

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ

Colegio de Ciencias e Ingenierías

**Detección Temprana y Seguimiento de Focos de Calor
Usando Técnicas de Procesamiento de Imágenes
Proyectos de Investigación y Desarrollo**

Cintya Tatiana Medina Ortiz

Ingeniería Eléctrica y Electrónica

Trabajo de titulación presentado como requisito
para la obtención del título de
Ingeniero Eléctrico y Electrónico

Quito, 21 de diciembre de 2016

UNIVERSIDAD SAN FRANCISCO DE QUITO USFQ
COLEGIO DE CIENCIAS E INGENIERÍAS

**HOJA DE CALIFICACIÓN
DE TRABAJO DE TITULACIÓN**

**Detección Temprana y Seguimiento de Focos de Calor Usando Técnicas de
Procesamiento de Imágenes**

Cintya Tatiana Medina Ortiz

Calificación:

Nombre del profesor, Título académico

Diego Benítez, Ph.D.

Firma del profesor

Quito, 21 de diciembre de 2016

Derechos de Autor

Por medio del presente documento certifico que he leído todas las Políticas y Manuales de la Universidad San Francisco de Quito USFQ, incluyendo la Política de Propiedad Intelectual USFQ, y estoy de acuerdo con su contenido, por lo que los derechos de propiedad intelectual del presente trabajo quedan sujetos a lo dispuesto en esas Políticas.

Asimismo, autorizo a la USFQ para que realice la digitalización y publicación de este trabajo en el repositorio virtual, de conformidad a lo dispuesto en el Art. 144 de la Ley Orgánica de Educación Superior.

Firma del estudiante:

Nombres y apellidos:

Cintya Tatiana Medina Ortiz

Código:

00106774

Cédula de Identidad:

1003583877

Lugar y fecha:

Quito, diciembre de 2016

RESUMEN

Los incendios forestales constituyen uno de los problemas más graves que amenazan los bosques alrededor del mundo. En los últimos años, el índice de incendios forestales a nivel mundial ha aumentado drásticamente. Es por esto que, en varios países europeos, ya existen sistemas que permiten la vigilancia automática de los bosques haciendo uso de varias herramientas tecnológicas como cámaras y sensores. En el presente trabajo se desarrolla un sistema con el fin de automatizar la supervisión de los bosques naturales ecuatorianos tomando en cuenta los requerimientos y equipo limitado. Se ha logrado crear un sistema para la detección temprana y el seguimiento de focos de calor haciendo uso de una cámara térmica y procesamiento digital de imágenes. Esta herramienta cuenta con un diseño de interfaz gráfica intuitivo, de manera que el usuario final pueda manejar la aplicación de forma sencilla. El sistema garantiza resultados con una baja probabilidad de falsas detecciones. Si bien el proyecto cumple con las expectativas del usuario final, existen varias proyecciones de trabajo futuro para desarrollar en una nueva versión. Algunas de estas proyecciones son el montaje de la cámara en un cuadricóptero para el procesamiento del video en tiempo real y la adquisición de una cámara térmica con un rango de operación de temperatura más amplio.

Palabras clave: focos de calor, cámara térmica, Matlab, seguimiento, detección, sustracción de fondo, filtro de Kalman, análisis Blob.

ABSTRACT

Wildfires are one of the most serious problems that threaten forests around the world. In the last years, the rate of forest fires worldwide has increased dramatically. For this reason, in several European countries, there is software that allow the automatic monitoring of forests using technological tools such as cameras and sensors. In this project, a system has been developed in order to improve the supervision of natural forest taking into account the initial requirements and a limited technological devices available. This software tool was created for the early detection and monitoring of forest fire using a thermal camera and digital image processing techniques. This tool has an intuitive graphical interface design, so that the user can manipulate the application in a simple way. The system guarantees results with a low probability of false detections. Although the project satisfies end user expectations, there are several further improvements that could be implemented as future work, included the use of a camera mounted in a drone in order to capture and process video in real time, and the use of a thermal camera with a better temperature range of operation.

Key words: Forest fire, thermal camera, Matlab, tracking, foreground detection, Kalman Filter, Blob Analysis.

