Artículo **Revista de Tecnologías Computacionales**
Septiembre 2018 Vol.2 No.7 8-18

Sistema de visión embebido para detección de movimiento de forma remota utilizando el internet de las cosas

Embedded vision system for remote motion detection using the internet of things

LARDIZÁBAL-LÓPEZ, David†*, GALLEGOS-BAÑUELOS, José Luis Guillermo, FLORES-DOMÍNGUEZ, Bardo Eugenio y LÓPEZ-HERRERA, Jesús José Nicolás

Tecnológico Nacional de México / Instituto Tecnológico de Chihuahua

ID 1er Autor: *David, Lardizábal-López* / **ORC ID**: 0000-0001-6142-5342, **Researcher ID Thomson**: G-3124-2018, **CVU CONACYT ID**: 900345

ID 1er Coautor: *José Luis Guillermo, Gallegos-Bañuelos* / **ORC ID**: 0000-0002-4804-6273, **Researcher ID Thomson**: G-6185-2018, **CVU CONACYT ID**: 455103

ID 2do Coautor: *Bardo Eugenio, Flores-Domínguez* / **ORC ID**: 0000-0003-2101-3786, **Researcher ID Thomson**: G-5964-2018, **CVU CONACYT ID**: 457518

ID 3er Coautor: *Jesús José Nicolás, López-Herrera* / **ORC ID**: 0000-0001-8612-7261, **Researcher ID Thomson**: G-6518-2018, **CVU CONACYT ID**: 091818

Recibido: Junio 26, 2018; Aceptado: Agosto 23, 2018

Resumen

En la actualidad se utilizan sistemas de visión para automatizar o mejorar un proceso, en esta propuesta se usará para asistencia en video vigilancia remota, el cual disminuye el error que pueda tener un sistema al monitorear cámaras de seguridad. El sistema está basado en software libre y sistemas embebidos, es capaz de detectar errores generados por el factor humano, por ejemplo, distracción del vigilante, baja resolución en pantalla para distinguir actividad en la observabilidad de las cámaras. Para solucionar este tipo de problemas se implementó un sistema de visión en el cual se utilizaron tres algoritmos de detección de movimiento denominados Resta, MOG y MOG2 a los cuales se le aplicaron pruebas de luz y actualización de fondo para obtener el algoritmo más adecuado. El sistema detecta si un intruso entra en un área prohibida o si algún objeto es removido. El sistema de visión encierra en un rectángulo de color negro el área donde se encontró actividad alertando al usuario para que tome las medidas necesarias.

Videovigilancia remota, Sistema de visión, Sistema embebido

Abstract

At present, vision systems are used to automate or improve a process, in this proposal it will be used for remote video surveillance assistance, which decreases the error that a system can have when monitoring security cameras. The system is based on free software and embedded systems, is able to detect errors generated by the human factor, for example, distraction of the watchman, low resolution on the screen to distinguish activity in the observability of the cameras. To solve this type of problems, a vision system was implemented in which three motion detection algorithms called Resta, MOG and MOG2 were used, to which light and background update tests were applied to obtain the most suitable algorithm. The system detects if an intruder enters a prohibited area or if an object is removed. The vision system encloses in a black rectangle the area where activity was found, alerting the user to take the necessary measures.

Remote video surveillance, Vision system, Embedded system

Citación: LARDIZÁBAL-LÓPEZ, David, GALLEGOS-BAÑUELOS, José Luis Guillermo, FLORES-DOMÍNGUEZ, Bardo Eugenio y LÓPEZ-HERRERA Jesús José Nicolás. Sistema de visión embebido para detección de movimiento de forma remota utilizando el internet de las cosas. Revista de Tecnologías Computacionales. 2018. 2-7: 8-18.

*Correspondencia al Autor (Correo Electrónico: dlardizabal@itchihuahua.edu.mx)
† Investigador contribuyendo como primer autor.

Introducción

En la actualidad muchas empresas, comercios, escuelas, etc., usan sistemas de vigilancia con cámaras monitoreadas por personas en una sala de control.

Un problema es que al ser establecimientos con zonas muy grandes se necesitan demasiadas cámaras para vigilar toda la zona y por lo tanto se requieren también demasiados monitores para que el vigilante pueda ver la zona que observa la cámara.

Esto ocasiona que los costos del sistema de vigilancia aumenten, anteriormente para reducir este costo se multiplexeaba el sistema de cámaras en un solo monitor para cambiar la visibilidad de las diferentes cámaras, pero eso hace que solo se pueda ver una cámara a la vez lo que reduce la capacidad de visión del vigilante, una solución que se le dio a ese problema fue dividir el monitor en 4 para poder observar varias cámaras a la vez, pero esto reducía la resolución de la cámara en el monitor por lo que el vigilante no puede observar a detalle, lo que puede ocasionar diferentes errores como falsos-positivos o positivos-falsos.

El proyecto plantea reducir este tipo de errores con un sistema de visión que es un ayudante automatizado para detectar si algún intruso cruza por el área o si algún objeto es removido del área, la forma de avisar al observador seria encerrando en un rectángulo verde el área donde el sistema de visión detecto actividad, con eso se facilita al observador la posición donde se encuentra un intruso o se extrajo un objeto para poner mayor atención en esa área y hacer su trabajo correspondiente.

