

Programa Didáctico de la Transformada de Hough

Vanessa S. Schimpf y Natalia R. Scioli

Trabajo práctico final de "Captura y Procesamiento Digital de Señales e Imágenes", II-FICH-UNL.

Resumen—El objetivo de este trabajo consiste en la realización de una aplicación de tipo didáctica que permita entender la Transformada de Hough y mostrar como se la puede usar para detectar líneas rectas en una imagen.

Palabras clave—Bordes, rectas, Hough.

I. INTRODUCCIÓN

La transformada de Hough permite encontrar las formas básicas que definen una imagen a la que previamente se le ha aplicado algún detector de bordes, quedando la imagen binaria. Estas formas pueden ser representadas mediante rectas, círculos o curvas. Este programa desarrolla la versión más simple que detecta solo rectas.

Este programa permite ver esta transformación paso a paso para facilitar la comprensión de la misma. También brinda la posibilidad al usuario de ingresar puntos y/o segmentos e ir observando que ocurre en el espacio de parámetros y cual es el resultado de la transformada inversa de los mismos. La última opción es una aplicación práctica de la transformada.

II. TRANSFORMADA DE HOUGH

A. Procesado global.

Para un punto (x_i, y_i) se puede generar la ecuación de la recta que lo contiene:

$$y_i = a \cdot x_i + b \quad [1]$$

Existen infinitas rectas que cumplen con la ecuación [1] para diversos valores de a (pendiente) y b (ordenada al origen).

Examinando el espacio paramétrico de a y b (plano ab) se puede observar que un punto (a_i, b_i) representa una determinada recta en el plano xy :

$$b = -x_i \cdot a + y_i \quad [2]$$

Por lo que para cada punto (x_i, y_i) pertenecientes a una recta, se obtiene otra recta en el espacio paramétrico ab . Todas éstas se intersecan en un punto (a', b') . Dicho punto es la representación de la recta en el plano xy que contiene todos los puntos (x_i, y_i) **colineales** (que tienen pendiente a' y ordenada al origen b').

El problema que surge de esta transformación es para aquellos casos donde la recta en el plano ab es vertical, ya que la pendiente de la recta es infinita.

Para solucionar este problema Hough propuso un método alternativo que consiste en expresar la ecuación de la recta en coordenadas polares. Ésta es la llamada Transformada de Hough.

B. Método de transformación.

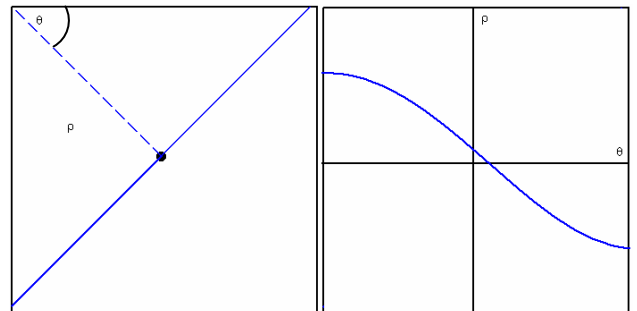
La ecuación básica para la transformada de Hough fue diseñada para detectar líneas rectas y curvas. Esto significa que cualquier línea recta en el espacio de la imagen xy es representada por un solo punto en el espacio de parámetros $\rho\theta$, y cualquier parte de esta línea recta es transformada en el mismo punto.

$$\rho = x_i \cdot \cos(\theta) + y_i \cdot \sin(\theta) \quad [3]$$

El rango del ángulo θ varía de $-\pi/2$ a $\pi/2$ medido con respecto al eje x .

El rango de ρ varía de $-N\sqrt{2}$ a $N\sqrt{2}$, siendo N la dimensión de la imagen cuadrada.

El método de Hough subdivide el espacio de parámetros en las denominadas células acumulador. La célula de coordenadas i, j con valor de acumulador A_{ij} corresponde al cuadrado asociado con las coordenadas a_i, b_j . Inicialmente estas células están puestas a cero. Después para cada punto del plano imagen se fija el parámetro θ y se calcula ρ de acuerdo a la ecuación [3].



