el computador realizará las tareas de procesamiento, emulando el comportamiento del cerebro. Las principales ventajas de la visión artificial respecto de la humana son:

- Es superior midiendo magnitudes físicas.
- Es más eficiente para la realización de tareas rutinarias.
- Es mejor en tareas de bajo nivel de proceso.

**Visión estereoscópica:** constituye un procedimiento para obtener la forma de los objetos en la escena. En este caso, la forma se determina mediante la distancia de los objetos en relación con un sistema de referencia, por lo cual se trata de un método para obtener la tercera dimensión [7].

La visión estereoscópica toma como referencia, el modelo estereoscópico biológico donde el desplazamiento relativo de los ojos permite obtener la profundidad de los objetos o tercera dimensión, mediante un simple proceso de triangulación a partir de las dos imágenes generadas por el mismo objeto de la escena 3D en cada ojo. Esto se debe al hecho de que los ojos están distanciados, y hace que las imágenes de los objetos en sendos ojos, se muestren desplazadas según la distancia entre los objetos y los ojos.

**Calibración de imágenes:** el proceso de calibración de imágenes trata de determinar los parámetros intrínsecos de las cámaras, tales como la distancia focal, el centro del plano sensor y la distorsión de la lente para poder luego, determinar la geometría de los objetos observados por la cámara [8].

Aunque existen distintos métodos para calibrar una cámara, el procedimiento básico es el mismo en todos ellos:

- Determinar con precisión un conjunto de puntos 3D del mundo exterior.
- Fijar sus correspondencias con las proyecciones de estos puntos 3D sobre la imagen proyectada 2D.
- Obtener mediante técnicas de optimización, la mejor solución para determinar los parámetros intrínsecos y extrínsecos.

**Correspondencia estéreo:** es el proceso mediante el cual dado un punto cualquiera de la escena 3D, se llega a determinar cuál es su proyección en sendas imágenes del par estereoscópico. La correspondencia constituye el principal problema dentro del proceso de la visión estereoscópica [8].

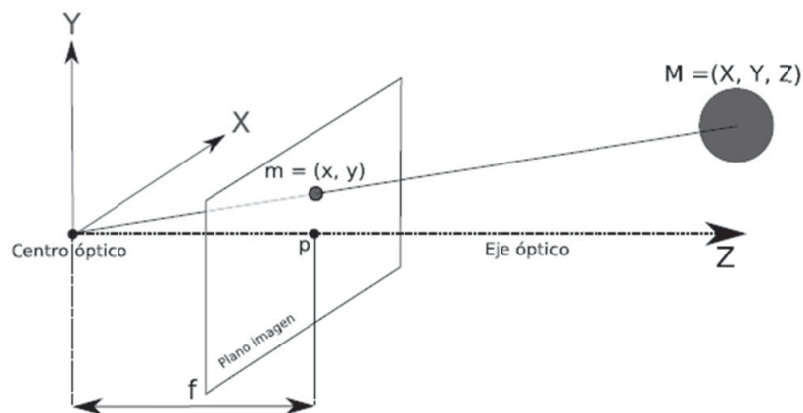
**Operador Canny:** es un algoritmo muy usado en la localización de contornos. Se caracteriza por evitar la ruptura de los bordes de los objetos. Su fundamento se basa en un proceso de optimización, teniendo en cuenta los siguientes objetivos por maximizar [6]:

- Aumentar la relación señal-ruido de la imagen.
- Disminuir todo lo posible, la distancia entre el borde detectado y el borde real.
- No identificar un borde por un único píxel, sino por un conjunto de píxeles que tengan una cierta conectividad.

**Transformada de Hough:** es un método utilizado para hallar líneas, que se expresan en coordenadas polares, y pueden ser detectadas encontrando el número de intersecciones entre curvas. En general, podemos definir un umbral del número mínimo de intersecciones necesarios para detectar una línea [6].

## 1.2. MODELO DE LA CÁMARA

El modelo pinhole asume que la cámara es una caja negra que tiene un pequeño agujero por donde ingresa una pequeña cantidad de luz; por este agujero se proyectan las imágenes de la superficie en forma inversa. El tamaño de la imagen está relacionado con la distancia del objeto que se ve reflejada por la distancia focal [9].



**Figura 1.** Modelo Pinhole.

**Fuente:** Tarlea Jiménez J. [9].

Cuando el plano de la imagen está ubicado por delante del centro óptico, se obtiene una proyección de la imagen sin inversión. En la Figura 1, se observa el modelo Pinhole cuando P (plano de la imagen) está delante de C (centro óptico).

- El centro óptico C es el punto de la lente donde cualquier rayo de luz que pasa por él, no sufre desviación.
- El eje óptico es la línea imaginaria que parte del centro óptico y corta perpendicularmente el plano imagen.