Los sistemas de videovigilancia o de monitoreo a través de video han evolucionado notablemente en los últimos años. Desde los sistemas analógicos que tuvieron su máxima representante en los CCTV (Circuitos Cerrados de Televisión), hasta los sistemas digitales modernos con sistemas digitales de transmisión inalámbrica. Todo esto con el fin de mejorar la seguridad, ya que es la garantía que tienen las personas de estar libres de todo daño, amenaza, peligro o riesgo, es la necesidad de sentirse protegido, contra todo aquello que pueda perturbar o atentar contra su integridad física, moral, social y hasta económica.

Para mantener la seguridad de empresas, establecimientos, casa, etc. Se crearon sistemas de seguridad que constan de componentes de software, hardware, dispositivos periféricos y equipo de control que serán controlados por un operador de seguridad.

Los diseñadores tienen la tarea de determinar el software y el hardware que se adecue a las necesidades del cliente. Esto permitirá tener un sistema que garantice que el usuario no sólo tenga confianza en el sistema, sino que además se sienta cómodo.

Para mejorar esta seguridad entra en papel una disciplina científica que se llama visión artificial que incluye métodos para adquirir, procesar y analizar imágenes del mundo real con el fin de producir información que pueda ser tratada por una máquina. Una manera simple de comprender este sistema es basarse en nuestros propios sentidos.

Los humanos usamos nuestros ojos para comprender el mundo que nos rodea, y la visión artificial trata de producir ese mismo efecto en máquinas. Éstas podrán percibir y entender una imagen o secuencia de imágenes y actuar según convenga en una determinada situación. La principal finalidad de la visión artificial es dotar a la máquina de "ojos" para ver lo que ocurre en el mundo real, y así poder tomar decisiones para automatizar cualquier proceso.

La visión artificial antes no era algo muy factible ya que se necesita una capacidad de procesamiento demasiado grande, pero con el paso de los años ha llegado a ser indispensable en muchos sistemas, en este proyecto se pretende usar un sistema embebido con una buena capacidad de procesamiento que es una Raspberry PI.

La Raspberry PI es una placa computadora (SBC) de bajo costo, se podría decir que es un ordenador de tamaño reducido, del tamaño de una tarjeta de crédito. Está formada por una placa que soporta varios componentes necesarios en un ordenador común y es capaz de comportarse como tal.

Con los sistemas de seguridad y el creciente uso de sistemas de visión se han creado los sistemas de visión integrados que son sistemas de visión artificial que difieren relativamente de las cámaras inteligentes.

ISSN: 2523-6814

Una de las ventajas de un sistema de visión artificial que presenta esta tecnología con respecto a las cámaras inteligentes tradicionales es que, con un solo elemento de proceso, se pueden conectar varios cabezales de visión remotos, reduciendo el coste en aplicaciones de visión donde se requieran varias tomas de la misma pieza. Tanto las cámaras inteligentes, como los sistemas de visión artificial integrados, pueden incluir sensores CCD/CMOS de muy alta definición, tanto en monocromo como en color. La potencia de cálculo de estos sistemas de visión artificial es suficientemente para poder resolver la mayoría de las aplicaciones de visión.

El área de seguridad y de detección de intrusos viene siendo más indispensable cada día. Definiremos como intrusión a un conjunto de acciones que intentan comprometer la integridad o confidencialidad de un recurso por lo tanto la detección de intrusos es el proceso de monitorizar los eventos que ocurren en un sistema para analizarlos en busca de algunos problemas de seguridad.

La videovigilancia es un tipo de seguridad constituida por componentes electrónicos de la más diversa naturaleza, según sean las características de su aplicación, por lo tanto, un sistema de visión protege a un bien de los riesgos para los que ha sido diseñado el sistema. La protección por medio de un sistema de visión artificial es aplicable a distintos ámbitos, por ejemplo, existen alarmas contra ladrones o cámaras de vigilancia usadas por bancos o comercios.

Para lograr un sistema de videovigilancia eficiente hay que tener en cuenta varios factores de entre los cuales la iluminación es la parte más crítica dentro de un sistema de visión. Las cámaras capturan la luz reflejada de los objetos impidiendo así la apreciación del ambiente a vigilar. El propósito de la iluminación utilizada en las aplicaciones de visión es controlar la forma en que la cámara vera el objeto.

Desarrollo

Dentro de los distintos tipos de cámaras con las cuales se puede implementar un sistema de videovigilancia se encuentran las cámaras IP. Las siglas IP significan (Internet Protocol).

La función básica de la cámara IP es la transmisión de video a distancia sobre la red IP. De esta manera es posible observar lo que ocurre en un sitio determinado desde distintos lugares mediante la autorización del sistema.

La implementación de la librería OpenCV en las cámaras permite realizar los procesos necesarios para lograr la detección de intrusos gracias a que OpenCV es una librería que contiene muchas estructuras de datos complejos y funciones de alto nivel utilizadas para procesos de flujo óptico, reconocimiento de patrones.

Metodología del sistema de visión

El sistema de visión está programado en software libre, utilizando el sistema operativo Rasbian de 32bits para ejecutarlo en la Raspberry 3b, con los entornos de desarrollo, Python en su versión 2.7 de 32 bits y OpenCV en su versión 2.4.

Lo primero fue desarrollar el algoritmo, el cual consta de recibir el video de una cámara, desde un video almacenado o desde una cámara IP, después se toma un frame del video y se verificara si el fondo es válido.

