

Medición de distancias entre dos objetos a partir de procesamiento de imagen

Measurement of distances between two objects from image processing

Dennis Augusto Duarte Dick¹
Sidney Domínguez²

Resumen

En base a la necesidad de un software que nos permitiera hacer la medición de distancias entre dos objetos a partir de procesamiento de imagen, se desarrolló un software de control en Labview para la obtención de las distancias en pixel y posteriormente en distancia real. El software se diseñó, con las herramientas que tiene Labview de procesamiento de imagen, para poder obtener estas distancias entre los objetos.

Palabras clave: Procesamiento de imagen, medición de distancia

Abstract

Based on the need for software that would allow us to measure distances between two objects from image processing, a control software was developed in Labview to obtain distances in pixel and then in real distance. The software was designed, with the tools that Labview has for image processing, to obtain these distances between objects.

Key Words: Image processing, distance measurement

Introducción

Antecedentes

El procesamiento de imagen es un método para realizar algunas operaciones en una imagen, en forma de obtener una imagen mejorada o para extraer de forma exitosa la información que contiene. Este es un tipo de procesamiento de señal en la que una imagen es la entrada, y la salida puede ser o una imagen o rasgos/características asociadas con la imagen. Ahora, el procesamiento de imagen se encuentra entre las tecnologías de mayor auge. Esta forma parte, en el área de la investigación dentro de las disciplinas de ingeniería y las ciencias de la computación [1]. El procesamiento de imagen básico incluye los siguientes tres pasos:

- Importar la imagen vía las herramientas de adquisición de imagen.
- Analizar y manipular la imagen.
- La salida en la que el resultado puede ser alterada o reportar que esta se basa en el análisis de la imagen.

Hay dos tipos de métodos usados para el procesamiento de imagen llamados, procesamiento de imagen análogo y digital. El procesamiento de imagen análogo puede ser usado para las copias impresas como impresiones y fotografías. El análisis de imagen se usa en varios fundamentos de interpretación mientras se usa en estas técnicas visuales. Las técnicas de procesamiento de imagen digital ayudan en las manipulaciones de las imágenes digitales mediante el uso de computadoras. Las tres fases que son de todo tipo de datos que deben de someterse al uso de la técnica digital, son el preprocesamiento, mejora y visualización, extracción de in-

¹ Estudiante Facultad de Ingeniería y Tecnología. Universidad de Montemorelos. vemayorz94@gmail.com

² Estudiante Facultad de Ingeniería y Tecnología. Universidad de Montemorelos. sydney@um.edu.mx

ción [2]. La técnica de procesamiento de imagen, en la actualidad se desempeña en diferentes áreas como la medicina, la exploración espacial, la visión por computadora, seguridad, etc. Como resultado directo de la reducción en el precio de las computadoras, el procesamiento digital de imágenes actualmente se puede efectuar (aunque con ciertas limitantes) en una computadora personal [3].

A continuación, se hablará sobre las aplicaciones anteriormente mencionadas del procesamiento de imagen. Comenzaremos hablando un poco sobre la aplicación en el área de la medicina. La más que tiene un auge en esta área es la radiografía digital, que se define como la radiografía de proyección. Mediante el uso de una computadora los datos que son adquiridos del paciente se procesan de manera digital para generar estas imágenes en digital. Estas imágenes posteriormente se muestran para que alguien especializado las pueda interpretar. Si bien la tarea de visualización de imágenes del radiólogo es evaluar la calidad general de la imagen (densidad de imagen, contraste, detalle, ruido, distorsión, artefactos y posicionamiento preciso de la anatomía, etc.), la tarea de visualización de imágenes del radiólogo es principalmente detección de lesiones [4].

El siguiente tema por tratar es la visión por computadora, es el conjunto de herramientas y métodos que permiten obtener, procesar y analizar imágenes del mundo real con la finalidad de que puedan ser tratadas por un ordenador. Esto permite automatizar una amplia gama de tareas al aportar a las máquinas la información que necesitan para la toma de decisiones correctas en cada una de las tareas en las que han sido asignadas [5].

Una vez que se habló sobre el procesamiento de imagen, se hablará sobre la medición de la distancia entre objetos. La transformación de distancia de una característica convierte una imagen digital binaria, que consta de píxeles de característica y no característica, es una imagen en la que todos los píxeles de no característica tienen un valor correspondiente a la distancia al píxel de característica más cercano. Calcular estas distancias es, en

principio, una operación global. Sin embargo, las operaciones globales son prohibitivamente costosas. Por lo tanto, son necesarios algoritmos que consideren solo vecindarios pequeños, pero que den una aproximación razonable de la distancia [6].