- La distancia focal  $f$  es la distancia que existe desde el centro óptico al plano focal.
- El plano imagen o plano focal se sitúa en  $Z = f$ . Es el plano virtual donde se forma la imagen sin inversión alguna
- El punto principal  $p$ , es la intersección del eje óptico con el plano imagen.

### 1.3. SELECCIÓN DE LAS CÁMARAS

Se contemplaron diferentes opciones de cámaras, como se puede observar en la Tabla 1, teniendo en cuenta la calidad de la imagen, velocidad y costo, así como el tipo de comunicación; por último se optó por cámaras web de alta definición LifeCam HD-5000 de Microsoft, ya que su relación costo/beneficio era el mejor.

**Tabla 1.** Cámaras contempladas

Cámara	Ventajas	Desventajas
SVS	Comunicación inalámbrica, librerías para el sistema	Pérdida de la comunicación y cambio en la tasa de recepción, resolución de 640x840 pixeles, tiempo de adquisición de 10 fps
Microsoft LifeCam HD-5000	Bajo costo, corrección de luz, tiempo de adquisición de 30 fps, resolución de 1280x720 o más pixeles	Comunicación alámbrica, autofocus
Webcam Logitech HD PRO C910	Alta calidad en la imagen, costo moderado, resolución de 1280 x 720 o más pixeles, tiempo de adquisición de 25 fps	Dificultad en el acople de las cámaras

## 2. PLATAFORMA DE PRUEBA

En esta sección, se hará una descripción de cómo se realizó el diseño de la plataforma móvil y de cuáles fueron los criterios utilizados para elegir los componentes electrónicos y mecánicos que fueron empleados.

### 2.1 ARQUITECTURA MECÁNICA

El sistema de visión escogido hace necesario que el computador esté embarcado en el robot móvil; por esta razón, el diseño se realizó partiendo de la idea de que el computador fuera parte integral del diseño. Las dimensiones básicas también fueron escogidas con base en las medidas del computador que va embarcado en el robot, siendo este un SONY VAIO PCG-5121 1L de 13.3".

El diseño CAD se hizo en Solid Works 2011. Se elaboró el prototipo del soporte para las cámaras, teniendo en cuenta que la distancia horizontal fuera la deseada (10cm-11cm), y garantizando

la altura desde el piso a las cámaras; esta medida se determinó con base en pruebas donde se tomaron fotos a distintas longitudes, observando que el rango de altura en donde mejor se comportaba, era de 35 cm a 45 cm. Las pruebas que se realizaron para encontrar este rango de operación, consistieron en elevar 5 cm el sistema estéreo; cuando las cámaras estaban muy bajas, se veía el suelo, pero no se podía distinguir con claridad los bordes del mismo; sin embargo cuando se subían mucho, aumentaban las vibraciones de la cámara y se generaba un mayor torque, condiciones no recomendables que generan errores significativos en la secuencia de imágenes.

**Tabla 2:** Características servomotores

<b>Peso</b>	<b>55 g</b>
<b>Dimensiones</b>	40.7x19.7x42.9
<b>Torque máximo</b>	12 kg/cm
<b>Velocidad</b>	0.25 s (4.8v), 0.20 s (6v)
<b>Voltaje</b>	4.8-7.2 V
<b>Rango temperatura</b>	0-50°C

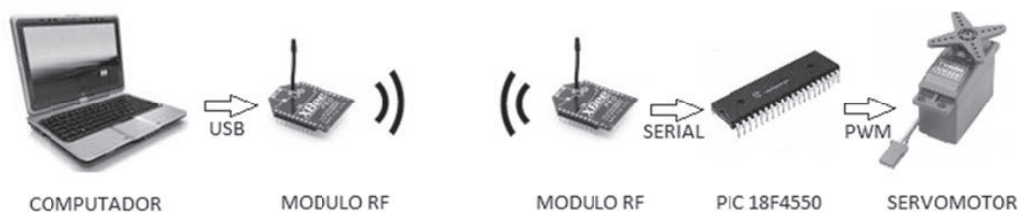


**Figura 2.** Diseño CAD

El tipo de locomoción es con ruedas en una configuración diferencial, donde todas las llantas son motrices para dar mayor capacidad de carga y superar inconvenientes como la rugosidad y desniveles del terreno; además de ello, este tipo de locomoción permite ejercer un control más sencillo, menos consumo de potencia y la posibilidad de adaptar un sistema odométrico, lo cual hace que este sistema sea el mejor para esta aplicación en particular. El peso aproximado considerado fue de 10 Kg, por lo cual se escogieron servomotores Tower Pro MG945 con características descritas en la Tabla 2. En la Figura 2A se muestra el diseño final CAD.

## 2.2 ARQUITECTURA ELECTRÓNICA

Al tener como base cuatro servomotores, el sistema mecánico, obliga a que la electrónica genere señales PWM para su manejo, y teniendo en cuenta que la comunicación con el computador se va a realizar de forma inalámbrica utilizando sistemas como el Xbee, se hace necesario el enlace serial entre la parte electrónica y el dispositivo RF; este sistema debe tener un costo reducido, pero generar buenos resultados, por lo cual se escogió el PIC18F4550 para su implementación, ya que cumple las características mencionadas y además, permite posteriores ampliaciones para diferentes aplicaciones.