Para correr el algoritmo en un sistema embebido el programa se debe optimizar lo más posible para funcionar en tiempo real por eso la imagen se debe cambiar a escala de grises, luego se le aplicara un algoritmo de detección de movimiento que es la parte primordial del sistema, una vez aplicado el algoritmo se detectan los contornos de la imagen utilizando un filtro de suavizado con un umbral, el siguiente paso consta de seleccionar el centro del contorno y asignar sus medidas a una variable, la detección de intrusos y la remoción de objeto será determinada por el área aplicando un límite de área para el valor de las variables, si esta sobrepasa el área permitida, se encerrara en un rectángulo de color negro, los pasos descritos se mostraran en diferentes pestañas para verificar su funcionamiento.

En la figura 1 se muestra el diagrama a bloques y a continuación la descripción detallada de cada parte del programa.

ISSN: 2523-6814

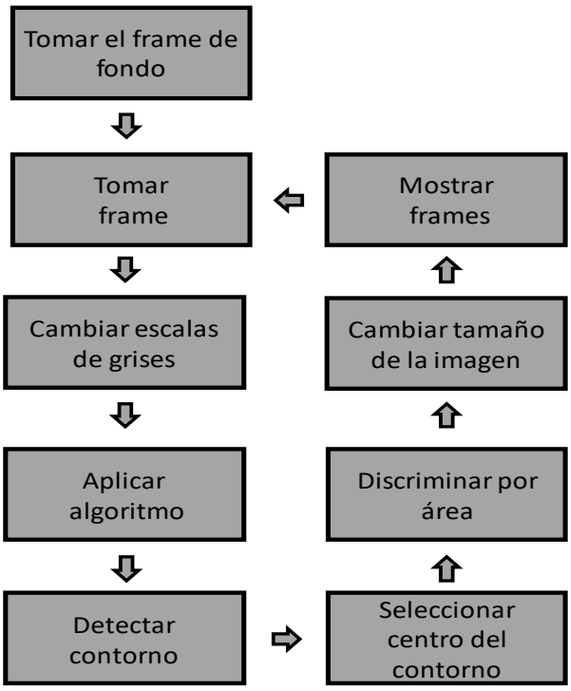

Figura 1 Diagrama de la metodología del algoritmo
Fuente: Elaboración propia

Tomar el frame. - En esta parte del programa lo que se hace es tomar el primer frame ya sea de una cámara conectada en tiempo real, de un archivo de video o de una cámara IP.

El primer frame obtenido es el que se establecerá como el fondo de la imagen. En esta parte del programa también se pueden modificar el tamaño de la imagen que se va a procesar.

Verificar fondo. - Al momento de tomar el primer frame con la cámara, la cámara tiene un cierto tiempo de respuesta el cual es en el que se inicializa el CCD. Si se tomara este primer frame el fondo estaría en color negro o gris dependiendo de la cámara dando datos inválidos lo que produciría un error en el algoritmo para solucionar este problema se aplica un delay después de inicializar la cámara y tomar el primer frame.

Cambiar a escala de grises. - Para aumentar la velocidad de procesamiento del programa se debe cambiar la imagen RGB a escala a grises, esto hace que de procesar 3 matrices solo se tenga que procesar 1 y así poder aumentar la velocidad de procesamiento del programa.

Aplicar algoritmo de detección de movimiento. - Esta es la parte esencial, ya que es la encargada de detectar si hay movimiento, esto se hace con diferentes algoritmos que serán descritos detalladamente después. Para detectar el movimiento se debe hacer una resta de la imagen del fondo con la imagen tomada después.

El resultado será la diferencia que se encontró en la imagen, a esa imagen se le aplicara un umbral para binarizar, el valor del umbral es un factor de calibración el cual dependerá del lugar donde se aplicara el sistema de visión.

Detectar contorno y seleccionar su centro. - A la imagen ya binarizada se le aplicara un algoritmo para detectar su contorno y así poder discriminar conforme a él. El algoritmo hará un barrido de la imagen de izquierda a derecha detectando conectividad entre los pixeles, al encontrar un blob se le asignará una variable para poder procesarlos después.

Discriminar por área. - La detección de movimiento y la remoción de objetos se detectará por área, aunque no es la mejor manera de discriminar se seleccionó esta opción ya que en el proyecto no se especifica qué tipo de objetos se va a remover o qué tipo de objetos se va a agregar al plano.

Una vez que se tienen las variables del contorno obtenidas del paso anterior se hará un filtro, que verificara si las variables del contorno sobrepasan cierta cantidad de área, esta área es otro factor de calibración puesto por el programador y depende del tamaño de los objetos que se deseen encontrar, si el contorno sobrepasa el filtro, el área seleccionada se encerrara en un rectángulo de color negro.

Mostrar frame. - Por último, se da la opción de mostrar en una ventana el frame capturado con la cámara y modificado por el programa, el frame binarizado con el umbral designado y el frame que detecta los contornos también se pueden modificar el tamaño de las ventanas dependiendo de la pantalla donde se mostrara.

ISSN: 2523-6814

Algoritmos de detección de movimiento

Un algoritmo de detección de movimiento se basa en la sustracción del fondo la cual consiste en tomar una imagen de la escena sin movimiento y restar los fotogramas sucesivos de un vídeo. A la imagen sin movimiento se le llama fondo o segundo plano. El fotograma que se analizaría sería del primer plano. Por lo tanto, tenemos un fondo al que vamos restando los diferentes fotogramas. No se requiere que el sujeto u objeto que se está intentando detectar deba tener algo que lo identifique como un sensor, un indicador o traje especial. Por el contrario, la substracción de fondo es muy sensible a los cambios de iluminación como las sombras o los cambios producidos por la luz natural. Una desventaja es que, si el sujeto u objeto tiene un color parecido al del fondo, no se detectara movimiento.

