

IMPLEMENTACIÓN DE LA TRANSFORMADA DE HOUGH PARA LA DETECCIÓN DE LÍNEAS PARA UN SISTEMA DE VISIÓN DE BAJO NIVEL

RESUMEN

En este artículo se presenta, un trabajo desarrollado en la detección de líneas para sistemas de visión de bajo nivel. Se muestran los resultados obtenidos en dicho trabajo, y su aplicación en el tratamiento digital de imágenes (preprocesamiento y descripción) con un microcontrolador de la familia MCS-51.

PALABRAS CLAVES: microcontroladores, lenguaje ensamblador, visión artificial, procesamiento de imágenes, transformada de Hough.

ABSTRACT

This paper presents a work developed for the detection of lines in a low level vision system. The results obtained in such work are shown as well as their application to digital image processing (preprocessing and description) using a MCS-51 microcontroller's family.

KEYWORDS: *Microcontrollers, assembler language, artificial vision, image processing, Hough Transform.*

JUAN PABLO URREA

Ingeniero Electrónico
Profesor Auxiliar, Departamento de Matemáticas.
Universidad Tecnológica de Pereira
jpurrea@utp.edu.co

EMMANUEL OSPINA

Ingeniero Electrónico
Profesor Auxiliar, Departamento de Ingeniería de Sistemas y Computación.
Universidad Tecnológica de Pereira
emmanuel_o@utp.edu.co

1. INTRODUCCIÓN

En el área de robótica es necesario tener sistemas de visión de bajo costo para identificar objetos primarios, como lo son figuras geométricas, líneas, etc., que permitan seguir trayectorias o evadir obstáculos. El procesamiento de imágenes proporciona algoritmos con los que se pueden identificar líneas, uno de ellos la transformada de Hough, la cual necesita cálculos de punto flotante. En la práctica no se dispone de un sistema económico con esta opción. Una solución es utilizar un microcontrolador de la familia MCS-51 el cual es un sistema de bajo costo, que permite realizar algoritmos con una precisión menor pero con resultados aceptables para el propósito que se busca.

En este trabajo se desarrolla un sistema de visión de bajo nivel, para la detección de líneas, utilizando una cámara digital QuickCam (B & N, de puerto paralelo) y un microcontrolador de la familia MCS-51, el cual, mediante el algoritmo de transformada de Hough, indicará la dirección a seguir en una trayectoria (que solo tiene ángulos rectos) por medio de una señal de control para los motores de un móvil.

Para el proyecto se desarrollaron algoritmos en lenguaje de alto nivel para la etapa de prueba en preprocesamiento y descripción de la imagen, para después traducirlos a lenguaje ensamblador y comparar los resultados obtenidos.

2. ADQUISICIÓN DE IMÁGENES

En este proyecto se toma como punto de partida el trabajo de grado "Diseño e implementación de un sistema de adquisición de imágenes vía puerto serie", en el cual un microcontrolador de la familia MCS-51 (80c32) obtiene imágenes de una cámara (QuickCam-B&N) con una resolución de 4 bits (16 niveles de gris) y las transmite utilizando una interfaz serie a un computador. Se desarrollaron dos programas:

1. Un programa en lenguaje ensamblador para manejar el protocolo de comunicación entre el microcontrolador y la cámara, y guardar una imagen en la memoria externa del microcontrolador.
2. Un programa Delphi para la recepción de datos provenientes de la memoria del microcontrolador por medio de un puerto serial y la visualización de la imagen en el computador.

Un análisis detallado de la tarjeta de adquisición y el software implementado se encuentran en [5], donde se incluyen los diagramas esquemáticos de los circuitos interfaz y el código fuente de cada uno de los programas (ensamblador y Delphi).

La puesta en funcionamiento, modificaciones y actualizaciones del hardware y software para la adquisición de imágenes se encuentran detalladas en [6].

En la Figura 1 se muestra la configuración general del hardware utilizado, en la Figura 2 se muestra la interfaz

de usuario del programa en Delphi y en la Figura 3 una imagen obtenida con el programa.