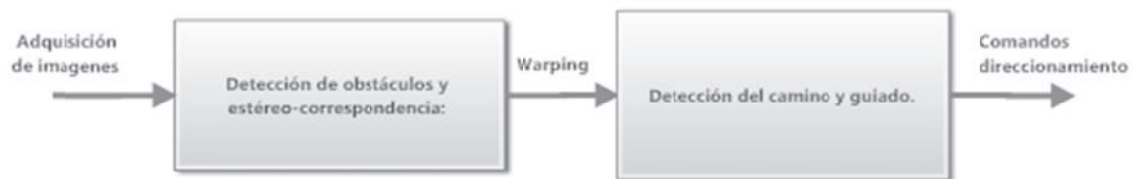
**Figura 3.** Arquitectura electrónica



El programa del microcontrolador recibe los caracteres 'a', 'd', 'i', 'b' y 's' para los movimientos hacia adelante, a la derecha, a la izquierda, hacia atrás y parar respectivamente. Según estas señales, se indica qué motor debe moverse en sentido horario o antihorario para generar el desplazamiento deseado. En la Figura 3, se muestra un esquema del funcionamiento de la arquitectura electrónica del sistema.

### 3. PROCESAMIENTO DE IMÁGENES

Esta sección muestra los diferentes subprogramas que fueron realizados para elaborar el software encargado de direccionar la plataforma de pruebas concebida. Estos subprogramas se agruparon en dos tareas principales: detección de obstáculos y estéreo-correspondencia y Detección del camino y guiado; sin embargo, existen programas auxiliares como Adquisición de imágenes, Sobreposición o Warping y Comandos de direccionamiento que cumplen la tarea de integrar las etapas primarias, como se puede observar en la Figura 4.



**Figura 4.** Etapas principales de software

Los programas auxiliares y el software principal se hicieron en Visual Studio 2010 como aplicación de consola win32, en lenguaje C++, utilizando las librerías de OpenCV en sus versiones 2.1 y 2.3.

#### 3.1 DETECCIÓN DE OBSTÁCULOS Y ESTÉREO-CORRESPONDENCIA

Para llevar a cabo este procedimiento, es necesario calibrar las cámaras, teniendo en cuenta los siguientes pasos [10]:

- Remover las distorsiones radiales y tangenciales de los lentes, para obtener imágenes no distorsionadas.
- Ajustar los ángulos y distancias entre las cámaras para realizar la rectificación, y generar imágenes alineadas y rectificadas.
- Encontrar las mismas características en la cámara izquierda y derecha. Este proceso se denomina correspondencia. Aquí, se genera un mapa de disparidad que se forma con la diferencia de las "x" entre las imágenes de las dos cámaras; no se toman diferencias en "y", porque con la rectificación, esa diferencia es 0.
- Si se conocen los parámetros de las cámaras, se puede hallar distancias mediante triangulación, para generar así mapas de profundidad. Este proceso se denomina reproyección.



Posterior a este proceso, para hallar un valor numérico de la distancia en donde se encuentra un obstáculo, se hace un promedio de los valores que forman el mapa de disparidad; dicho valor se reemplaza en la ecuación de triangulación, para generar la longitud hasta el objeto determinado. Esta distancia se utiliza para darle instrucciones al robot para que se detenga o gire.

Para realizar la estéreo correspondencia, se encuentran en las imágenes de derecha e izquierda, los puntos de homografía y después, se sobreponen las imágenes para generar una visión más amplia que permita al robot moverse adecuadamente.

### **3.2 DETECCIÓN DEL CAMINO Y GUIADO**

Después de revisar la bibliografía que existe sobre el tema, se determinaron los siguientes pasos para lograr el objetivo trazado:

- Transformada geométrica de la imagen por medio de la cual se obtiene una imagen en donde se ha eliminado el efecto de la perspectiva.
- Extracción de características; en este caso, la aplicación del filtro Canny con el fin de obtener los bordes.
- Transformada de Hough que halla las posibles líneas correspondientes al borde del camino.
- Extracción de líneas que corresponden al lado derecho e izquierdo del camino.
- Obtención del centro del camino, encontrando el punto medio entre las líneas halladas en el paso anterior.
- Guiado del móvil con base en el cálculo del cambio del ángulo de la línea central.
- Comunicación serial para enviar parámetros de guiado por medio de Xbee.

En la Figura 5, se muestra un diagrama general de los procesos llevados a cabo; el recuadro superior corresponde a la etapa principal de Detección de obstáculos y estéreo-correspondencia, el inferior muestra la Detección del camino y guiado, en cada división están los procesos correspondientes a los subprogramas ya mencionados, cada programa auxiliar se explica más adelante.