Dentro de la técnica de la substracción de fondo, existen dos modalidades que dependerá de cómo se obtiene el fondo o segundo plano, con imagen de referencia o con fotogramas anteriores.

Una de las modalidades es la substracción con imagen de referencia la cual consiste en tener una imagen de referencia donde no haya ningún objeto en movimiento. A partir de esta imagen se obtienen los elementos en movimiento restando cada fotograma con la imagen de referencia. Normalmente se toma el primer fotograma de una secuencia de vídeo. Es muy sensible a los cambios de luz. También es muy sensible a los movimientos de la cámara. Un movimiento muy pequeño puede hacer que se detecten falsos positivos en la escena. Por el contrario, este método funciona muy bien en entornos con iluminación.

La otra modalidad se llama substracción con fotogramas anteriores en esta modalidad, el fondo o segundo plano se obtiene de los fotogramas anteriores. La técnica consiste en tomar una imagen de referencia, dejar pasar un tiempo aplicando un retardo y empezar a comparar con los fotogramas que vamos obteniendo. Este retardo dependerá de factores como la velocidad de los objetos.

Una de las mayores desventajas es que si el objeto o la persona en movimiento se quedan quietos, no se detectaran si se calibra erróneamente.

Sin embargo, es un método bastante robusto a los cambios de iluminación y a los movimientos de cámara. Consigue estabilizarse pasado un tiempo.

En el proyecto para la detección de movimiento se usaron 3 algoritmos para pruebas los cuales son el algoritmo de resta con diseño y elaboración propia y 2 algoritmos proporcionados por OpenCV en su librería llamados MOG y MOG2. Los 3 algoritmos se sometieron a 4 diferentes pruebas para determinar cuál algoritmo es el más eficiente para proseguir con el proyecto, las pruebas son el ingresar un objeto al primer plano, quitar un objeto del primer plano, el camuflaje que checara si el fondo y lo que provoca el movimiento son del mismo color.

Por último, la prueba de luz que cambiara la intensidad de la luz para que el sistema se auto-compense después de cierto tiempo para seguir recibiendo datos válidos. Después de hacer las pruebas a los algoritmos se analizarán sus resultados y se elegirá uno para continuar con el proyecto.

Algoritmos de resta

Este algoritmo es de los más básicos para detección de movimiento, es mediante la sustracción de fondo lo que hace este algoritmo es tomar el primer frame como fondo y se le van restando los fotogramas siguientes del video.

En este algoritmo se deben modificar diferentes parámetros algunos de ellos son:

- El filtro de suavizado gaussiano, en el cual si modificamos los números 21 va cambiando la robustez del suavizado de la imagen entre más alto sea este número mayor será el suavizado.

Cv2.GaussianBlur(gris, (21, 21), 0)

- La aplicación del umbral es un factor de calibración ya que según las condiciones de iluminación este podrá variar, en esta función el umbral actual es de 25 pero puede variar desde 0 hasta 255. Mientras más grande sea el valor más robusto será la discriminación.

ISSN: 2523-6814

cv2.threshold(25,255, cv2.THRESH_BINARY)

En la figura 2 se aplica la primera prueba la cual consiste en remover un objeto del fondo aplicando el algoritmo de resta, se observa que, si detecto el objeto removido, pero además da varios errores de falsos positivos en este caso encontró 8 falsos-positivos y el objeto removido.

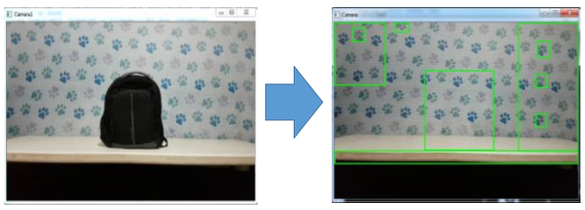

Figura 2 Prueba 1 con algoritmo de Resta
Fuente: Elaboración propia

En la figura 3 se aplica la segunda prueba la cual consiste en ingresar un objeto al fondo, aplicando el algoritmo de resta se observa que si detecto el objeto ingresado sin detectar errores.

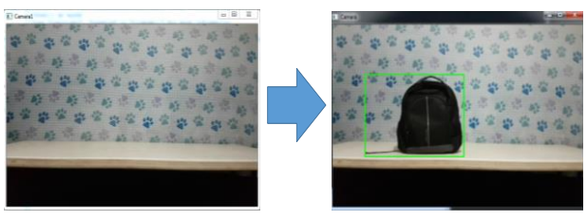

Figura 3 Prueba 2 con algoritmo de Resta
Fuente: Elaboración propia

En la figura 4 se aplica la tercera prueba la cual consiste en remover dos objetos del fondo uno del color del fondo y uno de color distinto, aplicando el algoritmo de resta se observa que detecto los 2 objetos, pero también tiene un error al detectar un falso-positivo.

