

***MEDICION DE DISTANCIAS POR MEDIO DE
VISION ROBOTICA***

JUAN SEBASTIAN PELAEZ URIBE

UNIVERSIDAD EAFIT

DEPARTAMENTO DE INFORMATICA Y SISTEMAS

MEDELLIN

2008

MEDICION DE DISTANCIAS POR MEDIO DE VISION
ROBOTICA

JUAN SEBASTIAN PELAEZ URIBE

***Trabajo de grado para optar por el título de
Ingeniero de Sistemas***

Asesor de Investigación

ADALBERTO GABRIEL DÍAZ

UNIVERSIDAD EAFIT

DEPARTAMENTO DE INFORMATICA Y SISTEMAS

MEDELLIN

2008

TABLA DE CONTENIDO

<u>TABLA DE CONTENIDO</u>	<u>IV</u>
<u>INTRODUCCIÓN.....</u>	<u>VII</u>
<u>A QUIÉN VA DIRIGIDO ESTE DOCUMENTO.....</u>	<u>IX</u>
<u>1. OBJETIVOS.....</u>	<u>10</u>
<u>1.1 OBJETIVO GENERAL</u>	<u>10</u>
<u>1.2 OBJETIVOS ESPECÍFICOS</u>	<u>10</u>
<u>2. MARCO TEORICO</u>	<u>11</u>
<u>2.1 TRASFORMADA DE FOURIER</u>	<u>11</u>
<u>2.2 TRASFORMADA DISCRETA DE FOURIER.....</u>	<u>12</u>
<u>2.2.1 ¿CÓMO FUNCIONA LA DFT EN EL ANÁLISIS DE IMÁGENES?</u>	<u>12</u>
<u>2.3 TRASFORMADA RAPIDA DE FOURIER (FFT)</u>	<u>13</u>
<u>3. ESTADO DEL ARTE</u>	<u>14</u>

<u>4. HIPOTESIS</u>	<u>16</u>
<u>5. PRUEBA DE LA HIPOTESIS.....</u>	<u>17</u>
<u>5.1 VISIÓN GENERAL DE LA SOLUCIÓN</u>	<u>17</u>
<u>5.2 PROBLEMAS ENCONTRADOS.....</u>	<u>23</u>
<u>5.3 CALIBRACIÓN DE LA FUENTE DE ADQUISICIÓN Y OBTENCIÓN DE DISTANCIAS.....</u>	<u>26</u>
<u>6. IMPORTANCIA DEL PROBLEMA EN LA CARRERA Y EL MEDIO</u>	<u>29</u>
<u>7. CONCLUSIONES.....</u>	<u>30</u>
<u>8. RECOMENDACIONES Y TRABAJO FUTURO</u>	<u>31</u>
<u>REFERENCIAS BIBLIOGRÁFICAS</u>	<u>32</u>
<u>ANEXOS.....</u>	<u>35</u>
<u>ANEXO 1: ALGORITMOS PRINCIPALES.....</u>	<u>35</u>
<u>ANEXO 2: COMO UTILIZAR LOS ALGORITMOS.....</u>	<u>38</u>

TABLA DE IMÁGENES

FIGURA 1. IMAGEN A 50CM DE LA FUENTE	18
FIGURA 2. TRANSFORMADA DE FOURIER.....	18
FIGURA 3. TRANSFORMADA RÁPIDA DE FOURIER CON FILTRO LOGARÍTMICO	19
FIGURA 4. TRANSFORMADA EN LOS CUATRO CUADRANTES.....	20
FIGURA 5. TRANSFORMADA DE FOURIER. PRIMER CUADRANTE	20
FIGURA 6. IMAGEN A 75 CM DE LA FUENTE	21
FIGURA 7. TRANSFORMADA IMAGEN 0.75M.....	22
FIGURA 8. PRIMERA ESTRUCTURA ANALIZADA	24
FIGURA 9. VALOR OBTENIDO ANALIZANDO LA IMAGEN MANUALMENTE EN MATLAB	24
FIGURA 10. SEGUNDA ESTRUCTURA ANALIZADA.....	24
FIGURA 11. COORDENADAS OBTENIDAS AL ANALIZAR LA TRANSFORMADA MANUALMENTE	25
FIGURA 12. ALGORITMOS EJECUTADOS EN MATLAB SOBRE FIGURA 12	26
FIGURA 13. IMAGEN A 25 CM DEL PUNTO DE ENFOQUE.....	27
FIGURA 14. IMAGEN A 1 METRO DEL PUNTO DE ENFOQUE.	27
FIGURA 15. GRÁFICO DE DISTANCIA VS FX	28
FIGURA 16. GRÁFICO DE DISTANCIA VS FY	28

INTRODUCCIÓN

Siempre que se abre los ojos, se observa el lugar que nos rodea y se responde a la pregunta ¿dónde me encuentro?, ¿qué es esto?, ¿que estoy haciendo?, ¿qué puedo hacer? Igual situación ocurre cuando se está explorando un sitio nuevo, se admira y se busca responder una serie de interrogantes.

Se podría ver que una gran parte de las fuentes algorítmicas se apoyan en los sistemas biológicos; pues aquí también ocurre algo similar al sustituir las actividades humanas por una máquina, un robot, un vehículo o una aeronave autónoma. Pensaríamos que estas también buscarían responder a unos interrogantes y que éstas respuestas les ayudarían a definir su comportamiento. Pues es verdad, el comportamiento de estas máquinas se ha definido no en forma prescriptiva a unos cuantos casos de situaciones, sino un sinnúmero de posibilidades estocásticas, las cuales no podríamos cubrir con un vasto algoritmo prescriptivo, porque ellas vienen determinadas por la probabilidad de ser o de ocurrir. En tal caso el sistema responderá en forma inteligente a una situación que no se había concebido pero que es capaz de analizarlas y tomar decisiones.