Planteamiento del problema:

La espirometría es un examen fisiológico que mide cómo un individuo inhala o exhala volúmenes de aire en un determinado tiempo. La principal señal medida en la espirometría puede ser el volumen o el flujo de aire. Esta prueba como tal es inestimable en la detección de la salud respiratoria general de la misma manera que la presión sanguínea proporciona información importante sobre la salud cardiovascular general [7]. El espirómetro es un dispositivo médico, que nos permite saber el estado actual de nuestros pulmones, y el principal objetivo es conocer la capacidad de los pulmones. A la medición anteriormente mencionado se le conoce como espirometría.

Por lo cual, queremos hacer la implementación reconocimiento de imagen, que a través de este nos permita hacer el rastreo de las posiciones de un marcador colocado en la caja torácica de una persona para poder obtener la oscilación de la caja torácica al realizar el proceso de respiración, y de esta forma tratar de aproximar el volumen de capacidad pulmonar, es decir, realizar una medición espirométrica. a. Pregunta de investigación:

¿Cómo se relacionan las distancias en píxeles de una imagen y las distancias reales de los objetos?

Justificación:

La espirometría es una prueba simple que se usa para ayudar a diagnosticar y controlar ciertas afecciones pulmonares al medir la cantidad de aire que puede expulsar en una respiración forzada. Se lleva a cabo utilizando un dispositivo llamado espirómetro, que es una pequeña máquina conectada por un cable a una boquilla [8].

Considerando la importancia de la evaluación respiratoria se propone un método alternativo en la obtención de una evaluación respiratoria. El objetivo de esta

investigación es desarrollar un nuevo método para realizar espirometrías a partir de obtener la oscilación de la caja torácica por medio del procesamiento digital de imagen, a fin de desarrollar un método alternativo para su aplicación en la clínica práctica y de laboratorio.

Objetivos

Generales.

Desarrollar un software que a través de procesamiento de imagen nos permita relacionar los valores de las distancias en píxeles de una imagen a las distancias reales de los objetos.

Específicos.

1. Adquisición de video en tiempo real, a través de lab-view.
2. Implementar un filtro en el video, para que haya una disminución o nos permita eliminar el ruido.
3. Obtención de la posición de un objeto en tiempo real.
4. Obtener la distancia entre dos marcadores designados.
5. Relacionar la señal de estos píxeles, con una distancia real.
6. Obtención de la gráfica en tiempo real, del movimiento de los marcadores.

Hipótesis:

Es posible hacer un software que, a través de técnicas de procesamiento de imagen, nos permita relacionar los valores de distancias en píxeles en una imagen con distancias reales, y que pueda realizar una gráfica a partir de este comportamiento.

Método

Diseño experimental de medición:

Diagrama de flujo del software:

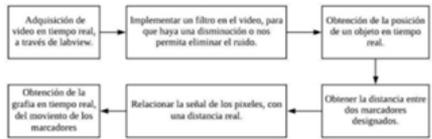


Fig 1. Nos muestra el diagrama de flujo del software.

Adquisición de video en tiempo real: Para la adquisición del video en tiempo real, lo que se realizó en la ventana de programación de labview fue, en la figura 2:

1. función Open Camera.
2. función de Configure Grab.
3. función Close Camera.
4. Seleccionamos la función IMAQ Create.

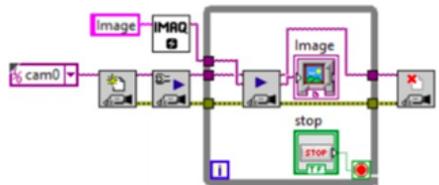


Fig 2. Nos muestra la programación para la adquisición de video.

La función de Open Camera nos permitió, como lo dice el nombre hacer la apertura de la cámara y posteriormente hacer la consulta de los atributos de la cámara, poder cargar un archivo de configuración de la cámara y crear una referencia única a la cámara. Después esas salidas que tiene el Open Camera se conectaron con la función de Configure Grab que nos permite hacer la configuración y empezar la adquisición. Una vez que hace la adquisición esta se repite continuamente en un anillo de búferes. Esta función de Configure Grab se conectó a la función de Grab 2 para poder, hacer la adquisición del fotograma más actual en Image Out. Posteriormente la salida ya mencionada, que es Image Out

se le conecto un display, para poder ver la adquisición en tiempo real la cual se ve en la Figura 3.