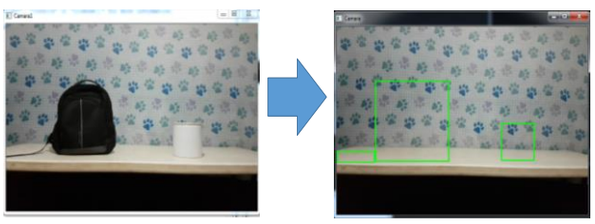

Figura 4 Prueba 3 con algoritmo de Resta
Fuente: Elaboración propia

En la figura 5 se aplica la cuarta prueba la cual consiste en un cambio de iluminación, aplicando el algoritmo de resta se observa que encuentra un error ya que al ser diferente al fondo lo detecta como movimiento al momento de encender de nuevo la iluminación seguía detectando el error ya que este algoritmo es muy sensible a la iluminación y al no actualizarse el fondo nunca se auto compensa.

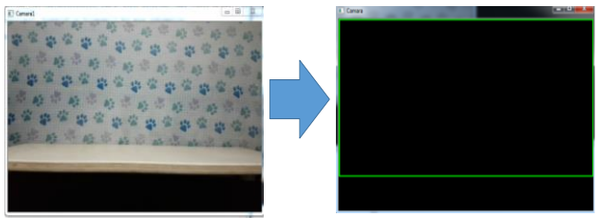

Figura 5 Prueba 4 con algoritmo de Resta
Fuente: Elaboración propia.

Algoritmos MOG

El BackgroundSubtractiorMOG es un algoritmo de la modalidad de substracción con fotogramas anteriores está basado en las mezclas Gaussiana/Algoritmos de segmentación de primer plano. Utiliza un método para modelar cada pixel de fondo mediante una mezcla de k distribuciones Gaussianas.

Los pesos de las mezclas representan las proporciones de tiempo que esos colores permanecen en la escena.

Los colores de fondo probables son los que permanecen más largos y más estáticos (InProc.ICPR, 2004). Este algoritmo es una función de OpenCV el cual se manda a llamar con el nombre:

cv2.bgsegm.createBackgroundSubtractorMOG(history=0,nmixtures=0,backgroundRatio=0, noiseSigma=0)

El cual se le deben calibrar los argumentos donde:

- History: Tamaño del histórico.
- Nmistures: Número de mezclas Gaussianas.
- Backgroundratio: Relación de fondo.
- Noisesigma: Potencia del ruido (desviación estándar de la luminosidad o de cada canal de color). 0 significa que obtiene un valor automático.

ISSN: 2523-6814

En la figura 6 se aplica la primera prueba la cual consiste en remover un objeto del fondo aplicando el algoritmo MOG, se observa que si detecto el objeto removido sin errores.

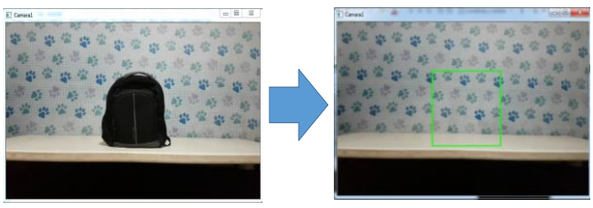

Figura 6 Prueba 1 con algoritmo de MOG
Fuente: Elaboración propia

En la figura 7 se aplica la segunda prueba la cual consiste en ingresar un objeto al fondo, aplicando el algoritmo de MOG se observa que si detecto el objeto ingresado sin detectar errores.

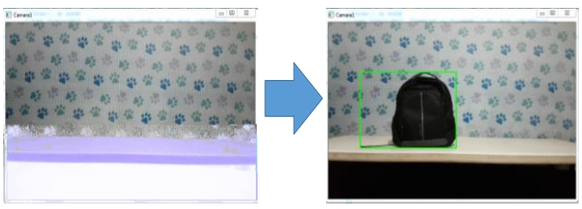

Figura 7 Prueba 2 con algoritmo de MOG
Fuente. Elaboración propia

En la figura 8 se aplica la tercera prueba la cual consiste en remover dos objetos del fondo uno del color del fondo y uno de color distinto, aplicando el algoritmo de MOG se observa que solo detecto el objeto de otro color, pero el de color blanco no lo detecto.

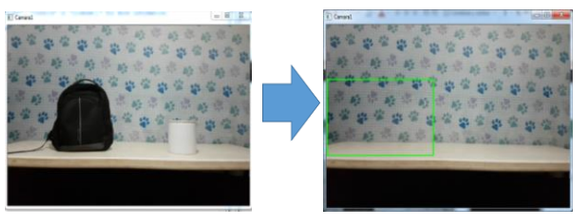

Figura 8 Prueba 3 con algoritmo de MOG
Fuente: Elaboración propia

En la figura 9 se aplica la cuarta prueba la cual consiste en un cambio de iluminación, aplicando el algoritmo de MOG se observa que encuentra un error ya que al ser diferente al fondo lo detecta como movimiento al momento de encender de nuevo la iluminación deja de detectar el error si la iluminación se va atenuando poco a poco el sistema se auto compensa con el paso del tiempo.