TABLA DE CONTENIDO

Introducción.....	8
Justificación e importancia del proyecto.....	8
Antecedentes.....	10
Teletron Euroricerche	10
Ciclope.....	10
IForestFire.....	10
Objetivos del proyecto.....	11
Desarrollo del Tema	13
Planificación	13
Discriminación de objetos en primer plano.	14
Análisis Blob.....	14
Filtro de Kalman.	15
Diseño.....	15
Implementación.....	17
Creación de objetos.	18
Inicializar trayectorias.....	18
Leer un cuadro del video.....	19
Detectar objetos.....	19
Predecir nueva ubicación de trayectorias existentes.	20
Asignar detecciones a trayectorias.....	20
Actualizar trayectorias asignadas.....	20
Actualizar trayectorias no asignadas.	21
Eliminar trayectorias perdidas.	21
Crear nuevas trayectorias.....	21
Visualizar resultados.	21
Conclusiones.....	22
Sugerencias y Observaciones.....	24
Referencias Bibliográficas.....	26
Anexo A: Algoritmo desarrollado en Matlab	28

ÍNDICE DE FIGURAS

Figura 1. Pantalla de inicio	16
Figura 2. Pantalla principal.....	17
Figura 3. Diagrama de flujo del algoritmo propuesto.....	18

INTRODUCCIÓN

Justificación e importancia del proyecto

Los incendios forestales constituyen uno de los problemas más graves que amenazan los bosques tanto en nuestro país, así como alrededor del mundo. Hasta el mes de marzo del 2016, la Unidad Nacional para la Gestión de Riesgos y Desastres colombiana, por ejemplo, informó que los incendios forestales, en lo que va del año, afectaron alrededor de 60000 hectáreas de cultivos, páramos y bosques de niebla (NGRD, 2016). Las pérdidas económicas que el país tuvo que afrontar con el fin de atender estas emergencias fueron de alrededor de 11 mil millones de pesos colombianos, lo que equivale a 3 millones 630 mil dólares americanos (NGRD, 2016). Más de estos casos se han evidenciado alrededor del mundo.

En Israel, el pasado 22 de noviembre del 2016, se detectaron cerca de 200 incendios forestales en distintos puntos del norte del país (CNN, 2016). Los operativos para controlar el fuego duraron una semana, alrededor de 60 mil personas fueron evacuadas y se solicitó ayuda internacional para combatir el avance de las llamas utilizando medios aéreos (CNN, 2016). Por otra parte, en la provincia de San Luis ubicada en el centro norte de Argentina, el 28 de agosto del 2016 se registró un incendio forestal que afectó cerca de 8500 hectáreas de flora y fauna (ANSL, 2016). Este incidente provocó dificultades para potabilizar el agua en los lugares afectados además de afectar la actividad económica de los habitantes.

Por otro lado, en un ámbito local, de acuerdo con los datos proporcionados por el Cuerpo de Bomberos de Distrito Metropolitano de Quito, en el año 2016 los incendios forestales consumieron 121 hectáreas de bosque (CBDMQ, 2016). De acuerdo al informe anual del Cuerpo de Bomberos, las áreas más afectadas fueron Nayón, La Pulida e Ingañaro. A diferencia de este último año, en el 2015 se quemaron 2668 hectáreas (CBDMQ, 2016). El

Cuerpo de Bomberos señala que durante este año las áreas más afectadas fueron los bosques metropolitanos y reservas naturales, especialmente en la región de Puenbo.

Estos ejemplos de hechos ocurridos tanto en Ecuador como alrededor del mundo, demuestran que la inmediata detección de un foco de calor es la clave para evitar que se llegue a declarar un incendio. Es por esto que, en varios países, ya se han tomado medidas al respecto. En la actualidad, los bosques y reservas forestales son monitoreados mediante una red de cámaras que detectan la presencia temprana de un incendio. De esta manera, es posible dar aviso en tiempo real a la central de control para que el equipo encargado tome las medidas respectivas con el fin de extinguir el fuego localizado. Con esto, se asegura la preservación de los bosques naturales evitando, al mismo tiempo, pérdidas económicas.

Ecuador es un país que cuenta con una alta biodiversidad en flora y fauna. Lamentablemente en este país, las ventajas que la tecnología puede ofrecer en el campo de incendios forestales no han sido bien aprovechadas. Es por este motivo que, en los últimos años, varias instituciones académicas han centrado sus investigaciones en este campo con el fin de hacer uso de sensores y cámaras para automatizar el proceso de vigilancia de incendios forestales. Un claro ejemplo es el proyecto de tesis presentado en mayo del presente año por la Universidad Técnica de Ambato titulado como “Sistema Electrónico de Alerta de Incendios Forestales Mediante Procesamiento Digital de Imágenes para la Empresa Municipal del Cuerpo de Bomberos de Ambato”. Se conoce que este sistema utiliza una cámara digital para la obtención del video (Valle, 2016); sin embargo, no se ha encontrado información sobre su funcionamiento en la estación del Cuerpo de Bomberos de Ambato.