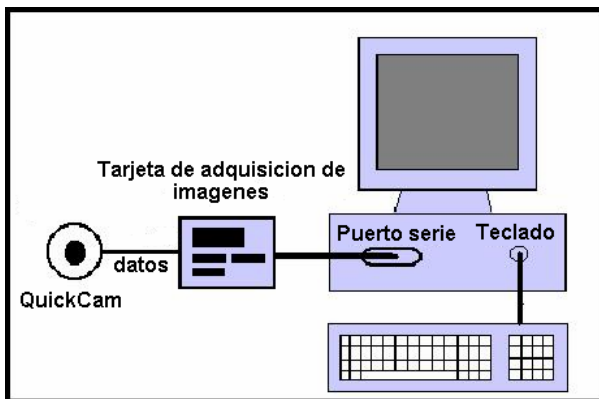


Figura 1. Configuración QuickCam-Tarjeta-PC.

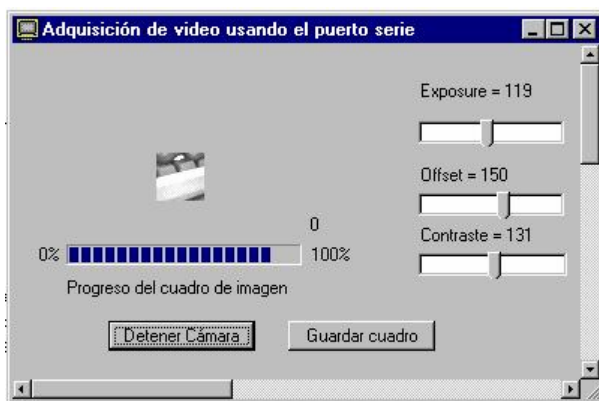


Figura 2. Interface usuario.

La utilización del programa en Delphi es clave, para visualizar, evaluar y corregir los resultados de los algoritmos implementados en el microcontrolador, pues permite comparar los resultados con obtenidos en un lenguaje de alto nivel (Builder C++).

3. PROCESAMIENTO DE LA IMAGEN

La imagen tomada en el proceso de adquisición es una línea negra sobre un fondo blanco, la cual es una parte de la trayectoria a seguir. Por lo tanto, se necesita extraer la línea (umbralización) y obtener sus características (detección de bordes), para realizar la descripción de esta de una manera más simple.

3.1 Umbralización del histograma

Supóngase una imagen como la presentada en la Figura 3, la cual contiene una cinta negra sobre un fondo blanco y su correspondiente histograma de niveles de gris (Figura 4). Una forma clara de separar la cinta del fondo es elegir un umbral T que separe dichas regiones, es decir, un punto (x, y) , tal que, $f(x, y) > T$ sea un punto

que pertenezca a la cinta. De lo contrario este punto será del fondo. Entonces una imagen umbralizada $g(x, y)$ se define en [1] la ecuación (1).

$$g(x, y) = \begin{cases} 1 & \text{si } f(x, y) > T \\ 0 & \text{si } f(x, y) \leq T \end{cases} \quad (1)$$

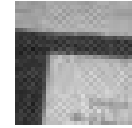


Figura 3. Imagen con 16 niveles de gris (1024 píxeles).

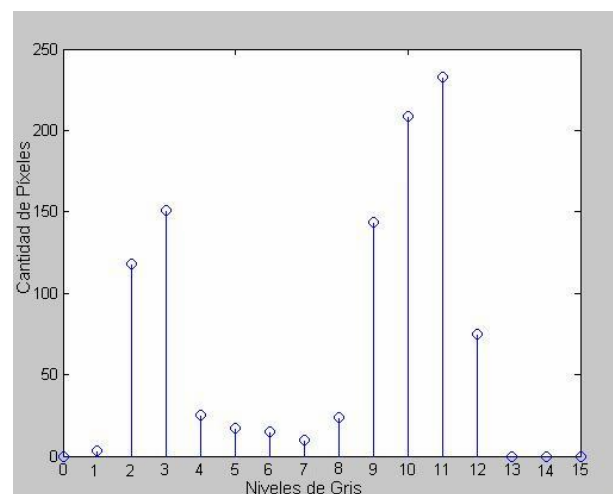


Figura 4. Histograma de la Figura 3.