Este plano de $\rho\theta$ es igual al ab con la diferencia de que en vez de representar la transformada de un punto como una recta, se representa como una curva sinusoidal. De esta forma, M puntos colineales en el plano xy producen M curvas sinusoidales en el plano $\rho\theta$, las cuales se intersecan en el punto (ρ_i, θ_i) con valor del acumulador igual a M .

La transformada de Hough inversa toma el valor de punto (ρ_i, θ_i) y dibuja la recta correspondiente en el plano xy .

III. PROGRAMA DIDACTICO

A. Descripción Funcional

El programa ofrece cuatro opciones para analizar la transformación de Hough desde distintos puntos de vista. Estas son:

1. Hough paso a paso: Permite al usuario observar paso a paso como se va calculando la transformada de Hough directa e inversa de un punto ingresado por el usuario en el plano xy . Mostrando un segmento de recta de color amarillo con longitud igual a la magnitud de ρ , y pendiente igual a la tangente de θ . Muestra además una recta de color verde perpendicular a la recta amarilla pasando por el punto seleccionado. En la imagen de la transformada directa se va dibujando la curva sinusoidal correspondiente al ρ_i y θ_i de la iteración, y en la imagen de la transformada inversa se van dibujando cada una de las 'infinitas' rectas que pasan por el punto seleccionado, correspondiendo en cada paso a la recta verde dibujada en la imagen a transformar. Ver ilustración en sección *Apéndices* subsección *Imágenes*.

2. Hough de puntos: Permite observar la transformación directa de un conjunto de puntos que el usuario determine. En este inciso se puede observar claramente las intersecciones de las curvas sinusoidales correspondientes a cada punto. Luego, al hacer clic en una de estas intersecciones, se puede ver la transformada inversa del punto acumulador representado por el clic.

3. Hough de Segmentos: El usuario puede ingresar segmentos de líneas y ver su respectivas transformadas directa e inversa.

4. Hough aplicada: El programa permite elegir una imagen de una lista de imágenes para calcular su transformada. Luego informa al usuario el valor del máximo acumulador para que ingrese un umbral. Todos los acumuladores con valor superior o igual al umbral serán antitransformados. Ver ilustración en sección *Apéndices* subsección *Imágenes*.

B. Implementación

Para calcular la transformada de Hough directa se aplicaron los siguientes pasos generales:

- Recorrer la imagen binaria.
- Por cada píxel mayor a cero, se recorre θ
- Por cada θ se calcula el ρ correspondiente y se suma 1 en el acumulador ρ_i , θ_i en la imagen de la transformada.

A modo de ejemplo, se adjuntó la implementación de la transformada utilizada en la opción 2 en la sección *Apéndices* subsección *Implementaciones*.

Para calcular la transformada de Hough inversa, se siguieron los pasos generales descritos a continuación:

- Recorrer la imagen de la transformada directa.
- Por cada píxel mayor a cero, se calcula el ρ_i, θ_i representativos y con ellos las coordenadas que forman la recta en el plano xy .

Para la opción 2 no se recorre la transformada completa, sino que se antitransforma el punto seleccionado en la dicha imagen.

Para la opción 3 se antitransforman solo los acumuladores cuyos valores sean mayores o iguales al 70%

del máximo valor de la transformada. Esta implementación esta adjuntada en la subsección de *Implementaciones* de la sección de *Apéndices*.

Para la opción 4 se realiza la transformada inversa según un umbral ingresado por el usuario. Además se muestra la imagen original, en la que se ha aplicado la imagen resultante de la transformación inversa para aquellos puntos que forman parte de los bordes. Ver en sección *Apéndices* subsección *Implementación*, la porción de código que realiza esta última imagen.

IV. CONCLUSIONES

El programa desarrollado muestra en forma gráfica el cálculo paso a paso de la transformada de Hough, presentado de una manera más didáctica al usuario. La implementación del mismo demuestra que los cálculos son simples, por lo cual sería una herramienta pertinente como instrumento mediador para ayudar a la comprensión del educando de la transformación en cuestión.