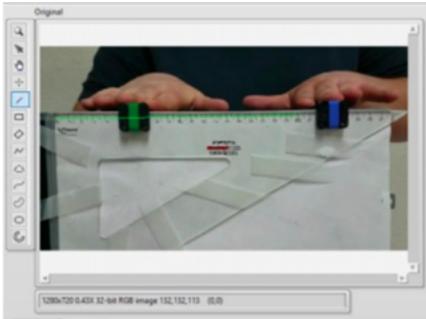


Fig 3. Se muestra la interfaz de la adquisición de video en tiempo real.

Implementar un filtro en el video:

Para hacer la implementación de un filtro, en labview se realizó lo que muestra en la Figura 4. Esta es la parte del programa que nos permite hacer el filtrado de los colores.

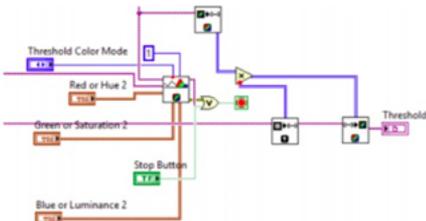


Fig 4. Nos muestra la parte del código que hace el filtrado de colores.

Esto se hizo gracias a las funciones de color threshold, color to image array, image to array and array to color image. El color threshold aplica un umbral (threshold) a los tres planos de una imagen RGB o HSL y coloca el resultado en una imagen de 8 bits. El color to image array nos permitió extraer los píxeles de la imagen en color o de parte de la imagen en color en una matriz 2D. El image to array nos permitió extraer la copia de los píxeles de una imagen, o parte de una imagen, en una matriz LabVIEW 2D. El array to color image nos permitió crear la imagen en color a partir de una matriz 2D. En la Figura 5 podemos ver cómo es la interfaz gráfica para el threshold.

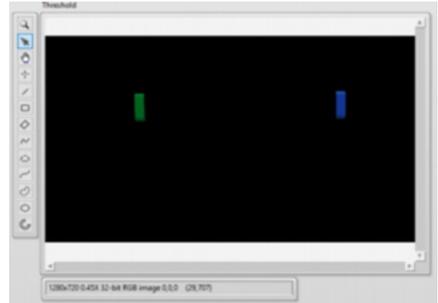


Fig 5. Nos muestra la interfaz gráfica del threshold.

Obtención de la posición de un objeto en tiempo real: Para poder obtener la posición de estos marcadores, primero fue necesario crear las plantillas para que el programa hiciera la detección de estos. Posteriormente se usó la función set up color pattern matching que nos permite entrenar al programa para poder hacer la detección de estos objetos. Lo siguiente fue usar la función de match pattern, para poder hacer la búsqueda de los colores de estos objetos y posteriormente poder con esto obtener el número de matches posibles. Por último, se generó un case y dentro de él un cluster que nos daba la posición y el ángulo, y este a su vez va conectado con otro cluster que acomoda los datos por nombre. Al final de este, a la salida se le crea un indicador que nos muestra su posición. En la Figura 6 podemos ver el indicador que nos da la posición, en cada uno de los ejes respectivamente.

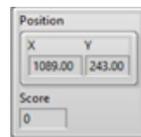


Fig 6. Nos muestra la posición de los objetos en los respectivos ejes.

Obtener la distancia entre dos marcadores designados: Para poder obtener la distancia en píxeles de estos marcadores, se usaron los anteriores, además de usar una herramienta que cuando nos rastree los objetos, nos dibuje un contorno. Después la información de los matches, se conecta a un array de indexación y este se conecta a un cluster para obtener un bounding box. Después una de las salidas

de cada uno de este array de indexación está conectado con una herramienta que se llama overlay line que nos permite dibujar una línea entre los objetos. Por último, las otras salidas de estos arrays de indexación, van conectados a un build array, después este se conecta con imaq point distances, que es la herramienta que nos va a permitir hacer la medición entre estos dos objetos. Además, este último se conecta a un index array para posteriormente poder obtener la distancia.