En el histograma de la Figura 4, se notan dos lóbulos bien diferenciados (fondo y línea) por un valle que está entre los niveles de gris 4 y 8. Para obtener una imagen binaria en el sistema de visión se escoge un umbral, en este caso $T = 5$, se recorre la imagen que está en la memoria externa del microcontrolador y se cambia el valor de cada píxel (es decir cada posición de memoria), por uno o cero dependiendo si está por encima o por debajo del umbral.

3.2 Detección de bordes

Un borde es el límite entre dos regiones con niveles de gris distintos, de manera que la transición entre dichas regiones se pueda mirar como un cambio abrupto. La idea básica para la detección de bordes es el cálculo de una derivada, teniendo en cuenta que para una constante el valor es cero y para un cambio será diferente de cero.

3.2.1 Laplaciano

El Laplaciano es un operador escalar de segunda derivada y para funciones en dos dimensiones definido por la ecuación (2).

$$\nabla^2 f(x, y) = \frac{\partial^2}{\partial x^2} f(x, y) + \frac{\partial^2}{\partial y^2} f(x, y) \quad (2)$$

La ecuación se implementa en forma digital haciendo la convolución de una imagen con una región o máscara de 3x3 (Figura 5).

z1	z2	z3
z4	z5	z6
z7	z8	z9

Figura 5. Región de imagen de 3x3.

El principio para la definición de este operador es que el coeficiente asociado al píxel central sea positivo y los coeficientes asociados a los píxeles exteriores sean negativos, tal que la suma de todos los coeficientes sea cero. Al hacer la convolución de una de las máscaras de la Figura 6 con la imagen, el resultado será cero cuando el punto central tenga el mismo valor que sus vecinos ([1], [2]).

0	-1	0
-1	4	-1
0	-1	0

-1	-1	-1
-1	8	-1
-1	-1	-1

Figura 6. Operadores laplacianos.

3.2.2 Algoritmo de detección de bordes

Para el caso particular de una imagen binaria, se eligió el operador Laplaciano con elemento central $z_5=4$, definido en la ecuación (3). Es claro que si el valor del píxel central es diferente de alguno de sus 4 vecinos se detecta un borde.

$$\nabla^2 f = 4z_5 - (z_2 + z_4 + z_6 + z_8) \quad (3)$$

En el sistema de visión, para cada píxel de la imagen binaria (que se encuentra almacenada en la memoria externa del microcontrolador) se implementa el algoritmo de la siguiente manera:

- Se recorre la imagen (la memoria), buscando un píxel negro, es decir, si es blanco se omite el cálculo.
- Cuando lo encuentra busca en los 4 vecinos por lo menos un píxel blanco el cual activa una bandera que indica que hay un borde. Si todos los 4 vecinos son negros no es borde.

Por lo tanto se hace una reducción importante de tiempo al solo operar sobre píxeles negros y, evitar sumas y

multiplicaciones. El resultado del algoritmo se observa en la Figura 7.

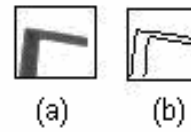


Figura 7. a) Imagen original, b) Imagen después de aplicar la detección de bordes.

4. TRANSFORMADA DE HOUGH

La transformada de Hough es una técnica utilizada para aislar características de forma particular dentro de una imagen. La idea básica es encontrar curvas que puedan ser parametrizadas como líneas rectas, polinomios y círculos.

Se puede analíticamente describir un segmento de línea en varias formas. Sin embargo una ecuación conveniente para describir un conjunto de líneas es la notación paramétrica o normal (ecuación (4)).

$$\rho = x \cos \theta + y \sin \theta \quad (4)$$

Donde ρ es la longitud de una normal desde el origen hasta la línea y θ es el ángulo de ρ con respecto al eje x .