Percibir el espacio que nos rodea es un problema complejo que incluye sistemas de metrología, apreciación, identificación, caracterización, medición, valoración para que se pueda disponer de él.

El primer problema a resolver es la observación del espacio libre y la obtención de distancias de la fuente de adquisición hacia los objetos u obstáculos del escenario. Es importante destacar que para estas mediciones es necesario un sistema metrologico que sea muy confiable, veloz y acorde con la realidad, que además, tenga una infraestructura simple, comparada con los sistemas actuales de medición.

Lo anterior permite avanzar en el campo de la visión artificial y se plantea como base para procesos de levantamiento de la geometría del escenario, sobre el cual una cabeza robótica se podría desplazar.

Durante el desarrollo de la propuesta se realizó una ampliación del estudio y se analizo el estado actual del procesamiento de señales digitales, más específicamente, el procesamiento de imágenes, que buscaba garantizar la calidad y el nivel de trabajo y una contribución al conocimiento universal en este tema.

Luego de construir los algoritmos para el proceso metrologico, se realizaron pruebas que confirmaron la precisión y velocidad de los algoritmos de metrología, objetivo principal de este proyecto.

A QUIÉN VA DIRIGIDO ESTE DOCUMENTO

El trabajo de grado desarrollado tiene como objetivo estudiantes e investigadores familiarizados con procesamiento de imágenes y el entorno de desarrollo MATLAB®, del cual deben tener conocimientos intermedios sobre sus Toolboxes de procesamiento de imágenes y adquisición de imágenes.

Este trabajo fue desarrollado en ambientes normales, en los cuales se conto con una cámara, un computador personal convencional y alguna herramienta para exponer las imágenes (pantalla de otro computador o televisor, para mostrar imágenes y cambiarlas rápidamente, o paredes con estructuras bien definida, como muro de ladrillos), por lo que se puede probar en cualquier ambiente, sin ningún contratiempo.

1. OBJETIVOS

1.1 OBJETIVO GENERAL

Obtener la distancia entre una única fuente de adquisición de imágenes y una superficie que presente patrones simétricos en su estructura

1.2 OBJETIVOS ESPECÍFICOS

Desarrollar algorítmicas de metrología (medición de distancias) por medio de visión y procesamiento digital de imágenes.

Simplificar la logística necesaria para procesos de metrología y con esto reducir los costos que implica tipo de montajes.

Reducir la complejidad algorítmica y el costo computacional que conlleva el análisis metrologico de las imágenes.

2. MARCO TEORICO

2.1 TRASFORMADA DE FOURIER

La trasformada de Fourier es una importante herramienta matemática ampliamente utilizada en el análisis de señales digitales.

La trasformada permite convertir el dominio del tiempo o del espacio al dominio de la frecuencia. (FISHER, PERKINS, WALKER y WOLFART, 2003), permitiendo esto analizar diversas características de la señal por medio de las propiedades frecuenciales obtenidas (CALIFORNIA TECHNOCAL PUBLISHING, 2007).

Este tipo de trasformadas son utilizadas en un amplio rango de aplicaciones, que incluyen análisis, filtrado, reconstrucción y compresión de imágenes.

Pero calcular la transformada de Fourier es un procedimiento que conlleva altos costos computacionales, en especial si se analizan señales de más de una dimensión, como las imágenes en escala de grises(2 dimensiones), por lo que para el análisis de imagen se utiliza otro tipo de

transformada de Fourier, la transformada discreta de Fourier (DFT) (FISHER, PERKINS, WALKER y WOLFART, 2003).

2.2 *TRASFORMADA DISCRETA DE FOURIER.*

La Transformada discreta de Fourier es una muestra, o subconjunto de datos, de la Transformada de Fourier (FISHER, PERKINS, WALKER y WOLFART, 2003) y aunque no contiene la información de todas las frecuencias que describen la imagen, contiene un conjunto muy importante de estas frecuencias, que es suficiente para describir todas las características del dominio especificado.

Es por esto que este tipo de transformadas ha sido ampliamente utilizado en el análisis de imágenes, pues reduce los costos computacionales y describe completamente el comportamiento de la señal en el dominio espacial.

2.2.1 ¿Cómo Funciona la DFT en el Análisis de Imágenes?

Para una imagen cuadrada de $N \times N$ pixeles la transformada DFT viene dada por:

$$F(k, l) = \frac{1}{N^2} \sum_{a=0}^{N-1} \sum_{b=0}^{N-1} f(a, b) e^{-i2\pi(\frac{ka}{N} + \frac{lb}{N})}$$

Siendo $f(a,b)$ la función que representa cada uno de los puntos de la imagen en el dominio espacial, y el termino exponencial la función base para representar los puntos $F(k,l)$ en el dominio de Fourier.

Esta ecuación puede ser interpretada así: El valor de cada punto $F(k,l)$ es obtenido multiplicando la imagen en el dominio espacial, con la correspondiente función base y sumando los resultados.