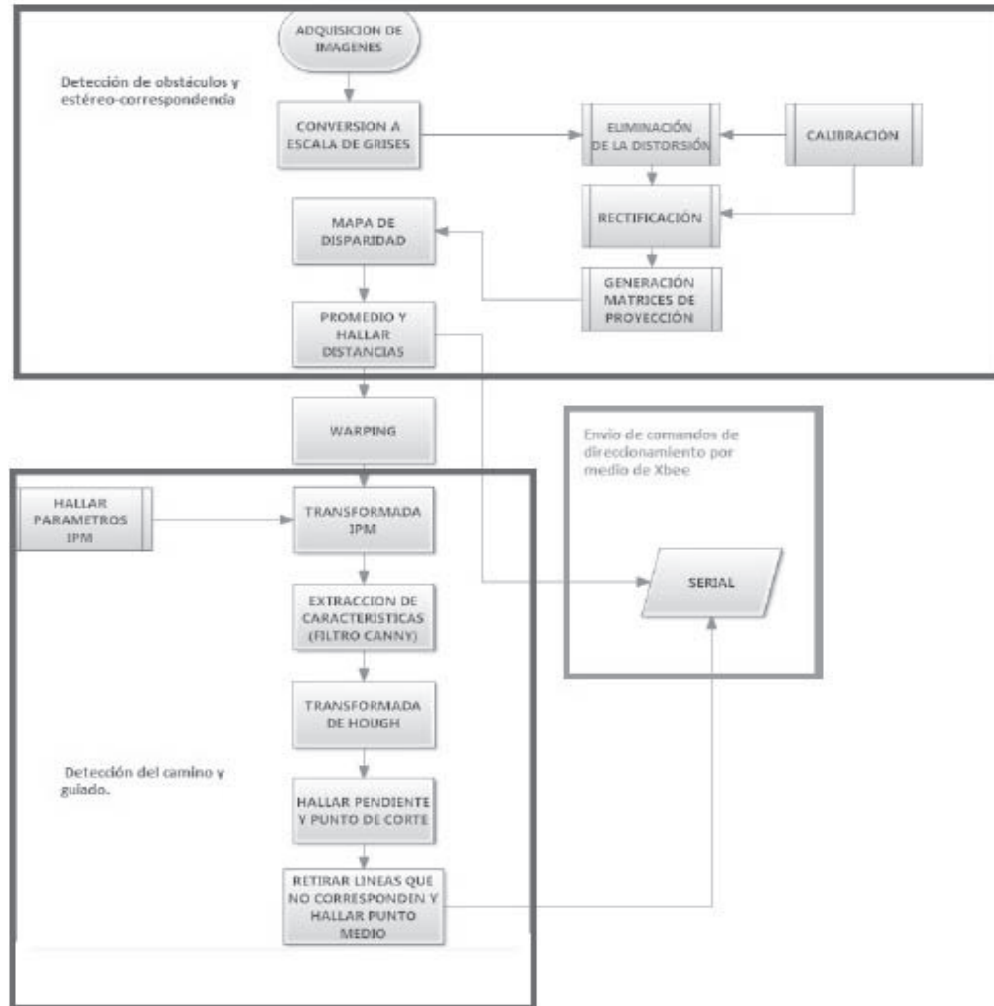
### **3.3. SUBPROCESOS Y PROGRAMAS AUXILIARES**

A continuación, se explica cómo se realizó cada subprograma, algunas funciones que se usaron de la librería OpenCV y se relacionan los resultados obtenidos.

#### **3.3.1. Generación del mapa de disparidad**

Para generar el mapa de disparidad, se leen los archivos .xml y se crea una imagen no distorsionada y rectificadas, utilizando la función `cvRemap()`; con las nuevas imágenes se ejecuta la función `cvFind Stereo Correspondence ()` que tiene como parámetros las dos imágenes rectificadas en escala de grises, el tipo de algoritmo, la máxima disparidad, etc. Tras ejecutar la función, se genera un

arreglo con las dimensiones de la imagen y valores que oscilan entre 0 y 255 que corresponden a la disparidad; este valor es inversamente proporcional a la profundidad, por lo cual haciendo un promedio entre los valores que más se repiten y que están más cercanos a 255, se puede hallar la disparidad del objeto que se encuentre más próximo y por lo tanto, a qué distancia se encuentra el robot de un obstáculo. En la Figura 6, se muestra un mapa de disparidad generado en uno de los escenarios.



**Figura 5.** Diagrama de procesos del sistema propuesto.



**Figura 6.** Mapa de disparidad: es un mapa generado por la diferencia de ubicación del mismo punto observado en las dos cámaras. Si el color del pixel es más blanco, indica que el objeto está más cercano (en la imagen mostrada se invirtieron los colores para mejorar la visualización del lector; sin embargo, por requerimientos de la revista, todas las imágenes de este documento se encuentran en escala de grises).