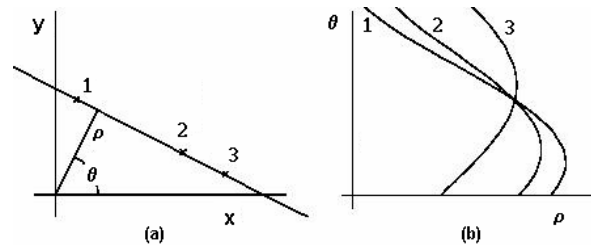


Figura 8. Representación gráfica de la transformada de Hough a) Espacio cartesiano b) Espacio paramétrico.

El espacio paramétrico se representa por una estructura rectangular de celdas, llamada arreglo acumulador y cuyos elementos son las celdas acumuladoras $A(\rho, \theta)$, las cuales son los rangos esperados de (ρ, θ) . Las celdas acumuladoras con una magnitud superior a un cierto umbral pueden ser consideradas como líneas posibles [1].

El cambio que se hizo al algoritmo de la transformada para implementarlo en el microcontrolador es básicamente crear un manejo simulado de matrices por medio de vectores, de tal forma que se calculan todos los ρ de un mismo ángulo, se encuentra el mayor y se guarda en un vector. Así hasta calcular todos los ρ de los nueve ángulos, al final se encuentra el mayor (ρ_i) entre el

vector de ρ ; la posición i es el θ esperado. Los cambios más específicos se encuentran en [6].

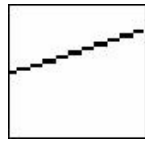


Figura 9. Imagen con línea de aprox. 20 grados.

Como ejemplo, en la Figura 9 se tiene una imagen con una línea de aproximadamente 20° a la cual se le aplica el algoritmo de la transformada de Hough. En la Figura 10 se muestran las diferentes curvas en el espacio paramétrico de la Figura 9. El punto de intersección (ρ, θ) que forman las diferentes curvas define la línea en el espacio paramétrico.

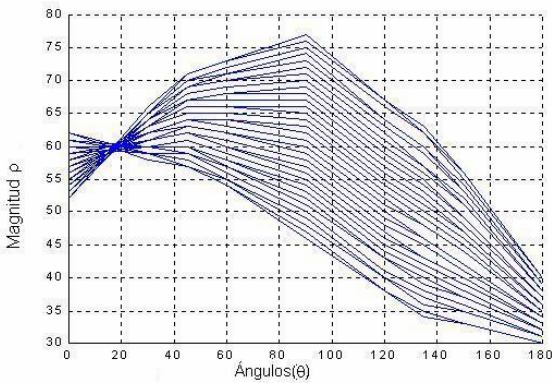


Figura 10. Transformada de Hough de una línea de 20 grados.

Para realizar el algoritmo, se hicieron pruebas en un simulador (CEIBO Debugger for Windows, Copyright (c) by CEIBO 1994, 2000), el cual permite cargar datos en la memoria externa desde un archivo .hex (para las pruebas la imagen se guardó en una archivo de este tipo), para determinar que los resultados de ρ y θ fueran correctos se compararon con los obtenidos en *hough.exe* hecho en Builder C++ ya que tienen la misma rutina. En la Figura 11 se muestra una imagen con una línea de 30° más ruido, incluyendo el resultado de los vectores acumuladores.

5. CÁLCULO DE LA TRAYECTORIA

Tomando los resultados de la transformada de Hough, el ángulo θ y los valores de ρ mayores para cada ángulo, se define una trayectoria a seguir, que para este caso serán tres direcciones (adelante, izquierda o derecha) ya que se probó el prototipo en una ruta que solo tenía ángulos rectos.

