Sistema de monitoreo remoto con detección de movimiento basado en visión por computadora

Alejandro Monroy Reyes[1], Ariadna M. Estrada Gaspar[1], Francisco J. Enríquez Aguilera[2], Gabriel Bravo Martínez[2], René Noriega Armendáriz[2], Javitt Higmar Nahitt Padilla Franco[2]

[1]Departamento de Ingeniería Industrial y Manufactura, Universidad Autónoma de Ciudad Juárez.
[2]Departamento de Ingeniería Eléctrica y Computación, Universidad Autónoma de Ciudad Juárez.

Resumen

En este trabajo se presenta un sistema de monitoreo remoto con detección de movimiento basado en visión por computadora. El sistema está compuesto por una cámara web con conexión USB montada en una plataforma de dos servomotores para controlar el giro horizontal y vertical. El algoritmo de visión artificial fue implementado utilizando el ambiente gráfico de LabVIEW y el Toolkit Vision Aquisition. Además, se utilizó un microcontrolador Arduino Uno para controlar el movimiento de los servomotores. El algoritmo es capaz de detectar movimiento en un ambiente cerrado y con iluminación controlada mediante sustracción de imágenes consecutivas y comparación con un factor de sensibilidad. La utilización de LabVIEW como ambiente de programación permite la modularización del algoritmo, una implementación simple y una interfaz atractiva para el usuario donde se pueden hacer modificaciones con facilidad. Finalmente se utilizó la herramienta de publicación Web de LabVIEW para subir la aplicación en un servidor y acceder a ella de forma remota desde una computadora personal a distancia o dispositivos móviles como teléfonos inteligentes y tabletas.

Palabras clave: LabVIEW, Arduino, Visión, servomotores y monitoreo remoto.

Introducción

Los sistemas de monitoreo convencionales, generalmente consisten de una cámara de video transmitiendo y una o varias persona monitoreando las 24 horas del día o por largos periodos de tiempo. Sin embargo, en este tipo de sistemas de vigilancia influye el factor humano (cansancio, distracciones, entre otros) por lo que no son confiables al cien por ciento. Debido a lo mencionado anteriormente la detección de movimiento representa un elemento clave en aplicaciones de vigilancia y seguridad hoy en día. La información más importante que debe ser observada por los usuarios del sistema de monitoreo es el cambio o movimiento entre imágenes sucesivas; esto se puede realizar sobre una serie de imágenes estáticas donde se capturé solo la información más relevante. De igual forma, es esta la información que se utiliza en el proceso de toma de decisiones, ya sea de un humano o de un sistema computarizado (Lita, Vissan, Ion. 2010), por lo tanto, la posibilidad de detectar movimiento automáticamente permite que la intervención humana se reduzca considerablemente y que la información tenga un mayor grado de confiabilidad (Mao Xiaobo, Yang Jing, 2011).

La detección de movimiento se puede lograr por numerosos métodos, siendo los más simples los enfoques que utilizan

sensores mecánicos conectados a un dispositivo de alerta. Este proyecto se basa en el procesamiento de imágenes y se utiliza el método por diferencia de dos imágenes consecutivas. Dicho método permite separar un objeto en movimiento del fondo estático al realizar una resta a nivel de pixeles (Yupeng Xu, Feihong Yu, 2013).

La sustracción entre dos imágenes consiste en restar a la imagen actual, la imagen anterior pixel por pixel. Si los pixeles tienen el mismo valor, quiere decir que no hubo ningún cambio en esa región y el resultado de la resta es cero. Por lo tanto, cuando el valor de la resta es diferente de cero, se puede inferir que hubo movimiento (Wang Lei, Shen Yuming, 2010). No obstante, el resultado obtenido debe ser procesado posteriormente para determinar si el cambio es sustancial como para considerarlo movimiento y no confundirlo con ruido en la imagen (Mehrubeoglu, M., Linh Manh Pham, 2011).