2.3 *TRASFORMADA RAPIDA DE FOURIER (FFT)*

Es un eficiente algoritmo que permite calcular de manera rápida la transformada discreta de Fourier de un determinado tipo de señal, permitiendo esto reducir costos computacionales que conlleva el cálculo de estas. De las diversas formas que existen para calcular la transformada de Fourier, la FFT es el único método práctico (CALIFORNIA TECHNICAL PUBLISHING, 2007)

3. ESTADO DEL ARTE

Actualmente, el procesamiento de imágenes ha sido utilizado para reconocer características de lo que alguna imagen nos está mostrando. Por medio de las técnicas actuales es posible reconstruir las imágenes partiendo de una imagen borrosa, detectar sus puntos de enfoque, detectar los patrones (objetos) que hacen parte de ella, pero hasta el momento, no se ha encontrado alguna técnica que permita reconocer la distancia de la imagen utilizando sólo una fuente de captura de imágenes. Por ejemplo, actualmente es necesario utilizar varias cámaras para realizar triangulaciones y así saber a qué distancia se encuentra determinada superficie, también, se utilizan además de cámaras, sensores de sonido o de luz que determinan la distancia partiendo del tiempo que se demore en retornar una señal que fue enviada a la superficie (ODA, 1996).

También existe un proyecto en el que se utiliza una cámara en movimiento para medir la distancia a un punto fijo, diferenciándose de este, en que la cámara debe capturar una imagen de este punto tres veces en diferentes instantes de tiempos y hacer cálculos con la velocidad, sin analizar la estructura y propiedades de la imagen, y sin poder analizar diferentes

puntos con la misma estructura (LAMPRECHT, RASS, FUCHS, Y KYAMAKYA, 2008).

Lo que pretende este proyecto de grado es aplicar técnicas existentes en el procesamiento de imágenes para construir un algoritmo metrologico utilizando una sola fuente de adquisición, que analice las características de las imágenes y pueda ser reutilizado en estructuras que presenten patrones similares.

4. HIPOTESIS

Analizando una imagen con una estructura bien definida, es posible, mediante el análisis de su transformada rápida de Fourier (FFT, por sus siglas en inglés), identificar la distancia entre ésta y la fuente de captura (lente de cámara). Esto debido a que mediante la FFT, se pueden identificar las frecuencias en las que se encuentran sus puntos más brillantes, y a medida que la fuente de captura se acerca o aleja de la estructura, las frecuencias de sus puntos brillantes aumentan o disminuyen.

Luego de identificar las distancias de la cámara a esas estructuras, dependiendo de las frecuencias obtenidas, se puede realizar una construcción de esta en algún lenguaje CAD o una herramienta, como las superficies 3D de MATLAB®, para su posterior visualización.

5. PRUEBA DE LA HIPOTESIS

5.1 Visión General de la Solución

Utilizando el entorno de desarrollo MATLAB®, o cualquier entorno de desarrollo que permita manipular imágenes y realizar operaciones matemáticas sobre ellas, se pueden realizar pruebas sobre imágenes adquiridas con cualquier fuente de captura de imágenes.

Para probar lo propuesto, se debe obtener primero el comportamiento de los puntos brillantes de la estructura que se desea obtener, y para esto se utiliza la transformada rápida de Fourier.

Al comenzar la prueba, se utilizó una imagen a 0.5m de distancia (Figura 1) de la fuente de captura, sobre la que se realizaron varios procedimientos utilizando transformadas de Fourier hasta obtener el comportamiento deseado.

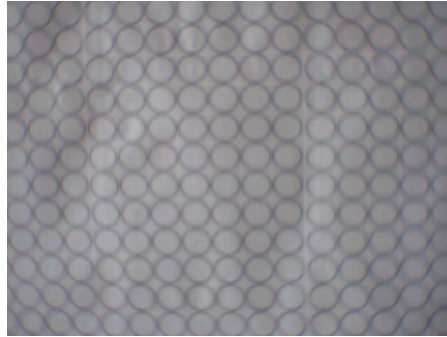


Figura 1. Imagen a 50cm de la fuente

Luego de realizar una conversión de la imagen a escala de grises por medio del comando `rgb2gray` (al ingresarla a MATLAB es RGB), y aplicar una transformada rápida de Fourier sobre la imagen adquirida, en MATLAB, se obtiene el siguiente resultado:

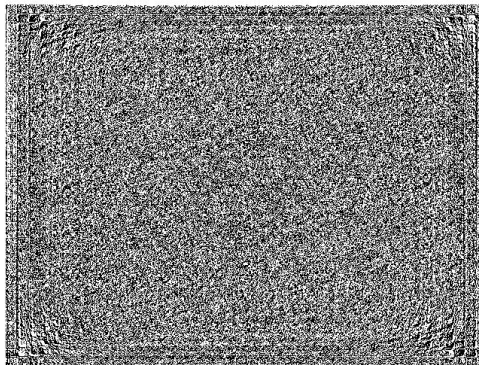


Figura 2. Transformada de Fourier

Con esta transformada no es posible determinar el comportamiento de los puntos brillantes, pues presentan una gran cantidad de puntos con valor 0, que son representados en la imagen como puntos blancos, por lo que fue necesario buscar la manera de mejorar la forma en cómo se muestra la

transformada, para esto, se utilizo este filtro logarítmico $(\log(\text{abs}(\text{"transformada"})), [])$ (UNIVERSITY OF SUSSEX, 2006), que permite ver mejor los detalles al controlar los puntos de frecuencia cero, que son dominantes en la transformada.

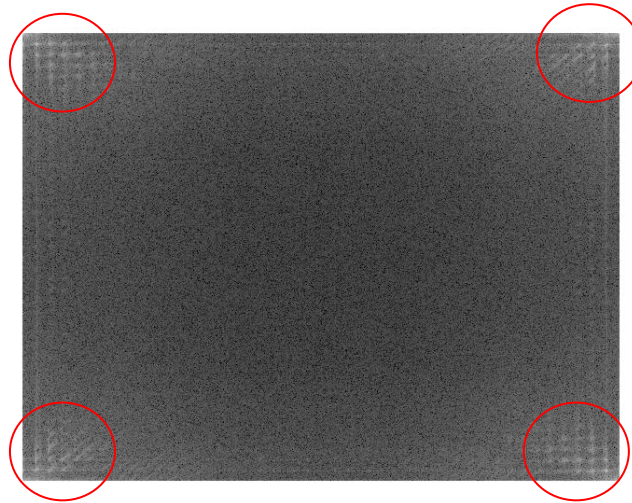


Figura 3. Transformada Rápida de Fourier con Filtro Logarítmico

Al poder apreciar de mejor manera los detalles, esta nueva imagen permite ver los puntos más brillantes, encerrados con círculos en la figura 3, con lo que se empieza a alcanzar el objetivo de este documento.