Antecedentes

A nivel internacional, existen varios sistemas comerciales similares al propuesto en este proyecto. Estos sistemas utilizan distintos métodos y técnicas para la detección de focos de calor que se mencionarán a continuación:

Teletron Euroricerche.

Esta empresa está conformada por 86 estaciones periféricas, 27 centros de operaciones y tiene a su vigilancia alrededor de 2.3 millones de hectáreas en varios países europeos como Italia, Portugal, Grecia y España (Teletron Euroricerche, 2016). Los sistemas de Teletron Euroricerche trabajan con cámaras panorámicas de visión multispectral de 360 grados. Estas cámaras operan tanto en la banda visible como en la banda infrarroja y consideran distintos parámetros ambientales como agua, aire, calidad del suelo, ruido, contaminación electromagnética y condiciones climáticas.

Ciclope.

Este sistema de origen portugués fue diseñado por el grupo de investigación INOV. Cubre alrededor de 1.3 millones de hectáreas de forma remota, lo que permite reducir el costo por hectárea (CICLOPE, 2010). Ciclope combina los beneficios de cámaras digitales e infrarrojas para su funcionamiento, así como sensores de radar láser y mediciones de condiciones meteorológicas.

IForestFire.

Este sistema tiene su base en Croacia. Opera de manera inteligente de tal manera que durante el día hace uso de cámaras de análisis de imágenes en luz visible, y durante la noche

utiliza cámaras térmicas (iForestFire, 2004). La herramienta detecta signos visibles de humo y fuego. En caso de sospecha de incendio el operador a cargo en la Central de Control define si se trata de una falsa alarma o si se realizan los procedimientos respectivos para controlar el incendio.

De acuerdo a una evaluación previa, se evidenció que no existe en Ecuador una herramienta tecnológica que sirva como ayuda en la labor del control de incendios forestales, aprovechando las facilidades que brinda una cámara térmica en este campo. Es así como nace la idea de este proyecto. Fue iniciativa de Carlos Andrade, profesor de la Universidad San Francisco de Quito, la creación de un sistema que permita aprovechar el potencial de una cámara térmica, la cual fue adquirida por el Departamento de Mecánica de la misma universidad, en el campo de incendios forestales. Esta herramienta permitirá la detección y el seguimiento de focos de calor en tiempo real. En un principio, el sistema utilizará un archivo de video guardado previamente tomado por la cámara térmica. Sin embargo, más adelante se planea montar la cámara en un cuadracóptero; así, el sistema recibirá el video que necesita ser procesado vía streaming con la finalidad de detectar el posible incendio en tiempo real.

Objetivos del proyecto

Este trabajo tiene como finalidad el diseño e implementación de un sistema para la detección temprana y seguimiento de focos de calor mediante el uso de una cámara térmica y técnicas de procesamiento digital de imágenes. Para lograr esta meta, se establecieron los siguientes objetivos específicos:

- Recopilar información relacionada con sistemas comerciales existentes para la detección automática de incendios forestales mediante el uso de una cámara térmica en Ecuador y alrededor del mundo.

- Desarrollar un algoritmo que permita la detección y el seguimiento de focos de calor a partir de videos obtenidos con una cámara térmica.
- Diseñar una aplicación con una interfaz gráfica sencilla para la visualización de los resultados en tiempo real.

Si bien los conceptos de procesamiento de imágenes que sirven como base teórica de este proyecto no son complicados, su desarrollo requiere de un tiempo de trabajo considerable para poder garantizar un correcto funcionamiento y satisfacción del usuario. Las siguientes secciones muestran en detalle los pasos que se realizaron en cuanto a la planificación, desarrollo e implementación del proyecto así como las dificultades encontradas y las recomendaciones propuestas.

DESARROLLO DEL TEMA

Planificación

La etapa de planificación tuvo como finalidad definir los requerimientos del sistema, tanto del software necesario para su desarrollo, así como el contenido a implementarse. Durante este período de planificación se mantuvieron varias reuniones con Carlos Andrade con el propósito de definir la aplicación del sistema a desarrollarse, los dispositivos y el tipo de contenido que se manejará, la información necesaria a visualizarse y las fechas tentativas para la entrega de avances del proyecto.

Con relación al contenido que el sistema debe presentar y su diseño, se requieren los siguientes componentes:

- Diseño simple
- Mínima cantidad de botones
- Una única ventana
- Visualización del video de muestra con los resultados en la misma ventana
- En los resultados, se debe señalar el lugar de inicio del foco de calor y seguir su trayectoria durante el video.

En cuanto a los requerimientos informáticos y técnicos con los que el sistema de detección debe contar, se tienen los siguientes:

- Uso de la cámara térmica marca FLIR modelo VUE
- Desarrollo del algoritmo en Matlab
- Uso de la herramienta Computer Vision System Toolbox de Matlab para el procesamiento de la imagen.
- El video para analizar estará guardado previamente en formato MP4.