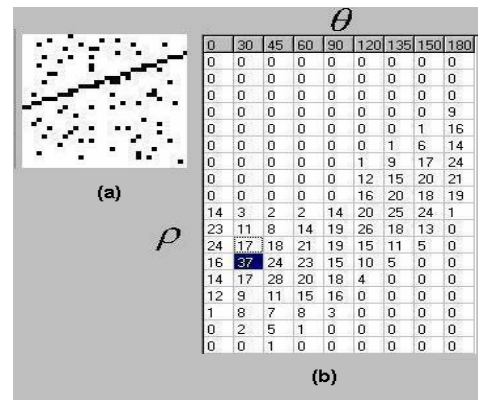


Figura 11. a) Imagen con ruido, b) Celdas de la transformada de Hough.

5.1. Dirección adelante

Cuando se tenga un máximo en la celda acumuladora que corresponde al ángulo de $\theta = 90^\circ$, se tomará la decisión de ir hacia adelante, con señales iguales para los motores.

5.2. Dirección izquierda o derecha

Al acercarse a una esquina se tienen 2 líneas cruzadas, una con ángulo de 90° y la otra con 0° o 180° . El móvil avanza hasta que la línea mayor sea la de 0° o 180° , e indicar que se realizará un giro. La transformada de Hough cuando encuentra una esquina no entrega un resultado confiable de si el ángulo está a 180° (que significaría izquierda) o a 0° (que significaría derecha) ya que realmente son los mismos. Pero sí entrega información sobre la dirección a seguir mediante las celdas acumuladoras, pues cuando el giro es hacia la izquierda las celdas desde 120° hasta 150° tienen un valor mayor a las que están desde 30° hasta 60° . Para girar de forma correcta se encuentra la mayor sumatoria entre los ángulos 120° , 135° y 150° que sería hacia la izquierda y la de 30° , 45° y 60° para la derecha (ver Figura 12 donde están resaltados los máximos para cada caso).

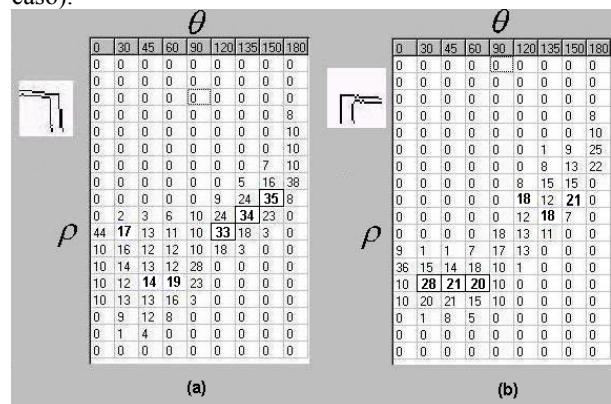


Figura 12. Esquinas y celdas acumuladoras a) Giro izquierda, b) Giro derecha.

Se puede notar que la sumatoria mayor en el caso de la Figura 12 (a), indica mayor tendencia hacia la izquierda.

Después de tomada la decisión del sentido del giro, se proporcionan las señales de PWM correspondientes a los dos motores. En la Figura 13 se muestra el móvil en el cual se probaron los algoritmos.

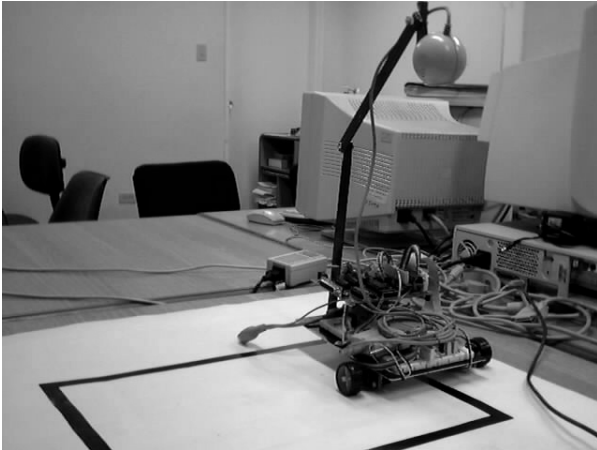


Figura 13. Móvil desarrollado.