Pero, los puntos pueden apreciarse de mejor manera si la transformada de Fourier se lleva a los cuatro cuadrantes del plano cartesiano y luego, a partir de estos, seleccionar el primer cuadrante, pues los cuatro, presentan el mismo comportamiento, como se puede ver en la figura 4, lo que hace necesario analizar solo uno de ellos. Luego se busca obtener las frecuencias de los puntos más brillantes (máximos locales) dentro de la transformada.

Para llevar la transformada al plano cartesiano se aplican la funciones `fftshift(log(1+abs( )))` y `gscale` sobre la transformada (GONZÁLES, 2004).

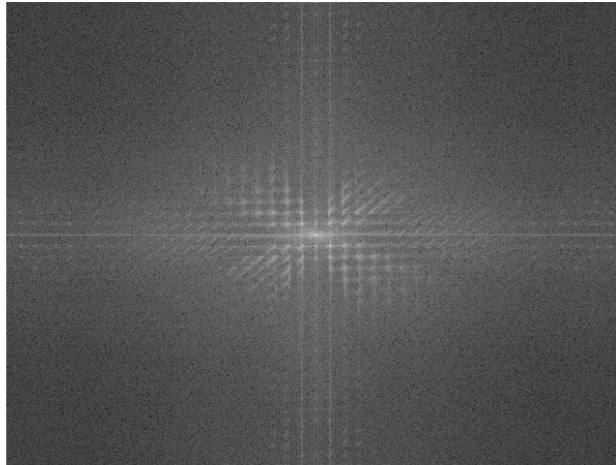


Figura 4. Transformada en los cuatro cuadrantes



Figura 5. Transformada de Fourier. Primer Cuadrante

Ya con este resultado se es posible obtener la frecuencia del primer punto más brillante en el plano (descartando el origen, pues por las pruebas

realizadas, en todas las transformadas es el de mayor brillo) mediante la obtención de los puntos más brillantes en los ejes x e y, que son las coordenadas del primer punto (la intersección de los puntos brillantes en los ejes x e y son los puntos brillantes del plano). En el ejemplo, figura 5, las coordenadas son (15,12), que son respectivamente las coordenadas en x e y, de los primeros puntos más brillantes en esos ejes.

Luego de obtener estas frecuencias se procedió a analizar una imagen con mayor distancia que la anterior y comparar sus resultados.

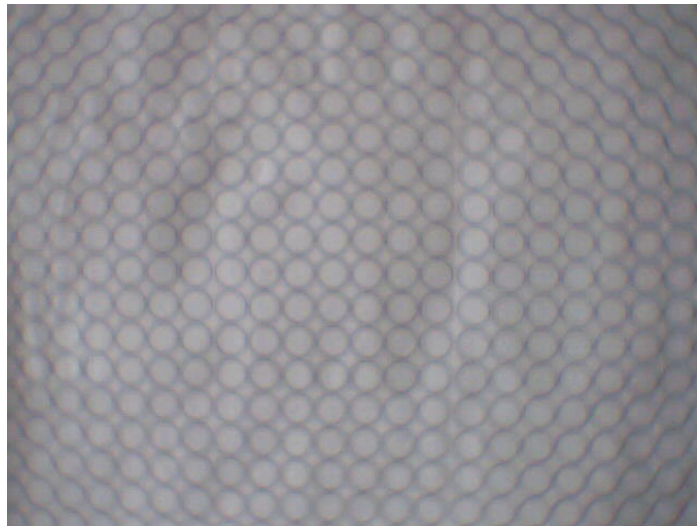


Figura 6. Imagen a 75 cm de la Fuente

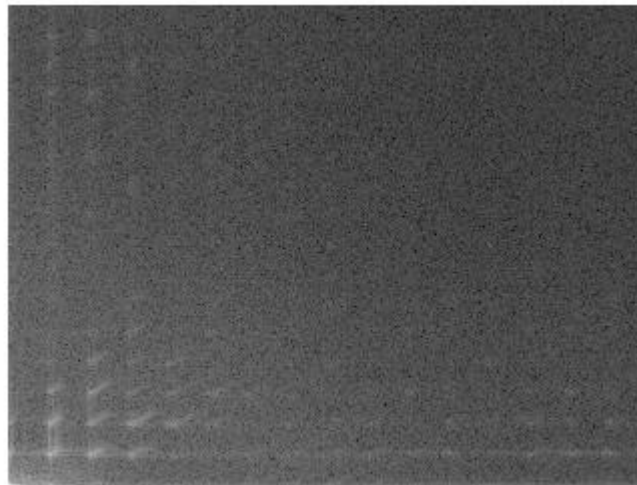


Figura 7. Transformada Imagen 0.75m

Al comparar las figuras 5 y 7, se observa que el comportamiento de las frecuencias de los primeros puntos brillantes son diferentes, es decir, la frecuencia es mayor si la imagen se encuentra más alejada, y menor, a medida que se acerca la fuente a la imagen. Aplicando el procedimiento en MATLAB® se observa lo mismo, las coordenadas del punto de más brillo, en la imagen cercana, son (15,12), menores a las que se obtuvieron con esta última: (21,15).