En un trabajo futuro, este programa se podría extender en forma simple para ilustrar la Transformada de Hough Circular, la cual excede a los objetivos iniciales del presente trabajo.

APÉNDICES

A. Algoritmos

1) Transformada Hough directa modificada para un píxel específico

```
void get_hough_directa (CImg<double> &transformada,
                      CImg<double> img,
                      int x, int y) {

    const unsigned width = transformada.dimx();
    const unsigned height = transformada.dimy();

    double scale_rho = h / (2 * sqrt( powf( float(h), 2) +
                                     powf( float(w), 2) )),
           scale_theta = width/M_PI,
           rho, theta;

    for (unsigned t=0; t < width; t++) {

        theta = t/scale_theta - M_PI/2;
        rho = x*cos(theta) + y*sin(theta);

        unsigned r = (unsigned)(rho*scale_rho + height/2.f);

        transformada(t,r)+= 1;
    }
}
```

2) Transformada Hough inversa modificada dado un umbral

```
void get_hough_inversa_segmento(CImg<double> &inversa,
                                CImg<double> img,
                                double umbral) {

    int width = img.dimx(),
        height = img.dimy(),
        y0, y1;

    double scale_rho = height / (2 * sqrt(pow( float(height), 2) +
                                           pow(float(width), 2))),
           scale_theta = width/M_PI,
           rho, theta;

    for (unsigned b=0; b < height; b++)
        for (unsigned a=0; a < width; a++)
```

```

if (img(a,b) >= umbral*0.7) {
    theta = a/scale_theta-M_PI/2;
    y0 = (int)(((b - (height/2.)) / scale_rho) / sin(theta));
    y1 = (int)(((b - (height/2.)) / scale_rho) / sin(theta) -
        width / tan(theta));
    inversa.draw_line(0, y0, width, y1, blue);
}
}
    
```

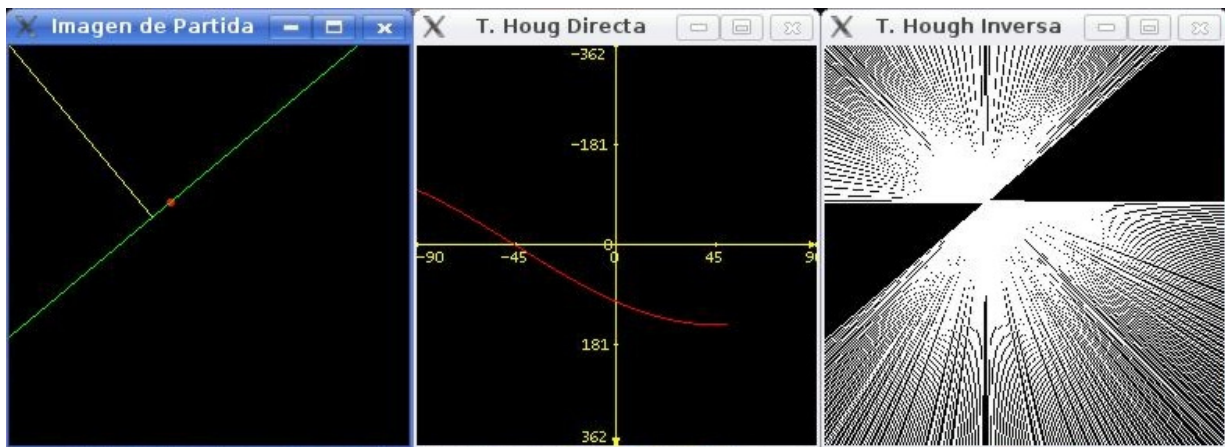
```

cimg_forXY(aplicacion,i,j){
    if( inversa(i, j, 0, 2) >= 1){
        if( img_bordes(i, j, 0, 0) >= 1 || img_bordes(i, j-1, 0, 0) >= 1 ||
            img_bordes(i, j+1, 0, 0) >= 1 || img_bordes(i-1, j, 0, 0) >= 1 ||
            img_bordes(i+1, j, 0, 0) >= 1 ){
                aplicacion(i,j,0,0)=0;
                aplicacion(i,j,0,1)=0;
                aplicacion(i,j,0,2)=255;
            }
        }
    }
}
    
```