Relacionar la señal de estos píxeles, con una distancia real: Para poder obtener la relación entre los píxeles y la distancia real, lo primero que se hizo fue en la imagen que se está adquiriendo, se creó una región de interés para conocer las coordenadas en la que está colocada esta región de interés que se puede ver en Figura 2. Una vez que se consiguen las coordenadas de los píxeles, una vez con esta información, lo que se hace después es una relación entre los datos de los píxeles que tenemos con la distancia real que vamos a adquirir. En la Figura 7 se muestra la relación entre los píxeles y la distancia real.



Fig 7. Nos muestra la obtención de la relación entre píxeles y la distancia en cm.

Obtención de la gráfica en tiempo real, del movimiento de los marcadores: Una vez que ya se obtuvieron los datos de las distancias en píxeles y la real, se generó una gráfica en tiempo real. El gráfico de forma de onda es un tipo especial de indicador numérico que muestra una o más gráficas de datos que normalmente se adquieren a una velocidad constante [32]. El siguiente panel frontal muestra un ejemplo de un gráfico de forma de onda. Para obtenerla en la programación, la distancia en píxeles se multiplica por la de cm, y el resultado es la gráfica en tiempo real. En la Figura 8 se muestra la gráfica.

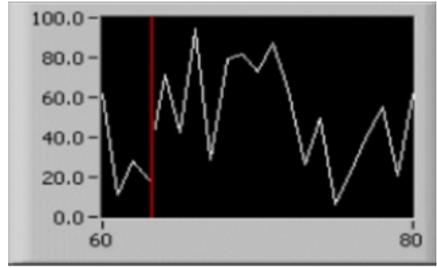


Fig 8. Nos muestra la gráfica.

Diseño de evaluación de software:

Para llevar a cabo la evaluación, lo que se realizó fue la visualización del software en funcionamiento. Lo primero que se realizó fue la adquisición del video para poder después convertirlo en formato avi, que es el formato de lectura de videos de labview. Ya que se tuvo, se hizo una modificación al código para que este hiciera la lectura de los videos y posteriormente los reprodujera. Lo segundo que se realizó fue observar en el programa, que el filtro de color a través del threshold, solo nos mostrara el color verde y azul, que fueron los que se escogieron para hacer la prueba. Lo tercero fue visualizar si el programa nos estaba entregando bien las posiciones de los objetos, en sus respectivos ejes. Lo cuarto fue verificar que las distancias en píxeles fueran las correctas y posteriormente verificar que las distancias reales en cm fueran las correctas. Lo quinto y lo último fue verificar que el movimiento en el video se estuviera graficando en tiempo real, en este punto se decidió que una vez que el video dejara de reproducirse nos mostrara todo el trazado de la gráfica, y que los datos obtenidos de las distancias y el graficado se pudieran guardar en un archivo xlsx, que es el formato de Excel.

Resultados

De acuerdo con lo estipulado, la adquisición de video en tiempo real se llevó exitosamente gracias a las herramientas de procesamiento de imagen de LabVIEW el cual se puede ver en la Figura 3.

Después la realización del filtro de colores a través del threshold, nos permitió obtener sólo los colores que se deseaban, para que cuando se desee realizar la prueba

ba, no haya algún otro color que interfiera con la medición. Figura 5 nos muestra el filtro de threshold.

Lo siguiente fue la obtención de la posición en tiempo real. Nos permitió saber la posición de estos objetos en el eje x, tanto como en el eje y. Esto se muestra en la Figura 9.

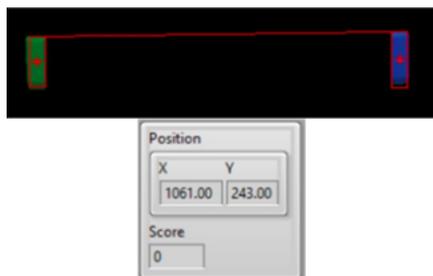


Fig 9. Nos muestra la obtención en tiempo real en los respectivos ejes.

Posteriormente, la obtención de la distancia entre dos objetos designados nos permitió saber cuál es la distancia en pixeles entre estos dos marcadores, se puede ver la muestra en la Figura 10.