Cabe aclarar que para llevar a cabo lo anterior es necesario realizar una calibración a la fuente de adquisición en la que se indiquen las distancias desde el punto de adquisición y la imagen en dos puntos diferentes, para así, por medio de interpolación, o extrapolación (Capítulo 5.3) calcular las distancias de las frecuencias que se obtengan posteriormente. Cuando se

plantee realizar esto para un patrón de estructura diferente al calibrado, es necesario realizar una nueva calibración.

Cuando se tienen las coordenadas identificadas, por medio de herramientas de visualización 3D, como superficies en MATLAB, se puede construir el escenario básico, siendo esto tema de inicio de nuevos proyectos en el campo de visión.

5.2 *Problemas encontrados*

Dependiendo del tipo de estructura que tenga la imagen que se va a analizar, es posible que los puntos brillantes en los ejes x e y, no coincidan con las coordenadas x e y del punto más brillante en la transformada.

La técnica utilizada en un principio en los algoritmos que permiten la detección de estas coordenadas, fue precisamente encontrar los puntos más brillantes en los ejes x e y, algo que funcionó en un principio al analizar imágenes con estructuras como la de la figura 10. Al analizar una estructura diferente, figura 12, con la función desarrollada en un principio (Anexo 1: analizarAnterior) se evidencio de manera clara este problema y se hizo necesario replantear la técnica a utilizar en los algoritmos.

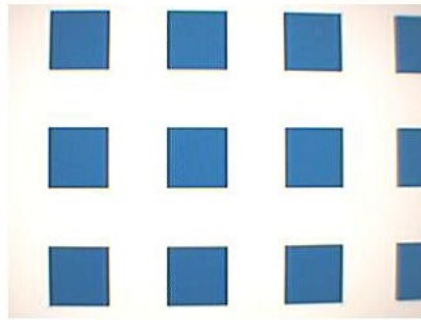


Figura 8. Primera Estructura Analizada

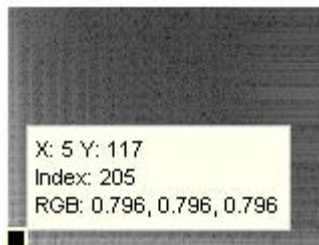


Figura 9. Valor Obtenido analizando la imagen manualmente en MATLAB

El resultado de la función `analizarAnterior` aplicada a la Figura 8 es: (frecuencia en x, frecuencia en y) = (5,4), valor que comparado con el obtenido mediante análisis manual en MATLAB es igual. (Figura 9).



Figura 10. Segunda Estructura Analizada

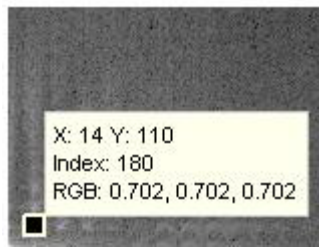


Figura 11. Coordenadas obtenidas al analizar la transformada manualmente

Aplicando esta misma función sobre la Figura 10, el resultado obtenido son las coordenadas (2,2), valores que se diferencian en gran medida a los obtenidos mediante el análisis manual en MATLAB (14,11) (Figura 11). La figura 13 arroja el resultado (14, 110), pues MATLAB trata la imagen como una matriz y el origen del eje y está en la parte superior, por lo que se realiza una conversión para ubicarlo en la parte inferior, dando esta conversión el resultado ($x = 14, y = 11$).

Para obtener solución a esto se hizo necesario construir un nuevo algoritmo (Anexo 1: analizar), en el que se excluyen los ejes x e y, que en algunas estructuras presentan puntos muy brillantes cerca al origen, sin que las coordenadas reflejen un punto brillante en el espectro de Fourier, por lo que este algoritmo busca los puntos brillantes dentro de este espectro. Aplicando este algoritmo a las Figuras 10 y 12, se obtuvieron las mismas coordenadas del análisis manual. Figura 10: (5,4) , Figura 12: (14,11).

```
>> [fx,fy] = analizarAnterior(cincuenta)

fx =

     2

fy =

     2

>> [fx,fy] = analizar(cincuenta)

fx =

    14

fy =

    11
```

Figura 12. Algoritmos ejecutados en MATLAB sobre Figura 12

5.3 *Calibración de la fuente de Adquisición y*

Obtención de Distancias.

Para obtener la distancia entre la fuente de captura de las imágenes, y estas, es necesario realizar una calibración por medio de la cual sea posible conocer el comportamiento de las frecuencias (en x e y independientemente) respecto a la distancia de ellas.

Para esto, se construye un método, que por medio de interpolación (función polyfit en MATLAB®) haya los coeficientes de los polinomios para las frecuencias x e y, y luego de ejecutarlo con imágenes iniciales a 50 y 75

centímetros, luego estos coeficientes son utilizados para hallar la distancia de las siguientes imágenes.



Figura 13. Imagen a 25 cm del punto de enfoque



Figura 14. Imagen a 1 metro del punto de enfoque.

Al ejecutar el método para obtener las distancias sobre la Figura 13, el resultado obtenido fue: 23 cm, y sobre la Figura 14: 99.6 cm. Esto da al algoritmo una precisión entre 90% y 99%.

La forma de calcular estas distancias consiste en promediar la distancia obtenida por medio del polinomio de frecuencias en x, con la obtenida con el polinomio de frecuencias en Y, las figuras 15 y 16 muestran las graficas de los polinomios x e y obtenidos a partir de las figuras 6 y 10.