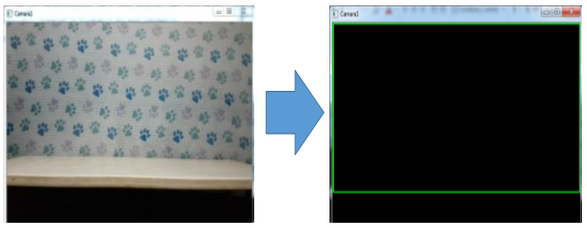

Figura 9 Prueba 4 con algoritmo de MOG
Fuente: Elaboración propia

Algoritmos MOG2

El BackgroundSubtractiorMOG2 es un algoritmo de la modalidad de substracción con fotogramas está basado en las mezclas Gaussianas/Antecedentes de segmentación. Una característica importante de este algoritmo es que selecciona el número apropiado de distribuciones Gaussianas para cada pixel. Proporciona una muy buena adaptación a escenas variables debido a cambios de iluminación.

Este algoritmo es una función de OpenCV el cual se manda a llamar con el nombre:

cv2.createBackgroundSubtractorMOG2(history =0,varThreshold=0, detectShadows=False)

El cual se le deben calibrar los argumentos donde:

- History: Tamaño del histórico.
- varThreshold: Umbral de la distancia de Mahalanobis al cuadrado entre el píxel y el modelo para decidir si un píxel está bien descrito por el fondo.
- DetectShadows: Con un valor verdadero (True) detecta las sombras. Esto reduce la velocidad así que si no se utiliza ponerlo a falso.

En la figura 10 se aplica la primera prueba la cual consiste en remover un objeto del fondo aplicando el algoritmo de MOG2, se observa que si detecto el objeto sin errores.

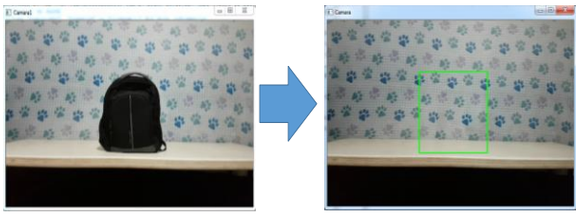

Figura 10 Prueba 1 con algoritmo de MOG2
Fuente: Elaboración propia

ISSN: 2523-6814

En la figura 11 se aplica la segunda prueba la cual consiste en ingresar un objeto al fondo, aplicando el algoritmo de MOG2 se observa que si detecto el objeto ingresado sin detectar errores.

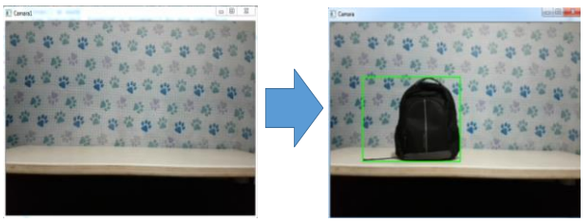

Figura 11 Prueba 2 con algoritmo de MOG2
Fuente: Elaboración propia

En la figura 12 se aplica la tercera prueba la cual consiste en remover dos objetos del fondo uno del color del fondo y uno de color distinto, aplicando el algoritmo de MOG2 se observa que detecto los dos objetos sin errores.

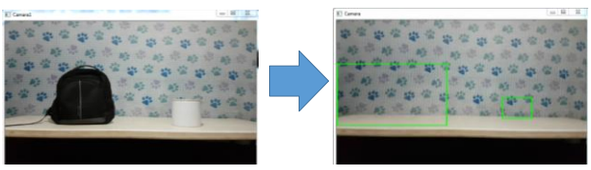

Figura 12 Prueba 3 con algoritmo de MOG2
Fuente: Elaboración propia

En la figura 13 se aplica la cuarta prueba la cual consiste en un cambio de iluminación, aplicando el algoritmo de MOG se observa que encuentra un error ya que al ser diferente al fondo lo detecta como movimiento al momento de encender de nuevo la iluminación deja de detectar el error si la iluminación se va atenuando poco a poco el sistema se auto compensa con el paso del tiempo.

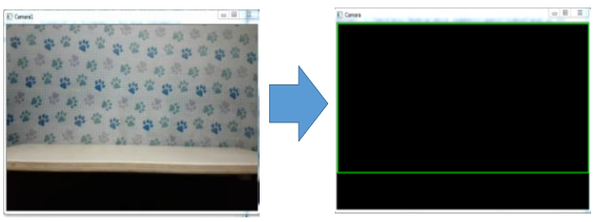

Figura 13 Prueba 4 con algoritmo de MOG2
Fuente: Elaboración propia

Obtención del video

Para obtener el rendimiento de procesamiento del video se hizo una prueba la cual consiste en ejecutar un video grabado a 60 FPS, obtenido de 3 formas diferentes las cuales son:

- Video ya grabado.
- Video por Función de OpenCV.
- Video vía streaming (IP).

Los frames obtenidos por el video ya grabado y por el video local en tiempo real fueron por funciones de Open CV y el video vía streaming fue hecho por función propia.

Resultados

El sistema de visión consta de 2 partes primordiales que sería la medición de velocidad en FPS del sistema y de las pruebas aplicadas para verificar su funcionamiento.

Para la primera parte se realizaron las 4 pruebas en los 3 algoritmos se calificaron como eficiente o deficiente dependiendo su resultado un resumen de las pruebas se muestra en la siguiente tabla.

Resultado de las pruebas aplicadas Prueba	Resta	MOG	MOG2
Quitar objeto del fondo	Deficiente	Eficiente	Eficiente
Agregar objeto al fondo	Eficiente	Eficiente	Eficiente
Camuflaje	Deficiente	Deficiente	Eficiente
Cambio de iluminación	Deficiente	Eficiente	Eficiente

Tabla 1 Resultados de las pruebas realizadas
Fuente: Elaboración propia

Aunque los datos mostrados en la tabla no son absolutos ya que cada algoritmo tiene sus datos de calibración y se puede dar el caso de que al cambiar los valores de los argumentos detecte correctamente o en caso contrario detecte errores.