Una vez delimitados todos los requerimientos de diseño, informáticos y técnicos, se realizó una investigación bibliográfica con el fin de seleccionar las técnicas de procesamiento de imágenes más adecuadas para este proyecto. Se determinó que la Discriminación de Objetos en Primer Plano, el Análisis Blob y el Filtro de Kalman son las técnicas más opcionadas que permitirán obtener los mejores resultados. Cabe mencionar que Matlab cuenta con funciones ya desarrolladas en la herramienta Computer Vision System Toolbox para aplicar las tres técnicas anteriormente mencionadas. Estas funciones integradas en el software se utilizarán con el fin de aprovechar todas sus ventajas, las cuales se mencionarán a continuación.

Discriminación de objetos en primer plano.

Esta es una técnica con la cual es posible detectar los cambios que se presentan en una secuencia de imágenes (MathWorks, 2016). Como su nombre lo dice, el objetivo final es separar el fondo de la imagen del primer plano con el fin de analizar los cambios de píxeles en cada una de las escenas. Esta función de Matlab devuelve una imagen resultante en forma binaria donde 1 representa los píxeles de primer plano y 0 representa los píxeles de fondo (Bouwmans, 2014).

Análisis Blob.

El Análisis Blob es una técnica que permite la detección de regiones en una imagen que poseen cualidades comunes; estas regiones pueden ser brillo color, forma, etc. (Elleithy, 2015). De acuerdo a Elleithy (2016), esta técnica de procesamiento de imágenes consta de tres fases. En la fase de extracción se obtiene una región correspondiente al objeto de análisis. En la fase de refinamiento se mejora la imagen eliminando el ruido existente. Finalmente, en

la fase de análisis se extrae la información necesaria como por ejemplo el área, el centro o la orientación del objeto de análisis. El Análisis Blob permite agrupar los píxeles del primer plano, tomados de la imagen resultante de la discriminación de objetos de primer plano, que poseen características comunes, también llamados blobs. Generalmente, estos grupos de píxeles representan un objeto en movimiento dentro del video (MatWorks, 2016).

Filtro de Kalman.

Es un algoritmo recursivo, diseñado para aplicaciones que operan en tiempo real. Fue desarrollado por Rudolf E. Kalman en 1960 con el propósito de identificar el estado oculto, o no medible, de un objeto en un sistema dinámico lineal cuando éste se encuentra sometido a ruido blanco auditivo (Chui, 2009). En Matlab, el filtro de Kalman permite estimar el estado de un sistema a partir de ciertos datos conocidos; como por ejemplo las mediciones de entrada actuales, el estado calculado previamente, y la matriz de incertidumbre (MathWorks, 2014). Este filtro hace uso del término *costo* el cual se define como la probabilidad logarítmica negativa de una detección correspondiente a una trayectoria (Chui, 2009).

Diseño

La etapa de diseño se caracterizó casi en su totalidad por el diseño de la interfaz gráfica del usuario, la cual fue desarrollada en Matlab GUIDE. Esta interfaz de Matlab es un entorno de programación visual que permite realizar y ejecutar programas que necesiten el ingreso continuo de datos (Barragán, 2009). Además, GUIDE de Matlab posee las características básicas de todos los programas visuales como Visual Basic o C++. Para este proyecto se trató de crear un diseño intuitivo y simple para que el usuario final no encuentre dificultad en manejar el sistema.

En cuanto al diseño general, la interfaz gráfica cuenta con una pantalla de bienvenida y otra pantalla principal. En la pantalla de bienvenida se visualiza una breve presentación del proyecto. Esta contiene una imagen de fondo, información del programa y un botón *Continuar* el cual permite acceder a la pantalla principal. La figura 1 se puede apreciar el diseño mencionado.



Figura 1. Pantalla de inicio

Finalmente, la pantalla principal cuenta con el área donde se presentará el video y sus resultados, y tres botones para el manejo completo del sistema: *Seleccionar video*, *Inicio* y *Stop* como se muestra en la figura 2. En un principio, únicamente el botón *Seleccionar video* se encuentra habilitado. Al presionar este botón se abre una ventana que permitirá buscar el video a analizar. Una vez escogido el video, se habilitan los dos botones restantes, *Inicio* y

Stop. Al presionar el botón *Inicio*, se da lugar al procesamiento del video y este mismo botón se deshabilita hasta que el video se haya reproducido completamente; los resultados se presentan en la parte izquierda de la pantalla. El botón *Stop* permite detener el video en cualquier momento, en caso de ser necesario. El proceso de habilitar y deshabilitar los botones permite un mejor manejo de excepciones en caso de que presente un mal manejo del sistema por parte del usuario final.