La aplicación presentada a continuación consiste en un sistema de monitoreo al que el usuario puede acceder por medio de una dirección web y modificar la posición de la cámara a través de la interfaz de usuario. Al modificar la posición de la cámara, el usuario es capaz de monitorear distintas posiciones y ángulos de su hogar, oficina, etc. Al detectar un movimiento, el sistema captura automáticamente la imagen de la cámara creando un historial para su evaluación posterior. La interfaz permite controlar la ruta de almacenamiento y la cantidad de imágenes máxima a almacenar. Si el límite de imágenes es alcanzado, la imagen más antigua se elimina y se guarda la más reciente.

En las siguientes secciones se presentan los materiales y métodos utilizados para el desarrollo del sistema. Además, se presenta la interfaz de usuario de forma detallada y se explica cada uno de los controles e indicadores. Por otra parte, se explica la comunicación de software entre LabVIEW y arduino para controlar el giro horizontal y vertical de los servomotores, así como la conexión física entre ellos. También, se explica detalladamente el algoritmo para la detección de movimiento utilizando *NI Vision and Motion*. Finalmente, se muestran los resultados obtenidos de la experimentación y pruebas del sistema así como las discusiones y trabajos a futuro que surgieron de la realización del proyecto.

Materiales y Métodos

Arquitectura del sistema

La figura 1 muestra un diagrama de las entradas y salidas del sistema de monitoreo remoto propuesto. Como entradas, se obtienen imágenes por medio de una cámara con conexión a PC mediante la interfaz de comunicación USB y comandos del usuario. Es necesario que la PC tenga suficiente espacio en el disco duro para almacenar las imágenes obtenidas por el movimiento detectado. Cabe mencionar, que la PC debe tener acceso a internet para permitir el monitoreo remoto.

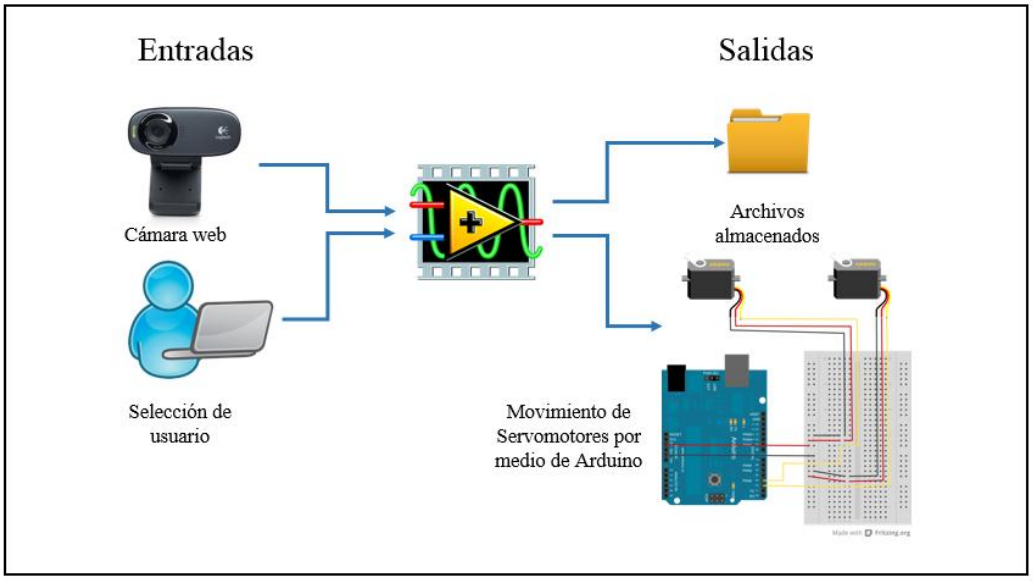

Figura 1. Entradas y salidas del Sistema.

Por otra parte, la cámara está montada sobre una plataforma de dos servomotores y le permiten girar en el eje vertical y horizontal para cubrir un mayor ángulo de visión. El usuario es capaz de controlar el giro de los servomotores mediante botones similares a las flechas de un teclado, además de poder guardar posiciones y acceder a ellas, por medio de tres memorias. Por lo tanto, las salidas constan de dos variables que controlan el ángulo de los servomotores y las imágenes almacenadas. En la figura 2 se puede observar un diagrama de flujo que muestra el proceso que se realiza para la detección del movimiento.