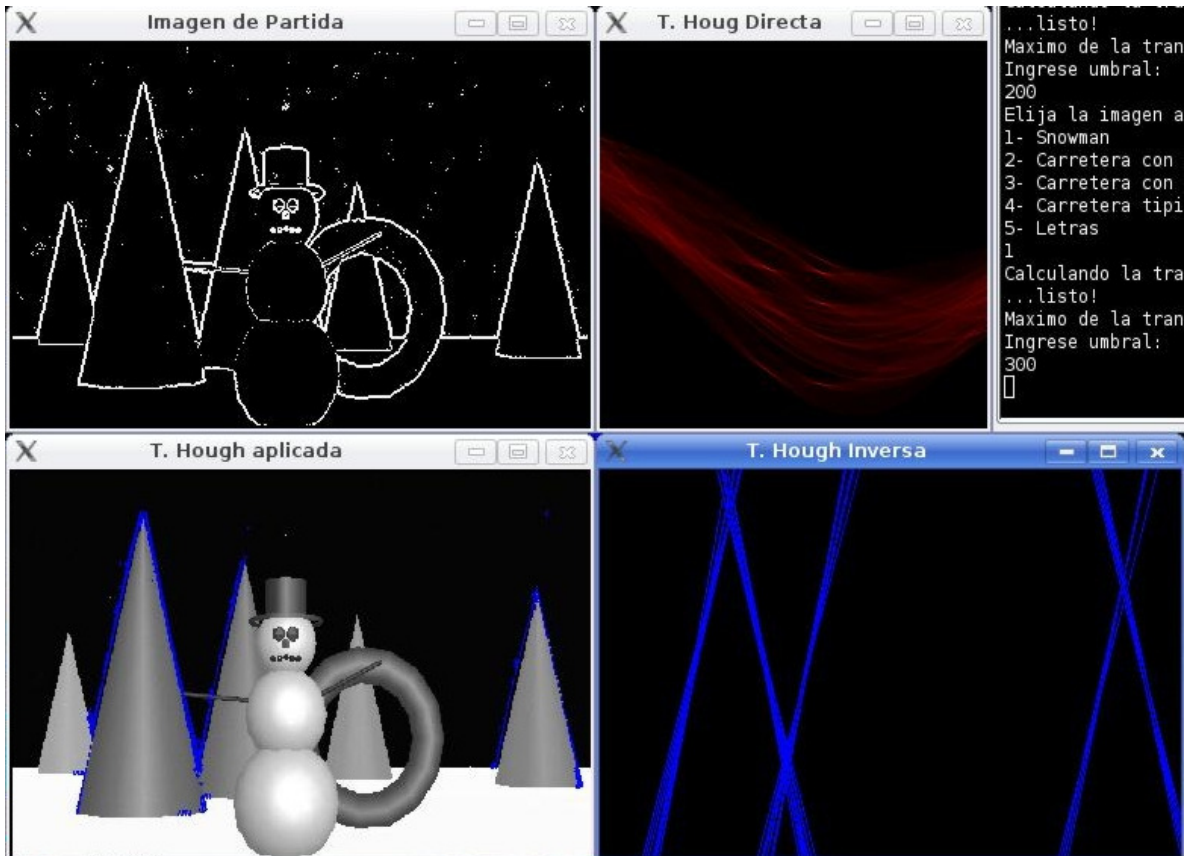
3) Porción de código donde se arma la última imagen de la opción 4:

B. Imágenes

1) Imagen 1: Opción 1- Hough por pasos



2) Imagen 2: Opción 4 – Hough Aplicada



BIBLIOGRAFÍA

- Rafael C. González. Richard E. Woods, *Tratamiento Digital de Imágenes*, USA: Editorial Addison-Wesley, 1992.
- <http://planetmath.org/encyclopedia/HoughTransform.html>
- <http://en.wikipedia.org>
- <http://homepages.inf.ed.ac.uk/rbf/HIPR2/hough.htm>