Figura 2. Pantalla principal

Implementación

La implementación del sistema se realizó de acuerdo al diagrama de flujo del algoritmo propuesto, mostrado en la figura 3. Cada punto en el diagrama de flujo corresponde a una función desarrollada en Matlab. El algoritmo desarrollado está disponible en el Anexo A de este documento.

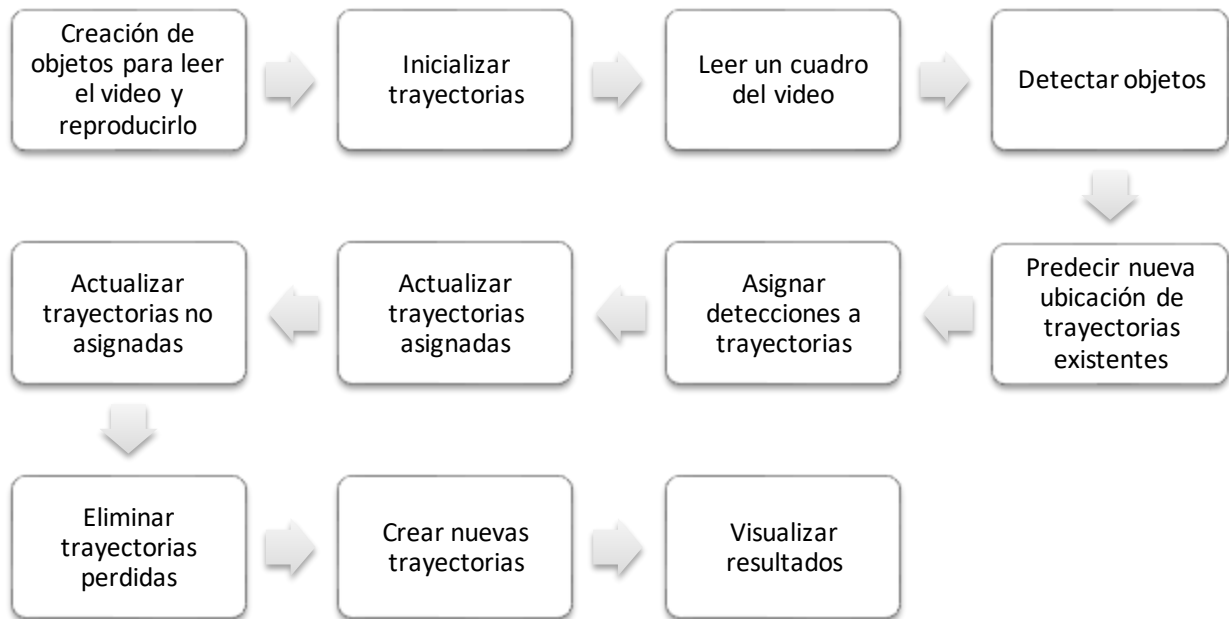


Figura 3. Diagrama de flujo del algoritmo propuesto

Creación de objetos.

En este punto se crean cuatro objetos que se manejarán a lo largo del sistema. El primer objeto tiene la finalidad de inicializar el video que se encuentra en la ruta escogida por el usuario final, este será el video que se requiere analizar. El segundo objeto se utilizará para la detección de piezas en movimiento haciendo uso de la técnica de discriminación de objetos en primer plano. El tercer objeto, por otra parte, manejará el Análisis Blob el cual detectará las regiones en la imagen que poseen cualidades comunes. Finalmente, el cuarto objeto será manipulado para presentar los resultados en la pantalla principal.

Inicializar trayectorias.

En esta función se crea un arreglo vacío para manejar las trayectorias encontradas. Este arreglo está compuesto por seis características.

- Id: permitirá identificar la trayectoria

- Cuadro: el cual se podrá visualizar en el video de resultados delimitando el foco de calor detectado.
- FiltroKalman: objeto que se utiliza para el seguimiento del foco de calor basándose en el movimiento constante.
- Constf: constante que determina el número de cuadros desde que la primera trayectoria fue detectada.
- TotalVisible: mínimo número de cuadros en los cuales la trayectoria detectada aparece como visible.
- TotalInvisible: máximo número de cuadros en los cuales la trayectoria detectada no se puede visualizar.

Leer un cuadro del video.

Esta es una función sencilla que tiene como finalidad leer el siguiente cuadro del video que se está analizando. Esto se lo realiza haciendo uso de comando *step* de Matlab.

Detectar objetos.