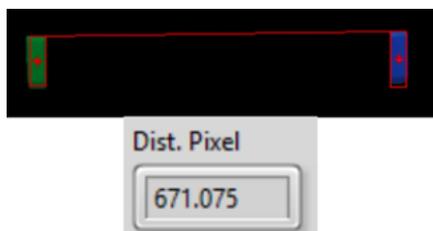


Fig 10. Nos muestra la obtención de la distancia entre los dos objetos

De acuerdo con lo estipulado, se puede obtener la relación entre la distancia de pixeles con una distancia real en cm, esto nos permitió obtener la distancia real en cm a través de una relación entre los pixeles y la distancia en cm, esto se muestra en la Figura 11.

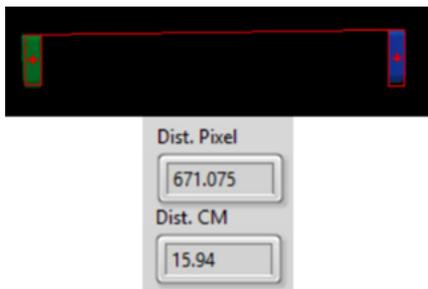


Fig 11. Nos muestra la obtención de distancia en cm con relación a la distancia en pixeles.

Y, por último, se hizo la obtención de la grafica en tiempo real del movimiento de los marcadores. La grafica nos permitió saber el comportamiento de los objetos cuando están en movimiento. Se muestra en la Figura 12 y 13.

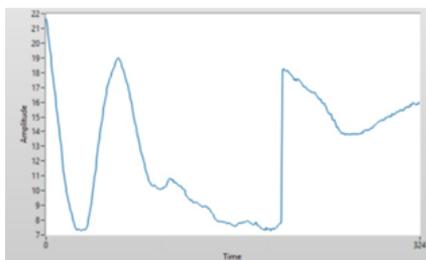


Fig 12. Nos muestra el trazado de la gráfica en tiempo real.

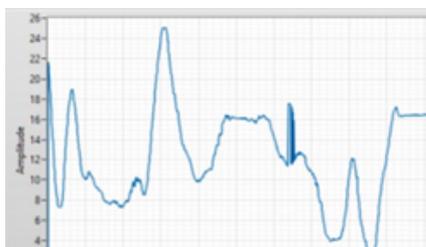


Fig 13. Nos muestra el trazado completo de la gráfica, después de la terminación del video.

Conclusiones

El proyecto concluyó que el funcionamiento del software permite la viabilidad en el uso para hacer diversas aplicaciones del movimiento de dos objetos y conocer la distancia entre ellos. El software que se realizó es la base para los trabajos a futuro. Para que después se le realicen mejoras tales como, calibraciones, nuevas funciones; que nos permitan trabajar en las capturas de campo que se necesitan, para comenzar la prueba espirométrica con la que se está proponiendo en proyectos a futuro, y saber si esta prueba es una buena alternativa.

Referencias

1. J. Russ, "The image processing handbook," 2016.
2. University of tartu, "1. Introduction to image processing | Digital Image Processing," January 2014, 2014. [Online]. Available: <https://sisu.ut.ee/imageprocessing/book/1>. [Accessed: 07-May-2019].
3. A. Universidad Nacional Autónoma de México. Centro de Investigaciones y Servicios Educativos and Universidad Nacional Autónoma de México. Instituto de Investigaciones sobre la Universidad y la Educación, Perfiles educativos., no.72. Instituto de Investigaciones sobre la Universidad y la Educación, 1996.
4. E. Seeram, "Digital Image Processing Concepts," in Digital Radiography, Singapore: Springer Singapore, 2019, pp. 21–39.
5. INFAIMON, "Visión por computador: qué es y cuáles son sus usos más comunes," INFAIMON S.L., 2018.
6. G. Borgefors, "Distance transformations in digital images," Comput. Vision, Graph. Image Process., vol. 34, no. 3, pp. 344–371, Jun. 1986.
7. M. R. Miller, "Standardisation of spirometry," Eur. Respir. J., vol. 26, no. 2, pp. 319–338, Aug. 2005.
8. R. Benítez-Pérez, ... L. T.-B.-N. y cirugía, and undefined 2016, "Espirometría: recomendaciones y procedimiento," medi-graphic.com.
9. KidsHealth, "Espirometría (para Padres) - KidsHealth," KidsHealth, 2015. [Online].
10. KidsHealth, "Espirómetro (para Niños) - KidsHealth," The Nemours Foundation.
11. Maria De Girodmedial, "Qué es un espirómetro y para qué sirve - El blog de Girodmedical | Magento Title !," GirodMedical, 2017. [Online]. Available: https://www.girodmedical.es/blog_es/espirometro-para-que-sirve/. [Accessed: 13-May-2019].
12. R. E. Sociedad Mexicana de Estudios sobre Tuberculosis y Enfermedades del Aparato Respiratorio. et al., Neumología y cirugía de tórax., vol. 75, no. 2. Unidad de Patología, Sanatorio de Huipulco, 2016.