Al hacer varias pruebas se decidió que el algoritmo de MOG2 es el indicado para el sistema, ya que fue el que mayor eficiencia dio a la hora de:

- Hacer cambios de iluminación.
- Tener pequeñas vibraciones en la posición de la cámara.
- Detectar objetos del mismo color que el del fondo.

ISSN: 2523-6814

La otra parte del proyecto fue a la hora de tomar el frame para el algoritmo ya que al capturar frames en video local en tiempo real, si el algoritmo tardaba mucho tiempo en procesar el frame, el siguiente frame se guardaba en el buffer de video y al pasar unos segundos se notaba el retraso que tenía el video bajando así los FPS. El algoritmo del IP es independiente del corrimiento del procesamiento por lo que no importa cuánto tarde el procesamiento siempre obtendrá el frame siguiente más rápido. En el grafico 1 se muestra el resultado de los FPS obtenidos después de transcurrir algún tiempo.

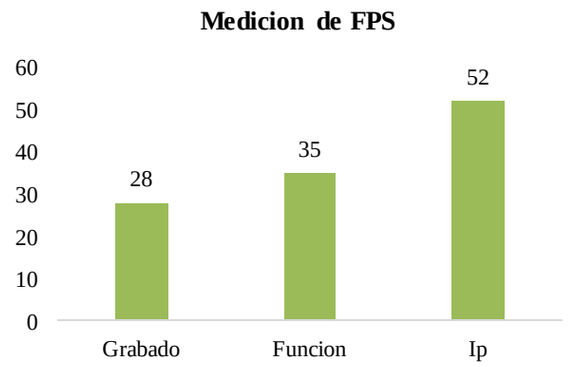

Gráfico 1 Medición de FPS en diferentes modos de captura de frames
Fuente: Elaboración propia

El sistema de visión en su forma final se usó con el algoritmo de MOG2 ya que en las pruebas fue el algoritmo que obtuvo mayor desempeño y para la obtención de frames se utilizó el de la cámara IP ya que en los resultados fue el que obtuvo mayor rapidez de procesamiento. Pará verificar su funcionamiento se sometió a una prueba en la biblioteca del Instituto Tecnológico de Chihuahua en la cual se extrajo un libro de los estantes.

En la figura 14 se observan cuatro ventanas las cuales tienen su nombre en la esquina superior derecha en la ventana de cámara se observa el video en tiempo real, en la ventana umbral se observa el umbral aplicado en este caso no se ve nada ya que es la primer imagen mostrada por la cámara y no ha detectado movimiento, la otra ventana es la de contornos la cual muestra los contornos encontrados y por último la imagen modificada que encierra en un rectángulo la detección de un intruso o la remoción de un objeto en este caso no muestra nada ya que es el primer frame tomado.

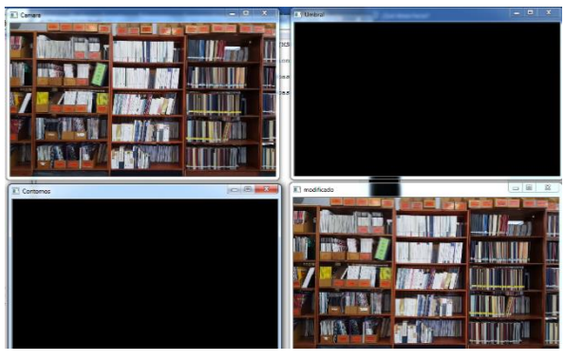

Figura 14 Interfaz final del sistema sin detectar remoción de objetos o intrusos
Fuente: Elaboración propia

En la figura 15 se muestra las mismas 4 ventanas pero con un libro ya removido, el libro removido es el de color verde que tiene como nombre Electro, en la ventana de cámara se observa que ya no está el libro pero no nos dice si el libro ha sido removido ya que esta ventana solo muestra lo que está pasando en el video, en la ventana de contornos ya se distingue que de color blanco se marca el contorno del libro removido y en la ventana de umbral se rellenan los contornos para poder encontrar su centro, en la ventana de modificado seria el resultado final la cual muestra la imagen ya modificada y encerrando en un rectángulo de color verde (se modificó a color verde para apreciar mejor el resultado) que un objeto ha sido removido.

En este ejemplo se muestra en colores para apreciar mejor que el libro ha sido removido, pero en el resultado final es en escala a grises.

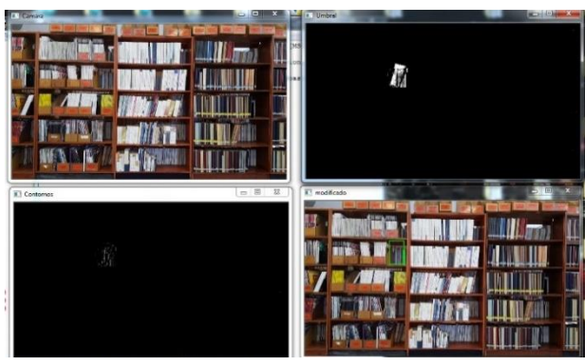

Figura 15 Interfaz final del sistema detectando objeto removido
Fuente: Elaboración propia

ISSN: 2523-6814

Discusión

El sistema se ejecuta en tiempo real en el sistema embebido, pero con la desventaja de hacerlo en escala a grises, ya que al procesar el video en RGB se vuelve demasiado lento esto no es una desventaja del todo ya que al detectar el movimiento se encierra en el recuadro de color negro dándonos las coordenadas del movimiento obteniendo lo que sería una ROI inteligente (región de interés).