En este punto se aplica la primera técnica de procesamiento de imágenes, la discriminación de objetos en primer plano. Como se explicó anteriormente, la función devuelve una imagen resultante en forma binaria donde 1 representa los pixeles de primer plano y 0 representa los pixeles de fondo. A esta imagen resultante, se le aplica el Análisis Blob. Como primer paso, se aplican operaciones morfológicas para eliminar el ruido y llenar agujeros en el grupo de blobs. La función devuelve los centroides de cada blob y los cuadros delimitadores para cada objeto en movimiento detectado.

Predecir nueva ubicación de trayectorias existentes.

En esta función se aplica el filtro de Kalman. La principal finalidad de esta función es la de predecir el centroide de cada trayectoria en el cuadro actual que se encuentra analizando, para luego actualizar el cuadro delimitador correspondiente a ese centroide.

Asignar detecciones a trayectorias.

Esta acción se la realiza minimizando el costo. Primeramente, se calcula el costo de asignar una detección a determinada trayectoria usando el dato de la distancia proporcionado por el filtro de Kalman. Este dato corresponde a la distancia Euclídeana entre el centroide que se predice en la trayectoria y el centroide de la detección. El valor de costo de no asignar una detección a una trayectoria depende directamente del rango de valores proporcionados por el dato de distancia. Esta constante se ajustó experimentalmente a un valor de 20. Al determinar un valor por debajo de 20, se incrementa la posibilidad de una división de la trayectoria. Por lo contrario, al determinar un valor superior a 20, aumenta la posibilidad de crear una sola trayectoria para distintos objetos en movimiento.

Actualizar trayectorias asignadas.

Como su nombre lo dice, esta función actualiza cada trayectoria asignada a la detección del objeto que corresponde. Para hacer esto, se utiliza el componente *correct* proporcionado por el filtro de Kalman con el fin de corregir la estimación de la ubicación. Luego, se asigna un nuevo cuadro que delimita el área del objeto. Finalmente, se aumenta en 1 el valor de la constante TotalVisible, así como de la constante constf la cual determina el número de cuadros desde que la primera trayectoria fue detectada.

Actualizar trayectorias no asignadas.

Esta es una función sencilla cuya principal finalidad es marcar las trayectorias no asignadas como invisibles. Cuando esto sucede, se incrementa en 1 el valor de la constante TotalInvisible.

Eliminar trayectorias perdidas.

Se encarga de eliminar las trayectorias que han sido invisibles durante varios cuadros consecutivos. La función realiza esta eliminación de trayectorias en función del id de la trayectoria en cuestión, y tomando en cuenta el valor de la contante TotalInvisible mencionada anteriormente.

Crear nuevas trayectorias.

Con esta función se crean nuevas trayectorias para las detecciones a las cuales todavía no se les ha asignado una. Para hacer esto, se asigna la nueva trayectoria al arreglo de trayectorias inicializado anteriormente y se incrementa en 1 el valor de id de la trayectoria.

Visualizar resultados.

El primer paso de esta función es insertar en el frame del video el cuadro delimitador de área para el objeto en movimiento detectado. Finalmente, se presenta el video con los resultados de la detección haciendo uso del objeto para video declarado al inicio del algoritmo.

CONCLUSIONES

Mediante el desarrollo de este proyecto fue posible lograr la creación de un sistema de detección automática temprana y seguimiento de focos de calor en Ecuador. La herramienta desarrollada cumplió con las expectativas y requerimientos iniciales establecidos los cuales fueron el uso de una cámara térmica y la detección en base a tres herramientas de Procesamiento Digital de Imágenes: Discriminación de Objetos en Primer Plano, Análisis Blob y Filtro de Kalman.

En cuanto a la investigación bibliográfica realizada previo al desarrollo del proyecto, se concluye que los sistemas más robustos en el campo de la detección de incendios forestales que hacen uso de una cámara térmica son desarrollados en el continente europeo. Los sistemas más efectivos tienen un funcionamiento dual. Es así que, durante el día se utilizan cámaras digitales de espectro visible para la detección de focos de calor; sin embargo, durante la noche se utilizan cámaras térmicas. Es claro que se hace uso de técnicas de procesamiento de imágenes más sofisticadas para que este funcionamiento dual produzca resultados confiables.