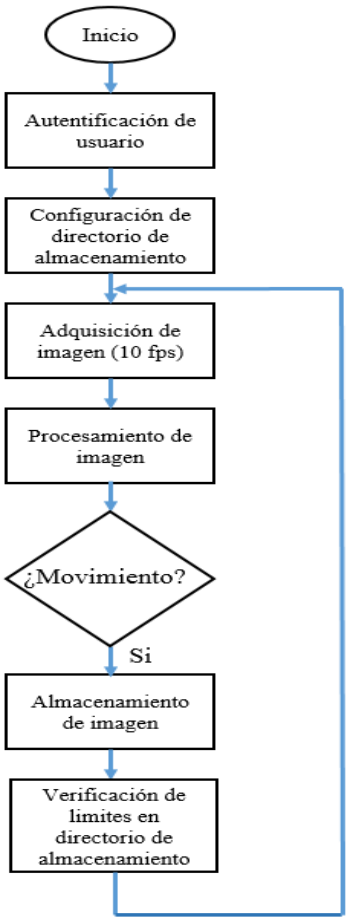

Figura 2. Diagrama de flujo de la detección de movimiento.

Interfaz de usuario

Como en la mayoría de las aplicaciones y sistemas, es necesario que solo el dueño de la información pueda acceder a ella con el fin de que la información este segura y no llegue a manos desconocidas. Debido a lo anterior, se agregó al sistema una pantalla de inicio de sesión en la cual el usuario introduce su nombre de usuario y contraseña, una vez validada su identidad el usuario podrá hacer uso de la aplicación de monitoreo remoto.

La interfaz de monitoreo se diseñó con el fin de dar facilidad al usuario de vigilar desde distintos ángulos y perspectivas algún punto de su interés. Por lo tanto, es necesario que el usuario pueda controlar la posición de la cámara de una forma sencilla. Para realizar el movimiento de la cámara el usuario puede presionar cada una de las flechas de dirección presentes en la interfaz, ya sea para mover la cámara hacia la derecha, izquierda, arriba, abajo, así como también un botón de *HOME* el cual regresa al usuario a la posición inicial de la cámara. Además, el usuario tiene la posibilidad de guardar las posiciones de la cámara que más le interesen, para esto basta con colocar la cámara en la posición deseada y después presionar los botones guardar. Para acceder a la posición guardada el usuario debe presionar cada uno de los botones etiquetados como M1, M2 y M3, al presionar alguno de los botones de memoria la cámara se posicionara automáticamente en la posición de monitoreo deseada.

Por otra parte el usuario puede ajustar la sensibilidad de detección, la cual indica si el movimiento debe ser poco o mucho para ser clasificado y detectado como un movimiento. La sensibilidad de movimiento se encuentra en un rango de 0 a 1, siendo 0 mayor sensibilidad y 1 la menor sensibilidad. Para mostrar al usuario que se ha detectado un movimiento se incluyó un indicador con una luz verde.

Además, el usuario puede decidir el número de imágenes que desea guardar en su computadora y del lado derecho de la interfaz se puede consultar un historial de cada una de las imágenes que han sido capturadas por el sistema. Al llegar al número máximo de imágenes el sistema automáticamente sobrescribe la primera imagen guardada por una más reciente. Lo anterior fue de importancia clave debido a que si se considera el peor de los casos en el que el sistema capture 10 imágenes por segundo, en un día se consumirían alrededor de 55 GB de almacenamiento.

Finalmente, en la figura 3 se muestra la interfaz de usuario implementada.