6. RESULTADOS

Se realizaron pruebas para cada una de las direcciones tomando en cuenta: ruido por cambios en la luminosidad y huecos en las líneas. En general, un cambio de luminosidad afecta el cálculo ya que aparece demasiado ruido o la imagen puede llegar a desaparecer. En la dirección adelante (ángulo= 90°) se tiene un 100% de acierto con huecos en la línea. Para los otros casos se muestran varias imágenes donde hubo error, en la Figura 14, donde el sentido de giro calculado difiere del que realmente es a simple vista.



Figura 14. Giros erróneos.

Este problema se corrigió implementando un contador de ángulos antes que el móvil llegue a la esquina, es decir, antes de la esquina la transformada de Hough aproxima las dos rectas de la imagen a un ángulo, entonces para la izquierda se tienen las siguientes posibilidades 120° , 135° y 150° . El algoritmo incrementará el contador *izquierda* cuando la transformada de uno de estos valores y para la derecha 30° , 45° y 60° . En este caso se incrementa el contador *derecha*, finalmente se encuentra el mayor entre estos dos contadores para obtener la dirección del giro.

7. CONCLUSIONES

La dificultad para el desarrollo de software en el microcontrolador es por falta de una herramienta de depuración en línea, que permita visualizar las variables cuando se utilizan interrupciones y temporizadores por esto, la elaboración de programas de apoyo en un lenguaje de alto nivel facilita la implementación de los algoritmos en ensamblador, además permite visualizar y comparar los resultados de manera gráfica. La comunicación serial con el microcontrolador permite evaluar cada uno de los algoritmos utilizados, es decir, posibilita de alguna forma la depuración del código en ensamblador.

La imagen binaria que se obtiene es de buena calidad ya que esta tiene un alto contraste con lo cual, al umbralizar se puede quitar ruido, por esto la implementación del Laplaciano se realiza de forma sencilla ya que es específico para imágenes con poco ruido; sin embargo, en el caso contrario, el algoritmo encontraría demasiados contornos. El detector de bordes se utiliza para disminuir la cantidad de píxeles de la imagen, ya que las celdas acumuladoras de transformada de Hough en el microcontrolador son de 1 byte.

La implementación de filtros en un sistema de visión de bajo nivel, puede ser limitada por la complejidad del código debido al manejo de matrices, al hacer la convolución de una máscara con la imagen.

El algoritmo de la transformada de Hough se desempeña bien en imágenes con poco ruido. Las imágenes no deben tener demasiado ruido o más de 256 píxeles que pertenezcan a una línea ya que cada celda acumuladora es de 8 bits. Para disminuir este problema se aplica previamente la detección de bordes para mantener la forma de la línea pero con menos puntos. Aunque por limitaciones de punto flotante se utilizan solo 9 ángulos, al aumentar los intervalos posibles de ρ se obtiene una mayor exactitud al hallar el θ esperado, inclusive si se toman más puntos de una imagen.

Además de seguir una trayectoria estos algoritmos se pueden utilizar en segmentación, descriptores de contorno, de región y caracterización, tomando en cuenta las limitaciones de procesamiento y de memoria del microcontrolador.

8. BIBLIOGRAFÍA

- [1] GONZÁLES Rafael C. y WOODS Richard E.. "Tratamiento digital de Imágenes". Addison-Wesley, 1992
- [2] CASTLEMAN Kenneth. "Digital Image processing" Prentice Hall, 1996

[3] TRUCCO Emmanuele y VERRI Alessandro. "Introductory Techniques for 3-D Computer Vision". Prentice Hall, 1998

[4] GONZÁLES José Adolfo. "Introducción a los microcontroladores". McGraw-Hill, 1992

[5] RIVERA Jorge Hernando. "Diseño e implementación de un sistema de adquisición de imágenes vía puerto serie". Universidad Nacional de Colombia, Sede Manizales. Trabajo de Grado. 2001.

[6] OSPINA Emmanuel y URREA Juan Pablo. "Implementación de la transformada de Hough para la detección de líneas para un sistema de visión de bajo nivel". Universidad Nacional de Colombia, Sede Manizales. Trabajo de Grado. 2002.