### 3.3.2. Cálculo de profundidad

El cálculo de la profundidad se hizo a partir de la ecuación 1, resultante cuando se dio la triangulación:

$$z = f \frac{b}{d} \quad (1)$$

En donde  $b$ , es la distancia que hay entre cámaras; en este caso 10.5 cm,  $f$  es la distancia focal que se ve reflejada tras la rectificación y tiene un valor de 338.43 mm y  $d$  es la disparidad que está siendo hallada en el proceso anterior, con el promedio de los valores del mapa de disparidad.

### 3.3.3. Sobreposición de imágenes

Este procedimiento se basó en dos imágenes tomadas del par de cámaras del sistema estéreo. A estas imágenes se les hace una transformación a escala de grises para que pueda ser tratada por las funciones encontradas en las librerías de OpenCV. Allí se utiliza la función `cvFindHomography()` que recibe como parámetros, la localización de los puntos de homografía obtenidos con la elección de los “keypoints” de la imagen. En la Figura 7, se observan los “keypoints” correspondientes a las imágenes izquierda y derecha, respectivamente.



**Figura 7.** Puntos de homografía: esta imagen muestra los puntos de homografía encontrados en las imágenes adquiridas por las cámaras izquierda y derecha, respectivamente; estos puntos se ubican donde existe un gran contraste de color o textura

Luego de haber ejecutado la función, se sobreponen dichas imágenes y se obtiene la vista panorámica deseada. Para encontrar la correspondencia entre las imágenes, se emplea el método de RANSAC (RANdom SAmple Consensus) [11].

La matriz de homografía hallada también es usada en el procesamiento online como base para la unión de las imágenes que están siendo capturadas por la cámara.

### 3.3.4. Redimensión

La detección de obstáculos y del camino se realizó de forma paralela; el hallazgo del camino se ejecutó con base en videos tomados previamente con una resolución de 1280X720 a 15 fotogramas/segundo. Sin embargo, se hizo necesario redimensionar la imagen a 426X320 pixeles, ya que con este tamaño, se hallaron las matrices necesarias para el proceso alterno (detección de obstáculos y estéreo-correspondencia).

Para hacer el procesamiento más rápido, se retira la parte superior de la imagen porque no brinda información útil y sí puede generar errores en el posterior procesamiento, por lo cual se queda únicamente con la región de interés que tiene un tamaño de 426x210 pixeles. La nueva imagen es guardada para ser utilizada en los programas auxiliares que permiten establecer los valores necesarios para la transformación geométrica, lo mismo que la correspondencia entre distancia y pixeles.

### 3.3.5. Transformación geométrica

El ángulo de vista bajo el cual se adquiere la escena y la distancia de los objetos desde la cámara (nombrado el efecto de perspectiva), contribuyen a asociar contenido diferente de información

a cada pixel de la imagen. La vista en perspectiva de la imagen original, tiene varias desventajas como la variación del ancho de las marcas y sobretodo, una escala de distancia no lineal.

Para solucionar este problema, se realizó una transformación geométrica llamada Inverse Perspective Mapping (IPM), que permite remover el efecto de la perspectiva. La aplicación de la IPM requiere un conocimiento de las condiciones específicas de adquisición (posición de la cámara, orientación, óptica), y algunas suposiciones de la escena, así que la IPM puede ser usada en ambientes estructurados donde por ejemplo: la cámara está montada en una posición fija o en situaciones donde la calibración del sistema y el ambiente pueden ser sensados.



**Figura 8.** Transformación geométrica: en la parte izquierda de la imagen, se puede ver la imagen original que ven las cámaras. La parte derecha representa la utilización del subprograma y los resultados al aplicar la IPM.

En esta etapa, se utilizó el programa auxiliar prubirdseye, el ángulo  $\alpha$ , la distancia focal y las distancias en "y" y "z"; que varían mediante trackbars, dando como resultado los siguientes valores:  $\alpha = -84^\circ$   $f=350$   $y= -11$   $z= 157$ ; el valor más susceptible a cambiar el resultado es el ángulo  $\alpha$ , y por ello, para cualquier cambio de inclinación debe ser variado; aquí existe un gran inconveniente porque un terreno irregular lo lleva a cambiar demasiado, por lo cual este valor tendría que ser calculado dinámicamente o el vehículo llevar un sistema de suspensión lo suficiente bueno para evitar los sobresaltos. En la Figura 8, se muestran los resultados de la IPM.

### 3.3.6. Detección del camino

Luego de haber obtenido los valores para obtener la transformada geométrica y la relación entre pixeles y distancia, se trabaja con una imagen de prueba a la cual se le aplica el filtro Canny.