Con respecto al algoritmo desarrollado, fue posible optimizar los recursos necesarios en cuanto a la cantidad de líneas de código para obtener buenos resultados mediante el uso de la herramienta Computer Vision System Toolbox de Matlab. Haciendo un correcto uso de las funciones ya desarrolladas disponibles en esta herramienta, fue posible aprovechar todas las características que brindan la Discriminación de Objetos en Primer Plano, el Análisis Blob y el Filtro de Kalman para la detección y el seguimiento de objetos. El uso del video tomado por una cámara térmica mejora los resultados generados a partir de la sustracción de fondo y de primer plano. Al ser esta la técnica base para la detección de objetos en movimiento, la

probabilidad baja de lecturas falsas es evidente. Es así como se concluye que el uso de una cámara térmica representa una ventaja en sistemas no comerciales de detección de focos de calor.

Finalmente, fue posible recopilar toda la información que se deseaba presentar en forma de resultados en una sola ventana gracias a la interfaz gráfica GUIDE de Matlab. La sencillez de la interfaz gráfica es vital ya que permite al usuario final, el cual probablemente no tiene una formación técnica en el ámbito académico, el correcto uso del sistema. A pesar de que la interfaz se diseñó para que el manejo de la aplicación fluya de una forma intuitiva, no se dejó de lado el manejo de excepciones con el fin de garantizar una excelente experiencia por parte del usuario a la vez que se garantiza la validez de los resultados.

SUGERENCIAS Y OBSERVACIONES

A pesar de que se cumplieron todos los objetivos propuestos para este proyecto, existen varios cambios que se deben tomar en cuenta con el fin de obtener mejores resultados y optimizar el funcionamiento del sistema en trabajos futuros. Las sugerencias que a continuación se mencionan, se dan en base a las pruebas realizadas y a los avances presentados como parte de la planificación del proyecto.

Durante las pruebas se utilizaron dos computadoras con distintas características. La primera, es una computadora portátil marca HP modelo Split 2; cuenta con un procesador Intel Core i3, velocidad de 1,40 GHz y memoria RAM de 4Gb. La segunda computadora utilizada es una PC; cuenta con un procesador Intel Core i3, velocidad 3,07 GHz y memoria RAM de 4Gb. La principal diferencia entre los dos dispositivos utilizados es la velocidad del procesador.

Debido a estas observaciones, con respecto al funcionamiento del sistema, se recomienda utilizar equipos de alta gama para el procesamiento del video. A lo largo de las pruebas realizadas, fue evidente que los mejores resultados se obtuvieron con la computadora de escritorio ya que se redujo el tiempo que toma el procesamiento del video en casi la mitad, en comparación con el tiempo que le tomó a la computadora portátil realizar el mismo proceso. Por lo tanto, lo más recomendable es utilizar una computadora con una velocidad del procesador mayor a 3 GHz y, al menos, 4Gb en RAM.

En cuanto al video que se necesita analizar, se recomienda que éste sea grabado en escala de grises tomando en cuenta el color negro como el identificador para el punto más caliente del cuadro. La escala de grises en una imagen térmica permite ver objetivos de bajo contraste incluso en escenas con amplios rangos de temperatura, un incendio forestal por

ejemplo (FLIR, 2015). Esta escala muestra detalles más sutiles, una imagen más nítida y de textura más fina. El uso de la escala de grises redujo la cantidad de objetos detectados erróneamente.

Por otra parte, varios estudios muestran que un foco de calor, en sus inicios, puede variar su temperatura entre 100 °C y 200 °C; una vez sobrepasada esta temperatura se declara como incendio forestal (FIC, 2011). Debido a esto, la última sugerencia, pero la más importante, es el uso de una cámara térmica con mejores características. La cámara utilizada en este proyecto es una cámara térmica marca FLIR modelo Vue, tiene un rango de operación desde -20 °C hasta 50 °C (FLIR, 2016). Es por esto que, cuando se enfoca un objeto que se encuentra a una temperatura mayor a 50 °C, los sensores de la cámara se saturan. Además, otra de las desventajas de este dispositivo es la autocalibración. Es así que, se obtiene una imagen térmica en función del punto más caliente que se encuentra dentro del cuadro visualizado por la cámara. Una vez que un objeto más caliente entra en el cuadro, la imagen se autocalibra provocando falsas detecciones de focos de calor.

Con el fin de minimizar en la medida de lo posible la probabilidad de falsas detecciones, se recomienda adquirir una cámara térmica marca FLIR modelo Tau 2 640. Este dispositivo opera dentro de un rango de temperaturas desde -30 °C hasta 550 °C y permite desactivar la autocalibración (FLIR, 2015). Debido al rango de operación más amplio en este modelo, no se producirán errores por saturación de los sensores de la cámara. Además no existirán falsas detecciones a causa de la autocalibración del dispositivo.

Si bien las sugerencias presentadas anteriormente indican que el sistema desarrollado para la detección temprana y seguimiento de focos de calor todavía puede mejorar, por ahora es una herramienta tecnológica de gran ayuda estratégica y única en el país.