13. How It Works Team, "The human ribcage - How It Works," How It Works Team , 2013. [Online]. Available: <https://www.howitworksdaily.com/the-human-ribcage/>. [Accessed: 13-May-2019].
14. Universidad de Cantabria, "G367: Tema 2. Mecánica respiratoria," Universidad de Cantabria, 2017. [Online]. Available: <https://ocw.unican.es/mod/page/view.php?id=552>. [Accessed: 13-May-2019].
15. Daniel Shiffman, "Images and Pixels Processing.org," Processing , 2008. [Online]. Available: <https://processing.org/tutorials/pixels/>. [Accessed: 13-May-2019].
16. P. M. Athanas and A. L. Abbott, "Image processing on a custom computing platform," Springer, Berlin, Heidelberg, 1994, pp. 156-167.
17. H. Chen, D. D. Hughes, T.-A. Chan, J. W. Sedat, and D. A. Agard, "IVE (Image Visualization Environment): A Software Platform for All Three-Dimensional Microscopy Applications," J. Struct. Biol., vol. 116, no. 1, pp. 56-60, Jan. 1996.
18. EcuRed, "Modelo RGB - EcuRed," EcuRed. [Online]. Available: https://www.ecured.cu/Modelo_RGB. [Accessed: 14-May-2019].
19. Isabel, "Módulo de color CMYK - blogartesvisuales," Módulo de color CMYK, 2007. [Online]. Available: <https://www.blogartesvisuales.net/disenografico/color/modelo-de-color-cmyk/>. [Accessed: 14-May-2019].
20. Antonio Herrera, "Modelos de color (RGB, CMYK, HSV/HSL) | Antonio Herrera," Modelos de color (RGB, CMYK, HSV/HSL), 2015.
21. M. Irani, B. Rousso, and S. Peleg, "Detecting and tracking multiple moving objects using temporal integration," Springer, Berlin, Heidelberg, 1992, pp. 282-287.
22. M. O.-B. universidad A. de P. http and undefined 2013, "Procesamiento digital de imágenes," mmartin.cs.buap.mx.
23. J. Elizondo, L. M.-D. U. Autónoma, and undefined 2002, "Fundamentos de procesamiento de imágenes," academia.edu.
24. Adrien Jolly, "LabVIEW que es? - Ingeniero especialista en probadores, medición, experto en LabVIEW, TestStand, Lab- Windows/CVI," 2007.
25. U. Manual, "TM IMAQ Vision for LabVIEW," North, vol. 88, no. 371007, pp. 331-339, 2004.
26. P. F. Felzenszwalb and D. P. Huttenlocher, "Distance Transforms of Sampled Functions," Theory Comput., vol. 8, no. 1, pp. 415-428, 2012.
27. G. Borgefors, "Hierarchical chamfer matching: a parametric edge matching algorithm," IEEE Trans. Pattern Anal. Mach. Intell., vol. 10, no. 6, pp. 849-865, 1988.
28. D. P. Huttenlocher, G. A. Klanderman, and W. J. Rucklidge, "Comparing images using the Hausdorff distance," IEEE Trans. Pattern Anal. Mach. Intell., vol. 15, no. 9, pp. 850-863, 1993.
29. H. Blum, "Biological shape and visual science (part I)," J. Theor. Biol., vol. 38, no. 2, pp. 205-287, Feb. 1973.
30. R. Unnikrishnan, C. Pantofaru, and M. Hebert, "Toward Objective Evaluation of Image Segmentation Algorithms," IEEE Trans. Pattern Anal. Mach. Intell., vol. 29, no. 6, pp. 929-944, Jun. 2007.
31. X.-Y. Wang and J. Bu, "A fast and robust image segmentation using FCM with spatial information," Digit. Signal Process., vol. 20, no. 4, pp. 1173-1182, Jul. 2010.
32. Labview, "LabVIEW Fundamentals Manual," no. August, 2005.