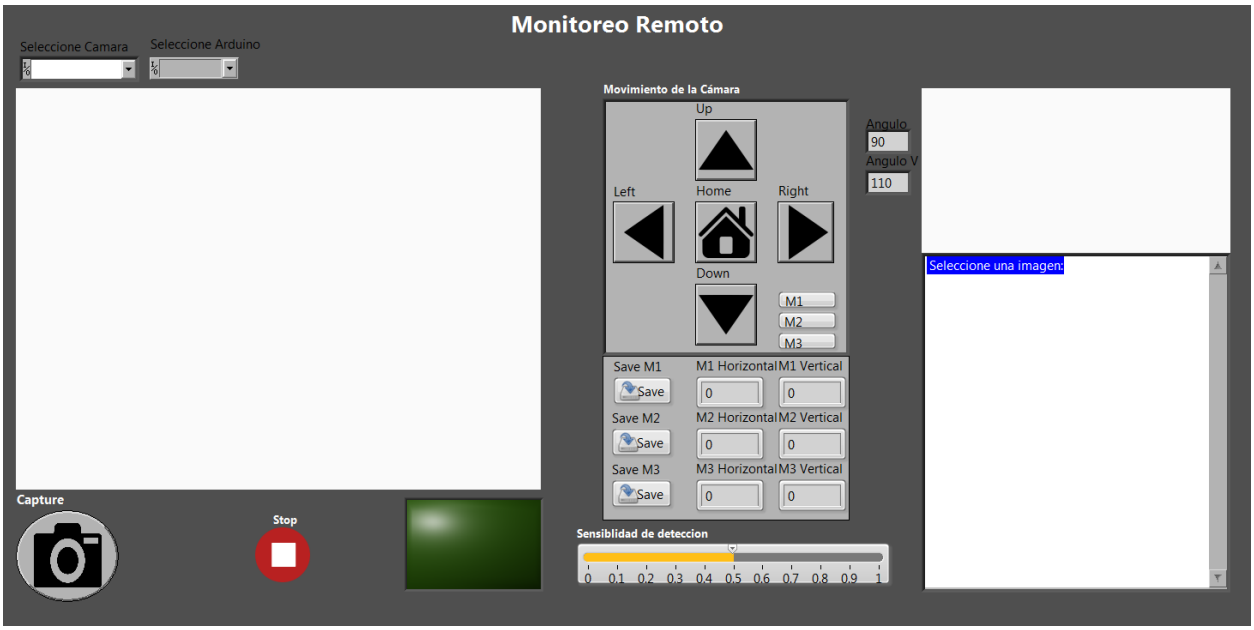

Figura 3. Interfaz de Usuario.

Movimiento de la cámara

El movimiento de la cámara es controlado por el usuario a través de controles en la interfaz. Se utilizó una programación basada en eventos que permite detectar cada vez que alguno de los botones de la interfaz es presionado. Cada vez que se oprime uno de los botones para mover la cámara, se aumenta o se disminuye en 5 grados el ángulo del servomotor en el eje seleccionado. Los límites de los servomotores son 180 grados en el movimiento horizontal y 150 grados en el movimiento vertical.

Para lograr la comunicación entre LabVIEW y los servomotores se utilizó una tarjeta Arduino Uno. LabVIEW cuenta con un toolkit para programar el Arduino con bloques y se puede descargar de forma gratuita.

La implementación del movimiento a través de Arduino consta de cuatro fases: abrir la conexión por puerto serial (COM), configurar los pines de salida, enviar el valor del ángulo a través del puerto serial y cerrar la conexión. Es necesario verificar que el consumo de corriente de los servomotores sea compatible con la corriente que otorga el Arduino, en caso de que no sea suficiente se debe de utilizar un circuito de acondicionamiento.

Uno de los inconvenientes que se presentaron al mover la cámara fue que al momento de girar, la sustracción de imágenes sigue funcionado y se detecta movimiento. Aunque la detección es correcta, el almacenamiento de imágenes no es necesario puesto que el usuario sabe que la cámara se está moviendo.

Algoritmo de detección de movimiento

Para realizar el algoritmo de detección de movimiento es necesario tener dos imágenes, la imagen anterior tomada por la cámara y la imagen actual. De las dos

imágenes originales en formato RGB (Red, Green, Blue) obtenidas desde la cámara, se obtuvo su representación en escala de grises con el fin de trabajar solo en un plano de la imagen y no en tres.

Utilizando las dos imágenes en escala de grises se realizó una operación de sustracción, en la cual a la imagen actual se le resto la imagen anterior (Imag(N) – Imag(N-1)). El resultado obtenido de la resta de dos imágenes consecutivas muestra la diferencia de cada uno de los valores de los pixeles, con lo cual, si la diferencia de cada uno de los valores es 0, se obtiene una imagen completamente negra que indica que no hubo un cambio o movimiento. En el caso contrario al provocarse un movimiento los valores de los pixeles son diferentes de cero por lo que se puede asumir que hubo un cambio.