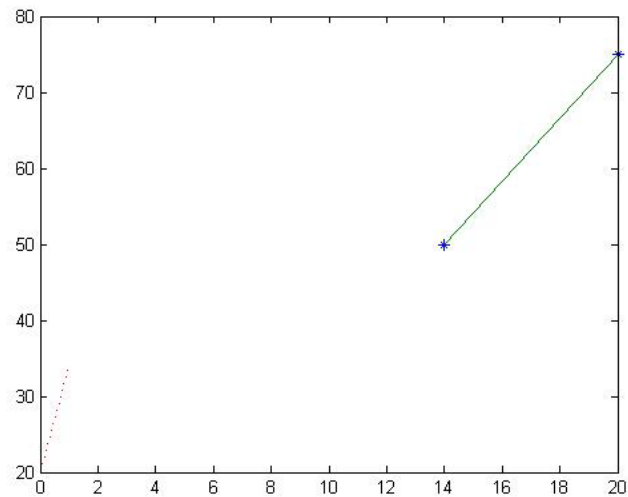


Figura 15. Gráfico de Distancia vs F_x

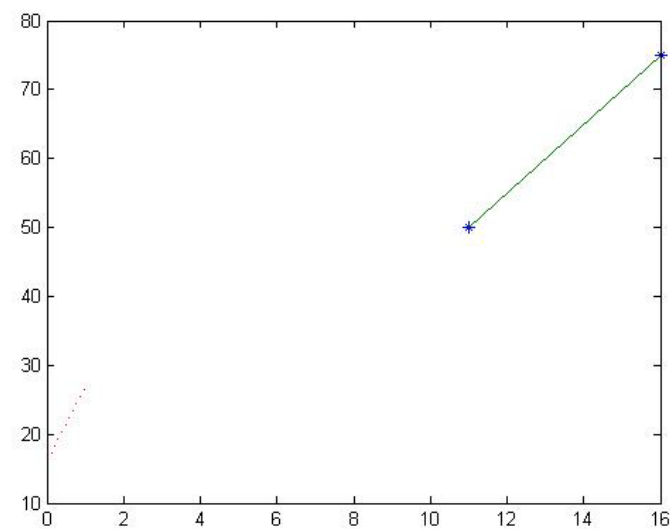


Figura 16. Gráfico de Distancia vs F_y

6. IMPORTANCIA DEL PROBLEMA EN LA CARRERA Y EL MEDIO

Con este proyecto se da inicio y aporta en gran medida a futuros proyectos de inspección visual en robótica e inspección de la calidad en procesos de producción, pues por medio del análisis de las características de las imágenes, se puede analizar la característica de los productos en un proceso de producción.

El proyecto permite el desarrollo de nuevas tecnologías para potencializar y aumentar el desarrollo y crecimiento de la industria del país. Estas tecnologías permitirán automatizar los procesos de control de calidad (análisis de características de los objetos), que permitan la creación de productos con excelente calidad.

Con el proyecto, también se pretende realizar aportes que permitan avanzar en diversas áreas del conocimiento y la industria, tal como la robótica, manufactura, la biotecnología y medicina, la construcción, etc.

7. CONCLUSIONES

Para identificar las distancias entre la imagen (estructura) y su fuente de captura, por los métodos utilizados en este proyecto, se debe tener una estructura definida para que sea posible por medio del análisis de Fourier identificar las frecuencias de los puntos brillantes.

Para realizar la medición de distancia hacia una superficie es necesario realizar calibración a la fuente de adquisición y trabajar siempre sobre una superficie con las mismas características.

Cuando se trabaja en procesamiento de imágenes digitales, no es una buena práctica realizar generalizaciones, es recomendable realizar pruebas con diferentes tipos de imagen

8. RECOMENDACIONES Y TRABAJO FUTURO

Este proyecto es base para el desarrollo de nuevos proyectos de visión artificial, que incluyen la construcción de escenarios complejos, detección de obstáculos y navegación autónoma de máquina, pues el primer problema a atacar es la medición de las distancias en el escenario.

Lo anterior se puede lograr con herramientas como redes neuronales, pues estas permiten realizar mediciones y la construcción de los escenarios sin necesidad de realizar calibraciones, ya que por medio de ellas es posible determinar el tipo de estructura y la calibración de forma automática.

Por esto se recomienda continuar el desarrollo en este campo aplicando redes neuronales, pues, esto permitirá más adelante construir algoritmos inteligentes que puedan determinar distancias y construir escenarios más precisos en tiempo real. Con el escenario construido, es posible desarrollar algorítmicas de navegación autónoma para máquinas o robots.

REFERENCIAS BIBLIOGRÁFICAS

CALIFORNIA TECHNICAL PUBLISHING. Digital Signal Processing, Capítulo 24: Linear Image Processing: Fourier Image Analysis. 2007.
<http://www.dspguide.com/ch24/5.htm>

FISHER, Robert y PERKINS, Simon y WALKER, Ashley y WOLFART, Erik. Fourier Transform. 2003.
<http://homepages.inf.ed.ac.uk/rbf/HIPR2/fourier.htm>

FORSYTH, David .A. y PONCE, Jean. Computer Vision: A Modern Approach. New Jersey: Pearson Education, 2003.

GONZÁLEZ, Rafael C. y WOODS, Richard E. y EDDINS, Steven L. Digital Image Processing using MATLAB. New Jersey: Pearson - Prentice Hall, 2004. 609 p.

KOVESÍ, Peter. MATLAB and OCTAVE functions for Image Processing. University of Western Australia: School of Computer Science and Software Engineering. <http://www.csse.uwa.edu.au/~pk/research/matlabfns/>

LAMPRECHT, Bernhard, RASS, Stefan, FUCHS, Simone y KYAMAKYA, Kyandoghere. Fusion of an uncalibrated camera with velocity

information for distance measurement from a moving camera on highways.