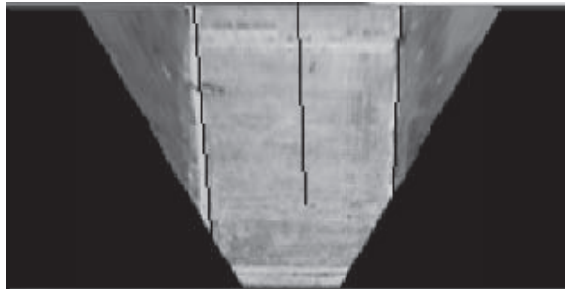
**Filtro Canny.** Por medio de la función `void cvCanny(const CvArr* image, CvArr* edges, double threshold1, double threshold2, int aperture_size=3)`, donde `image` es la imagen de entrada, `edges` la imagen de salida, ambas de un canal, `threshold1` y `threshold2`, es el primer y segundo umbral correspondientemente. Por último, `aperture_size` son los parámetros de apertura del operador sobel.

El menor valor entre `threshold1` y `threshold2` se utiliza para vincular los bordes, el mayor valor se emplea para encontrar los segmentos iniciales de los contornos. Los resultados de aplicar el filtro Canny se pueden ver en la Figura 9.



**Figura 9.** Filtro Canny (en la imagen mostrada se invirtieron los colores para mejorar la visualización del lector)

**Transformada de Hough.** Luego de aplicar el filtro de Canny, se utiliza la transformada de Hough por medio de la función en OpenCV `houghlines2`. Esta función devuelve una secuencia de puntos iniciales y finales correspondiente a cada línea detectada; con el fin de obtener sólo aquellas líneas que corresponden al borde del camino, se debe eliminar las líneas horizontales, además de hallar un promedio entre las demás líneas para obtener una sola que corresponda al borde. Para poder realizar el agrupamiento, se encuentran la pendiente y el punto de corte, y se plantea una relación de semejanza; es así como se promedian estas líneas parecidas y se encuentran los dos bordes del camino.



**Figura 10.** Transformada de Hough

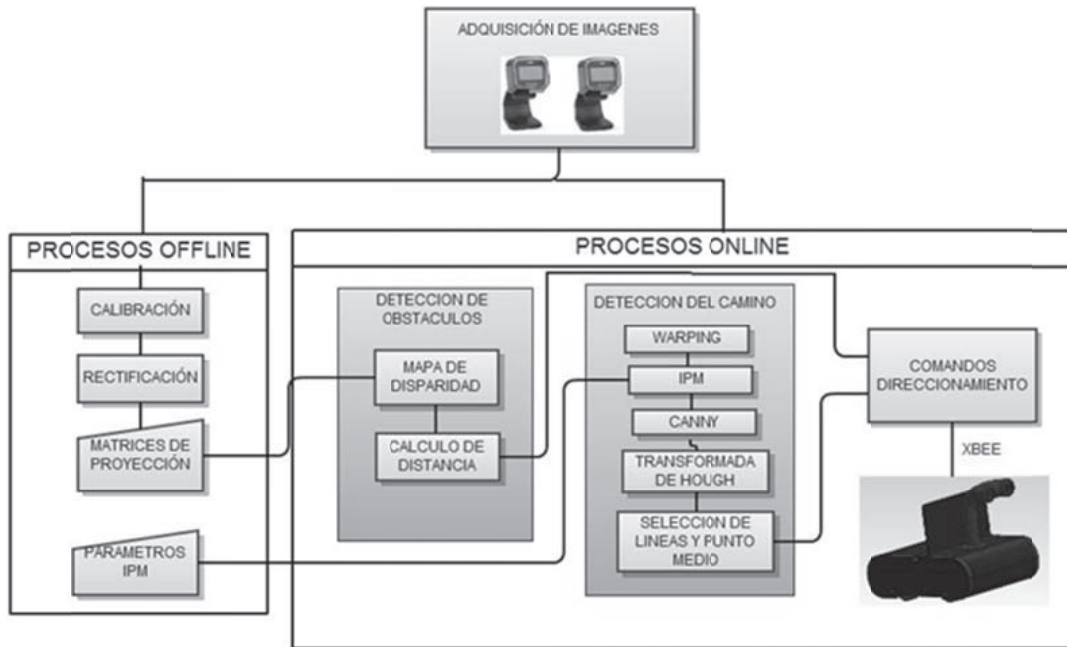
Con base en estas dos líneas previamente halladas, se determina su punto medio que corresponderá con el centro del camino. Los segmentos hallados y su línea central pueden verse en la Figura 10.

### 3.3.7. Etapa de detección de camino y guiado

Con base en los programas auxiliares y de prueba en donde se procesa sólo una imagen, aplicando las diferentes transformaciones y cálculos, se crea un nuevo programa en donde se manipula un video para guiar el móvil por el centro del camino previamente detectado; se determina el valor del pixel en x, correspondiente al centro de la imagen, y comparando con el valor medio de las coordenadas en x de la línea generada en el paso anterior, se va indicando al robot móvil mediante comandos enviados por Xbee, si debe moverse hacia la derecha o izquierda o continuar avanzando.