Debido a las características de las cámaras, la iluminación del ambiente y otros factores externos, es casi imposible que dos imágenes consecutivas sean exactamente iguales, por lo que la diferencia entre los valores de los pixeles de las dos imágenes no es cero. Una vez obtenida la resta de las imágenes, fue necesario aplicar una operación morfológica de erosión, en la cual se utiliza un elemento estructural que provoca que los pixeles que se encuentran aislados desaparezcan. De este modo es posible despreciar todas las imágenes en las que no haya un movimiento real. En la figura 4 se puede observar una imagen detectada antes de aplicar la operación morfológica de erosión, en la cual se observan algunos puntos aislados en la imagen. Después de la operación de erosión esos puntos aislados se eliminan, obteniendo una imagen en su mayoría negra.

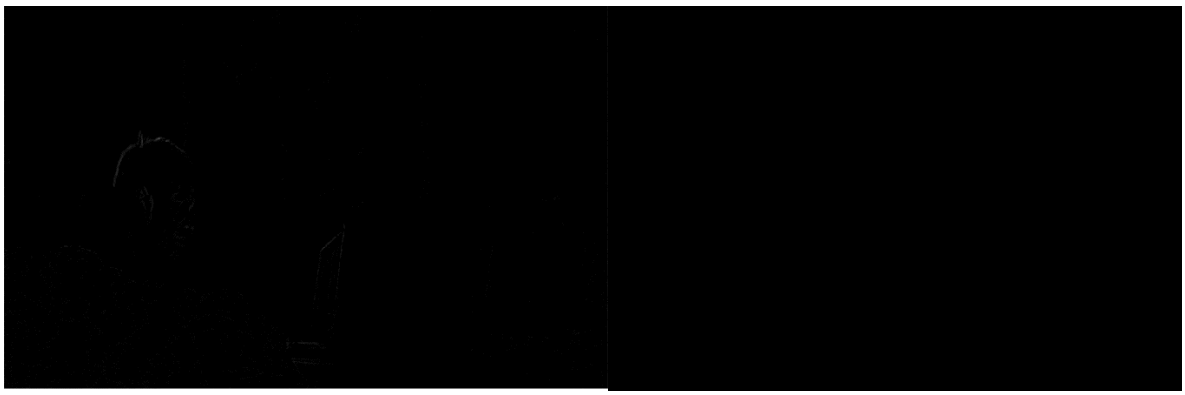

Figura 4. Imagen antes y después de la operación de erosión.

Finalmente, por medio del histograma de la imagen, el cual es una representación gráfica de la distribución de los pixeles en la imagen, se obtuvo el promedio de la imagen, de esta forma si el promedio de los pixeles es mayor a un factor de sensibilidad dado por el usuario, la imagen es considerada con cambios suficientes para determinar movimiento y es guardada, de lo contrario la imagen es despreciada ya que no contiene información útil. En la figura 5 se muestra una imagen en la cual se detectó un movimiento.

Figura 5. Detección de Movimiento.

Finalmente en la figura 6 se muestra una imagen del prototipo final.