Austria: Klagenfurt University, 2008.

http://ieeexplore.ieee.org/xpl/freeabs_all.jsp?tp=&arnumber=4510371&isnumber=4510344

LOPEZ MENDEZ, Ender José. Tutorial de MATLAB. Universidad del Zulia. Maracaibo, Abril de 2003.

<http://www.monografias.com/trabajos13/tumatlab/tumatlab.shtml>

MANDIC, Danilo P. y CHAMBERS, Jonathon A. Recurrent Neural Networks for Prediction. New Jersey: John Wiley & Sons, Electronic Ed, 2001. 297 p

MANN, Steve. Intelligent Image Processing. New Jersey: John Wiley & Sons, Electronic Ed, 2002. 359 p.

MathWorks Inc, The. Image Processing Toolbox™ User Guide. Natick: MathWorks, 2008. p 51-100, 303-316

NAVARRO, Daniel y CUEVAS, Erik. Visión por Computador utilizando MATLAB y el Toolbox de Procesamiento Digital de Imágenes. 35p.
<http://www.scribd.com/doc/23371/Procesamiento-de-imagenes-con-Matlab>

ODA, Hajime, SAITO, Hiroyuki KAWAI, Michio. Distance measuring device for a camera, 1995. <http://www.freepatentsonline.com/5389996.html>

RESTREPO RIVAS, Luis Guillermo, Una introducción a la visión de máquina. 2007. <http://luisguillermo.com/visiona.htm>

UNIVERSITY OF SUSSEX. Fourier Transforms Demonstrations. Brighton, 2006. http://www.cogs.susx.ac.uk/courses/acv/matlab_demos/fft_image_demo.html

ANEXOS

ANEXO 1: ALGORITMOS PRINCIPALES

1. analizarAnterior

```
function [fx,fy] = analizarAnterior(imag)

%Funcion que permite analizar una imagen
%
%Recibe como parametro de entrada una imagen y retorna las
frecuencias de
%sus puntos mas brillantes.
%

warning off all;

%Si la imagen no esta en escala de grises, se realiza el siguiente
%procedimiento
if isgray(imag) == 0
    imag=rgb2gray(imag);
end

%Calcula la transformada de Fourier para la imagen de entrada
fourier = fft2(imag);

%Aplica el filtro logaritmico
trans = fftshift(log(1+abs(fourier)));

%Transforma el espectro de Fourier a los cuatro cuadrantes
trans = gscale(trans);

[fil,col]=size(trans);

%De la transformada solo es necesario el primer cuadrante del plano
%XY, por eso se realiza esta conversion
```

```
%Selecciona el primer cuadrante del espectro resultante
trans = trans(2:(fil/2+1),(col/2+1):col);

%Busca el punto de mayor intensidad y devuelve sus coordenadas
[fx,fy] = mayores(trans);
```

End

2. analizar

```
function [fx,fy] = analizar(imag)

%Funcion que permite analizar una imagen
%
%Recibe como parametro de entrada una imagen y retorna las
frecuencias de
%sus puntos mas brillantes.
%

warning off all;

%Si la imagen no esta en escala de grises, se realiza el siguiente
%procedimiento
if isgray(imag) == 0
    imag=rgb2gray(imag);
end

fourier = fft2(imag);

trans = fftshift(log(1+abs(fourier)));

trans = gscale(trans);

[fil,col]=size(trans);

%De la transformada solo es necesario el primer cuadrante del plano
%XY, por eso se realiza esta conversion

trans = trans(2:(fil/2+1),(col/2+1):col);

%Calcula los puntos mas brillantes y sus coornedanas
[fx,fy] = mayorIntensidad(trans);
```

End

3. Calibrar

```
function [polinomiox,polinomioy] =  
calibrar(imagen1,distancia1,imagen2,distancia2)  
%funcion que obtiene los polinomios para encontrar la distancia de  
%la imagen. Los parametros a ingresar son:  
%  imagen1: Imagen con menor distancia  
%  distancia1: distancia de la imagen1  
%  imagen2: imagen con mayor distancia  
%  distancia2: distancia de la imagen2  
%  
%Retorna los polinomios que obtienen las distancias dependiendo de  
%las  
%frecuencias ingresadas para cada eje.  
%  
    [fx,fy] = analizar(imagen1);  
    frecuenciax(1) = fx;  
    frecuenciay(1) = fy;  
  
    [fx,fy] = analizar(imagen2);  
    frecuenciax(2) = fx;  
    frecuenciay(2) = fy;  
  
    distancia(1) = distancia1;  
    distancia(2) = distancia2;  
  
    polinomiox=polyfit(frecuenciax,distancia,1);  
    polinomioy=polyfit(frecuenciay,distancia,1);  
end
```