#### 4. INTEGRACIÓN DEL SISTEMA DE VISIÓN ARTIFICIAL A LA PLATAFORMA MÓVIL

Para que la plataforma móvil pudiera desempeñarse adecuadamente, fue necesario realizar la integración de los dos programas principales y enviar las instrucciones de movimiento del software mediante el Xbee para que el robot generara los desplazamientos correspondientes.



**Figura 11.** Integración de los programas principales

La Figura 11 representa la integración de software y el hardware de la plataforma móvil. Existen programas que no requieren ejecución permanente porque no se ven modificaciones en sus comportamientos y su tiempo de depuración es largo, por lo cual se compilaron offline, mientras que hay unos que necesitan de su constante actualización o programas online. Los subprogramas realizados tanto online como offline, se separaron en dos temas fundamentales: detección de obstáculos y detección del camino, los cuales fueron explicados detalladamente en la sección anterior.

La parte de detección de obstáculos tiene prioridad sobre la de detección del camino; si se presenta una oclusión en el camino, el móvil se detendrá hasta que dicho obstáculo sea retirado; se realizó así ya que el camino del escenario de prueba es muy delgado como para evitar el obstáculo y continuar.



## 5. RESULTADOS Y ANÁLISIS

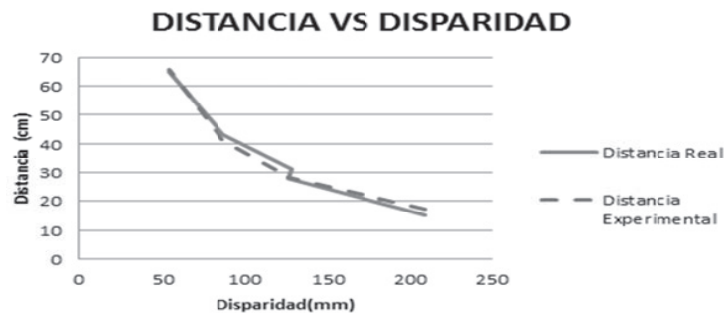
Las pruebas se hicieron con base a 13 escenarios en donde se grabaron videos correspondientes a la imagen derecha e izquierda, a los cuales se les aplicaron los algoritmos desarrollados, con el fin de comprobar el buen funcionamiento del programa desarrollado.

Tras realizar el proceso de calibración y rectificación, se halló la profundidad a diferentes distancias, obteniendo los datos de la Tabla 3. Al encontrar y promediar el error absoluto, el resultado es de 1.57 equivalente al 5.9%.

**Tabla 3.** Distancia y disparidad

Disparidad (mm)	Distancia real (cm)	Distancia Exp. (cm)	Error absoluto
209.03	15	17.00	2.00
126.10	28	28.18	0.18
127.77	31	27.98	3.19
86.64	43	41.01	1.99
54.23	65	65.52	0.52

Al realizar las gráficas correspondientes de Distancia vs Disparidad representadas en la Figura 12, se observa que tienen una relación inversamente proporcional.



**Figura 12.** Distancia vs Disparidad (gráfica generada en Excel)

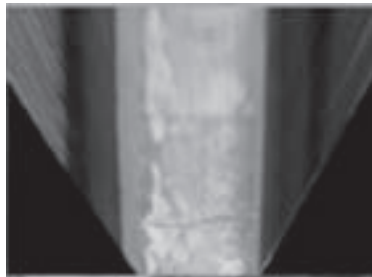
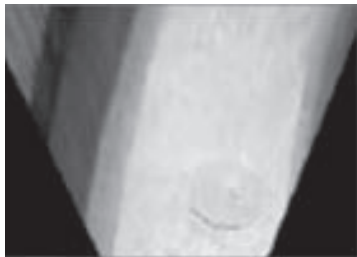
El cálculo de la distancia se ve limitado, si hay una superficie muy regular porque al estar viendo entornos similares, no se encuentra diferencia alguna por lo cual no se genera un mapa de disparidad adecuado, por esto el robot presenta fallas en la evasión de obstáculos con objetos grandes con textura y color uniforme, tales como paredes.

Se realizó el Warping de las imágenes, cuyas capturas son utilizadas como entrada en el programa Bird's eye que se encarga de hallar los parámetros adecuados para eliminar el efecto de la perspectiva.

**Tabla 4.** Parámetros de la transformada geométrica (Bird's eye).

Esenario	Ángulo		Distancia focal	Distancia z	Distancia y	
esc1	6	-84	350	277	522	22
esc2	10	-80	327	270	528	28
esc3	7	-83	350	358	553	53
esc4	7	-83	283	233	503	3
esc5	8	-82	350	270	528	28
esc6	6	-84	350	314	535	35
esc7	6	-84	350	314	535	35
esc8	6	-84	350	314	535	35
esc9	5	-85	421	327	516	16
esc10	5	-85	350	277	503	3
esc11	6	-84	350	314	541	41
esc12	7	-83	308	289	528	28
esc13	5	-85	352	239	491	-9

Los parámetros varían debido a diferentes factores como los cambios de inclinación, las irregularidades y el ancho del terreno, además de la posición inicial del robot; en la Tabla 4, se muestran dichos parámetros obtenidos de 13 escenarios diferentes de pruebas. En la Figura 13, se muestran los resultados de los escenarios que mostraron los comportamientos de éxito y de falla más notorios tras la ejecución del programa.