Esto se puede aplicar como un pre-procesamiento para aplicar un algoritmo más complejo o en su caso una red neuronal para reconocimiento de objetos, una red neuronal ocupa mucha capacidad de procesamiento pero con la ROI inteligente solo se puede aplicar la red neuronal en donde se detectó movimiento así reduciendo el consumo de esta y cumpliendo con una de las reglas principales de un sistema de visión que dice que el objeto capturado por la cámara debe ocupar el mayor espacio disponible, hay que recordar que el video se convierte en escala a grises pero esto no influiría ya que el reconocimiento de objetos con red neuronal se puede hacer en escala a grises.

El algoritmo fue probado en un sistema con más recursos pudiendo procesar el algoritmo en RGB en tiempo real siendo este más eficiente que en una RaspberryPi, dándonos a entender que en caso de querer procesar muchas cámaras es más eficiente y barato hacerlo en un pc que en un sistema embebido.

Otra ventaja es que al estar usando una cámara IP para la captura del video se hace uso de la tecnología del internet de las cosas ya que deja de ser solo una cámara IP y se convierte en una cámara IP inteligente la cual tiene la capacidad de detectar intrusos u objetos removidos.

Agradecimientos

Los autores agradecen al Tecnológico Nacional de México por las facilidades otorgadas para realizar este proyecto.

Conclusiones

En base a los resultados de las pruebas aplicadas al sistema de visión se observa que si hay cambios de iluminación muy drásticos el sistema tarda en auto-calibrarse, si en el lugar donde está sujeta la cámara hay pequeñas vibraciones o movimiento el sistema no las detecta, si el color del fondo es parecido al objeto que se desea encontrar no siempre lo detecta siendo esta la parte más difícil de calibrar, se recomienda que el sistema se utilice en zonas con variables controladas ya que así no tendrá falsos-positivos. Se aconseja usar un algoritmo diferente para la captura de frames a los proporcionados por OpenCV ya que al ser algoritmos muy complejos se quedan frames guardados en el buffer de video para no perder información y haciendo el sistema más lento. Para poder ejecutar el algoritmo en la RaspberryPI se tienen que convertir de RGB a escala a grises para que el procesamiento no sea tan largo.

Referencias

Cantabria. (2013). Estudio de sistemas de seguridad basado en la detección de intrusión física y tecnológica. Master. Universidad de Cantabria.

Contaval. (2016). ¿Qué es la visión artificial y para qué sirve? Recuperado el 29 de agosto del 2017, de Contaval: http://www.contaval.es/que-es-la-vision-artificial-y-para-que-sirve

Cuevas, Eric; Zaidivar, Daniel Pérez, Marco; (2016). Procesamiento digital de imágenes con Matlab y Simulink. Colombia: Alfaomega.

Escuela. (2001). Estructura y funcionalidad de un sistema de seguridad. Recuperado el 29 de agosto del 2017, de Escuela: http://catarina.udlap.mx/u_dl_a/tales/documentos/lis/lezama_l_a/capitulo1.pdf

Foscam, (2017). Cámaras IP, Recuperado el 29 de agosto del 2017, de Foscam: https://www.foscam.es/descarga/manual.pdf

González, D. (2003). Sistemas de Detección de Intrusiones. 1st ed. Free Software Foundation,
G. Bradski and A. Kaehler (2015), Aprendiendo OpenCV. O'Reilly Media.

ISSN: 2523-6814

Imágenes. (2007). Chacón M. Mario I. Procesamiento digital de imágenes, Editorial Trillas

Infaimon visión artificial. (2017). Visión Artificial. Recuperado el 29 de agosto del 2017 de Infaimon visión artificial: http://www.infaimon.com/es/sistemas-vision-integrados-industria

InProc.ICPR (2004). Improved Adaptive Gaussian Mixture Model for Background Subtraction. Zoran Zivkovic Intelligent and Autonomous Systems Group University of Amsterdam, The Netherlands

Lidia Contreras. (2013). Raspberry PI. Recuperado el 29 de agosto del 2017, de histinf: http://histinf.blogs.upv.es/author/

Salas Arriarán, Sergio. (2015). Todo sobre sistemas embebidos. Perú: Editorial UPC.

Samuel Greengard. (2015). El Internet de las Cosas. Cambridge, Massachusetts: The Mit Press.

University of Amsterdam (2009). Efficient adaptive density estimation per image pixel for the task of background subtraction, Zoran Zivkovic* Ferdinand van der Heijden, Faculty of Science, University of Amsterdam, Kruislaan 4903, 1098SJ Amsterdam, The Netherlands University of Twente, The Netherlands.

Venemedia. (2011). Definición de seguridad. Recuperado el 29 de agosto del 2017, de conceptodefinicion.de: http://conceptodefinicion.de/seguridad/

Visión Artificial. (2017). Visión Artificial Aplicada. Recuperado el 29 de agosto del 2017 de CIP ETI: http://www.etitudela.com/celula/downloads/visionartificial.pdf.

ISSN: 2523-6814