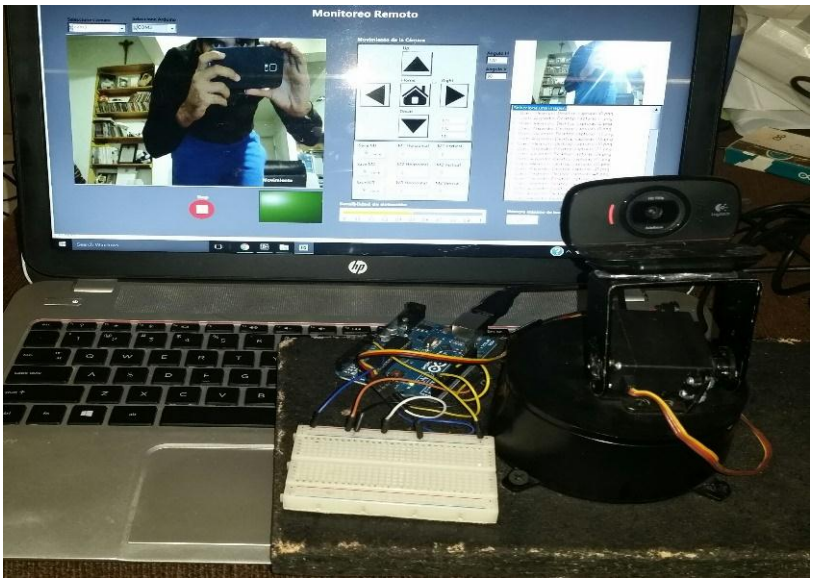

Figura 6. Prototipo final

Resultados y Discusión

Para validar el funcionamiento del algoritmo de detección de movimiento se realizaron pruebas en un ambiente cerrado y con iluminación controlada. Se utilizaron diferentes valores para el factor de sensibilidad y se encontró que mientras sea

más cercano a cero, la detección de movimiento es más sensitiva ya que incluso alcanza a detectar parpadeos.

Por lo tanto, para encontrar el valor óptimo de sensibilidad se debe hacer una evaluación previa del movimiento esperado y la posición de la cámara. Si se desea detectar un movimiento muy cercano a la cámara, este es detectado con mayor facilidad puesto que el objeto en movimiento abarca más pixeles y al realizar la sustracción de imágenes se encontrara un cambio mayor. Si el movimiento esperado ocurre alejado de la cámara, el objeto en cuestión ocupará menos área en la imagen y será más difícil de detectar.

Por medio de experimentación se encontró que si el caso esperado es que se detecte movimiento en objetos cercanos a la cámara, se debe utilizar un factor de sensibilidad de 0.5 a 1 para asegurar la funcionalidad del algoritmo. Por otro lado, si el movimiento esperado es lejano a la cámara se deben utilizar factores de sensibilidad de 0 a 0.5.

Conclusiones

La realización de un sistema de monitoreo remoto tiene una aplicación real en tareas de vigilancia. Mediante la realización de este proyecto se aplicaron los conocimientos técnicos obtenidos en el diplomado de programación grafica en LabVIEW, tales como la utilización de estructura de eventos, ciclos while, ciclos for, shift register, manejo de archivos, clúster de error, además del diseño de interfaces de usuario. Además de obtener nuevos conocimientos sobre la utilización de las librerías de Arduino y la adquisición y procesamiento de imágenes de una cámara.

Se creó una aplicación capaz de detectar movimiento en ambientes controlados y puede ser utilizada en operaciones de vigilancia y seguridad. La utilización de LabVIEW permitió crear una aplicación intuitiva para el usuario y fácil de modificar y mejorar. Como posibles inconvenientes de la aplicación, se encontró que al cambiar de posición la cámara el algoritmo detecta el movimiento y guarda las imágenes, un comportamiento que podría ser eliminado en un trabajo a futuro.

Referencias

Lita Ioan, Visan Daniel, Ion Cioc, 2010, "LabVIEW Application for Movement Detection Using Image Acquisition and Processing", Romania, IEEE 16th International Symposium for Design and Technology in Electronic Packaging (SIITME), pp. 23 – 26;

Thomas Anupama, Ashraf Asiya, Lal Lini, Mathew Minju, M. Jayashree, 2011, "Security Enhancement Using Motion Detection", India, International Conference on Signal Processing, Communication, Computing and Networking Technologies (ICSCCN), pp 552 – 557.

Yupeng Xu, Feihong Yu, 2013, "Research on the Image Acquisition and Camera Control of Machine Vision Camera Based on LabVIEW", 5th International Conference on Intelligent Human-Machine Systems and Cybernetics (IHMSC), pp. 26-27.

Wang Lei, Shen Yuming, 2010, "Design of Machine Vision Applications in Detection of Defects in High-Speed Bar Copper", International Conference on E-Product E-Service and E-Entertainment (ICEEE), pp.1-4.

Mehrubeoglu, M., Linh Manh Pham, Hung Thieu Le, Muddu R, Dongseok Ryu, 2011, "Real-time eye tracking using a smart camera", IEEE in Applied Imagery Pattern Recognition Workshop (AIPR), pp.1-7.

Mao Xiaobo, Yang Jing, 2011, "Research on object-background segmentation of color image based on LabVIEW", IEEE International Conference on Cyber Technology in Automation, Control, and Intelligent Systems (CYBER), pp.190-194.