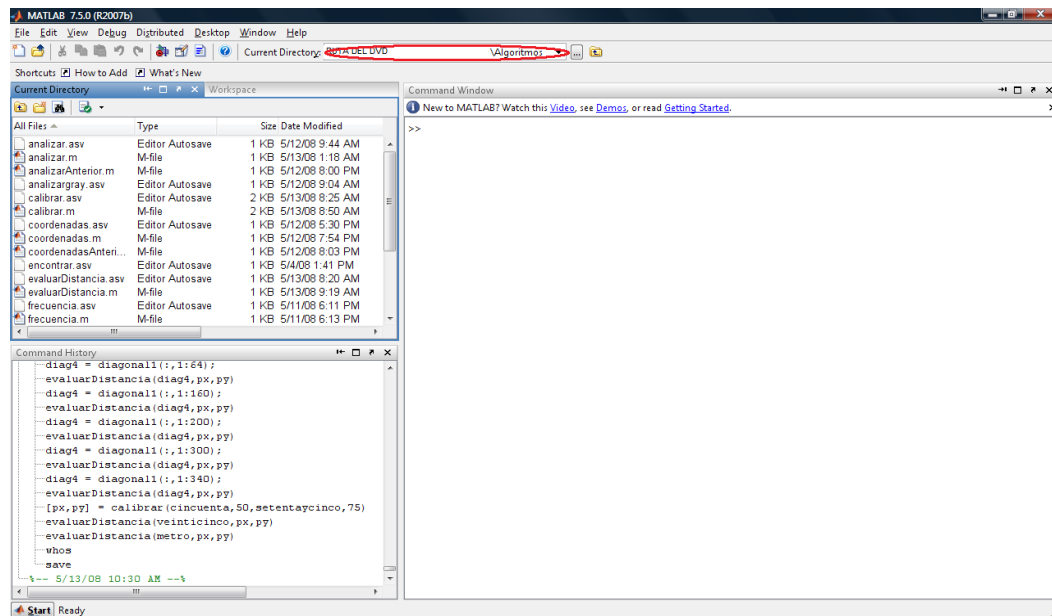
4. evaluarDistancia.

```
function distancia = evaluarDistancia(imagen, polinomiox,polinomioy)  
%Metodo que obtiene la distancia de una imagen luego  
%de haber calibrado la fuente de adquisicion.  
% Recibe como parametros la imagen y los polinomios  
% obtenidos con el proceso de calibracion.  
  
%Calcula las frecuencias de la imagen para hallar su distancias
```

```
[fx,fy] = analizar(imagen);  
distx = polyval(polinomiox,fx) ;  
disty =polyval(polinomioy,fy);  
  
%Luego de hayar las distancias respecto a x e y, las promedia  
%para el mejor resultado posible.  
distancia = (distx+disty)/2;  
end
```

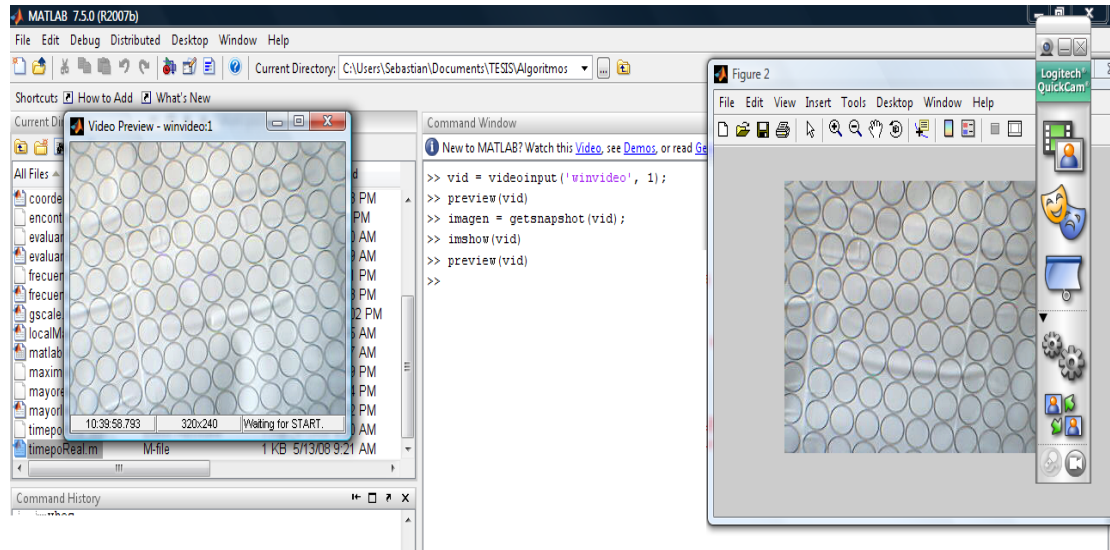
ANEXO 2: COMO UTILIZAR LOS ALGORITMOS

Para comenzar a utilizar los algoritmos es necesario, primero que todo, ubicar la ruta de MATLAB en la carpeta Algoritmos del DVD, como se muestra en la siguiente imagen.



Luego de realizar lo anterior, se realiza la carga de alguna(s) imágenes a la memoria, bien puede ser por medio de la instrucción `imagen = imread('rutadeimagen')` (Pueden ser utilizadas las imágenes de la carpeta

Figuras del DVD) o creando una fuente de video y obteniendo una imagen como se muestra a continuación.



Las instrucciones anteriores crean una fuente de captura de video y la instrucción `getsnapshot(vid)` permite capturar una imagen de lo que se la cámara está captando en ese momento.

Al finalizar la captura de la imagen o imágenes, se elimina la fuente de video y se limpia la memoria de MATLAB mediante los comandos `delete(vid)` y `clear vid`.

Con las imágenes en memoria, y teniendo la medición de las distancias de dos de estas, se procede a utilizar la función `calibrar` de la siguiente manera: `[polx,poly] = calibrar(imagen1,distancia1,imagen2,distancia2)`, con esto se obtienen los polinomios para calcular las distancias de las demás imágenes capturadas.

Luego, para analizar la distancia de alguna imagen, se utiliza la función evaluarDistancia, que recibe como parámetros la imagen, el polinomio del eje x y el polinomio del eje y, así: distancia = evaluarDistancia(imagen,polx,poly).

```
>> [polx,poly] = calibrar(cincuenta,50,setentaycinco,75)

polx =

    4.1667    -8.3333

poly =

    5.0000    -5.0000

>> distancia = evaluarDistancia(veinticinco,polx,poly)

distancia =

    22.9167

>> distancia = evaluarDistancia(metro,polx,poly)

distancia =

    99.5833

>> |
```