**Escenario 1****Resultado del Escenario 1****Escenario 2****Resultado del Escenario 2**



**Escenario 4**



**Resultado del Escenario 4**

**Figura 13.** Pruebas del sistema en algunos escenarios del programa.

El escenario 4 se puede considerar un escenario de falla debido a los cambios de inclinación repentinos, así como pasar de un campo abierto a un puente, lo cual hace que el reconocimiento de bordes no funcione de la manera deseada. Sin embargo, los escenarios 1 y 2 identifican en forma adecuada, los bordes y el centro del camino, por lo cual logran el objetivo de dirigir (enrutar), apropiadamente la plataforma.

## 6. CONCLUSIONES

Los algoritmos de detección de camino presentan dificultades por superar, como cambios de iluminación, terrenos irregulares y ambientes poco estructurados, reflejados principalmente en la implementación del programa en espacios interiores. A pesar de ello, se tiene una tasa de éxito significativa, obteniendo un error del 5.9% para encontrar obstáculos y alcanzando éxito en nueve de los 13 escenarios escogidos en la detección y seguimiento del camino, lo cual representa aproximadamente el 70% de resultados satisfactorios.

Para realizar el algoritmo, fue necesario usar el Operador Canny y la Transformada de Hough entre otros; estas funciones otorgaron buenos resultados en la mayoría de ocasiones; sin embargo, la extracción de bordes tenía inconvenientes con terrenos de pocos contrastes o curvas, por lo cual la implementación del agrupamiento K-means mejoró el reconocimiento de los límites del camino; no obstante, dicho algoritmo empleó un tiempo de cómputo alto que no justificaba el beneficio otorgado.

El cálculo de profundidad mediante mapas de disparidad, es exitoso en casos en los cuales se vea con claridad la diferencia entre la imagen obtenida por la cámara izquierda y la cámara derecha; estos resultados muestran un error de cerca del 6% y verifican la relación inversamente proporcional que existe entre la distancia y la disparidad.

**REFERENCIAS BIBLIOGRÁFICAS**

- [1] Jiménez Robayo C.E., (2008). Diseño y construcción de un robot Scara con un sistema de visión de máquina, que pueda ser utilizado como herramienta didáctica en asignaturas de robótica y control. Trabajo de Grado (Ingeniero en Mecatrónica). Facultad de ingeniería, Universidad Militar Nueva Granada. Bogotá.
- [2] Martínez Mejía D.A., (2005). Interceptor de trayectorias basado en visión artificial. Trabajo de Grado (Ingeniero Electrónico). Facultad de Ingeniería, Pontificia Universidad Javeriana. Bogotá. 103 p.
- [3] Aviña Cervantes J.G., (2005). Navigation visuelle d'un robot mobile dans un environnement d'extérieur semi-structuré. Toulouse, Ph.D. Thèse, L'Institut National Polytechnique de Toulouse. 202 p.
- [4] Aviña Cervantes J.G.; Deby M.; Hernández Marín A., (2003). Lane extraction and tracking for robot navigation in agricultural applications. In: Proceedings of ICAR 2003: The 11th International Conference on Advanced Robotics. Coimbra, Portugal.
- [5] Verrastro C. y Barneda R., (2006). Aplicación de técnicas de robótica e inteligencia artificial sobre un robot móvil utilizando el entorno de programación visual studio.net Módulo 3: Desarrollo de algoritmo de detección de líneas mediante visión artificial y control de trayectoria. Universidad Tecnológica Nacional. Facultad Regional. Grupo de Inteligencia Artificial y Robótica. Buenos Aires.
- [6] Automática e Informática Industrial de la Universidad Politécnica de Madrid Departamento de Electrónica. Apuntes visión artificial.
- [7] Guerrero Hernández J.M., Pajares Martinsanz G., y Guijarro Mata-García M., Técnicas de procesamiento de imágenes estereoscópicas. Departamento de Ingeniería de Software e Inteligencia Artificial. Universidad Complutense de Madrid.
- [8] Universidad Politécnica de Madrid. Prácticas de robótica y visión artificial.
- [9] Tarlea Jiménez J., (2009). Sistema de posicionamiento de objetos mediante visión estéreo embarcable en vehículos inteligentes. Trabajo de Grado (Ingeniero en Telecomunicaciones). Facultad de Ingeniería. Universidad de Alcalá, España, 104p.
- [10] Bradsky G., and Kaehler A., (2008). Learning OpenCV: Computer Vision with the OpenCV library.
- [11] Davies E.R., (2012). Computer and Machine Vision: Theory, Algorithms, Practicalities.