REFERENCIAS BIBLIOGRÁFICAS

- ANSL. (27 de 08 de 2016). *Trabajo Interministerial*. Recuperado el 01 de 12 de 2016, de Agencia de Noticias San Luis: <http://agenciasanluis.com/notas/2016/08/27/el-comite-de-crisis-informo-los-relevamientos-y-acciones-realizadas-para-combatir-los-incendios/>
- Barragán, D. (2009). *Manual de Interfaz Gráfica de Usuario en Matlab*. Loja: MATPIC.
- Bouwman, T. (2014). *Background Modeling and Foreground Detection for Video Surveillance*. Paris: Université de La Rochelle.
- CBDMQ. (14 de 09 de 2016). *Incendios Forestales*. Recuperado el 01 de 12 de 2016, de Cuerpo de Bomberos del Distrito Metropolitano de Quito: <https://www.bomberosquito.gob.ec/index.php/forestales>
- Chui, C. (2009). *Kalman Filtering: With Real-Time Applications*. Texas: Springer.
- CICLOPE. (12 de 11 de 2010). *Casos de Sucesso*. Recuperado el 01 de 12 de 2016, de INOV: <http://www.inov.pt/index/casos-de-sucesso/166-ciclope.html>
- CNN. (25 de 11 de 2016). *Miles de evacuados por incendios en Israel*. Recuperado el 1 de 12 de 2016, de CNN en español: <http://cnnespanol.cnn.com/2016/11/25/miles-de-evacuados-por-incendios-en-israel/>
- Elleithy, K. (2015). *New Trends in Networking, Computing, E-learning, Systems Sciences, and Engineering*. New York: Springer.
- FIC. (2011). *Incendios forestales*. Recuperado el 02 de 12 de 2016, de Fundación para la investigación del clima: <http://www.ficlina.org/incendios-forestales/>
- FLIR. (2015). *Flexible Scene Enhancement (FSX™) Provides Clear Images in Critical Environments*. Recuperado el 02 de 12 de 2016, de FLIR - Firefighting: <http://www.flir.com/fire/display/?id=66949>
- FLIR. (2015). *Tau 2 Specifications*. Recuperado el 02 de 12 de 2016, de OEM Cameras & Components: <http://www.flir.com/cores/display/?id=54717>
- FLIR. (2016). *Specifications*. Recuperado el 02 de 12 de 2016, de FLIR Vue: <http://www.flir.com/suas/content/?id=68335>
- iForestFire. (15 de 08 de 2004). *What is iForestFire*. Recuperado el 01 de 12 de 2016, de Intelligent ForestFire Monitoring System: <http://www.lama.hr/en/solutions/integral-solutions/iforestfire/>

MathWorks. (15 de 01 de 2014). *Design and use Kalman filters in MATLAB and Simulink*.

Recuperado el 02 de 12 de 2016, de MathWorks.com:

<https://www.mathworks.com/discovery/kalman-filter.html>

MathWorks. (15 de 01 de 2016). *Vision.ForegroundDetector System object*. Recuperado el 02 de 12 de 2016, de MathWorks.com:

<https://www.mathworks.com/help/vision/ref/vision.foregrounddetector-class.html>

MatWorks. (15 de 01 de 2016). *Blob Analysis*. Recuperado el 02 de 12 de 2016, de

MatWorks.com: <https://www.mathworks.com/help/vision/ref/blobanalysis.html>

NGRD. (13 de 03 de 2016). *Manos criminales principales causantes de los incendios forestales*. Recuperado el 2 de 12 de 2016, de Unidad Nacional para la Gestión del Riesgo de Desastres - Colombia:

<http://portal.gestiondelriesgo.gov.co/Paginas/Noticias/2016/Manos-criminales-principales-causantes-de-los-incendios-forestales.aspx>

Teletron Euroricerche. (29 de 06 de 2016). *Storie di Progetti*. Recuperado el 01 de 12 de 2016, de Teletron Euroricerche:

<http://www.sardegnaprogrammazione.it/monitoraggio/it/progetti/teletron-euroricerche-srl>

Valle, F. (2016). *Sistema Electrónico de Alerta de Incendios Forestales Mediante Procesamiento Digital de Imágenes para la Empresa Municipal del Cuerpo de Bomberos de Ambato*. Ambato: Universidad Técnica de Ambato.

ANEXO A: ALGORITMO DESARROLLADO EN MATLAB

```

function focos_de_calor(nombre_video)

obj = objetos();
tracks = arreglo_tracks();
num_track = 1;
cont = ~isDone(obj.leer);

while cont
    frame = leer_frame();
    [centros, cuadros, fondo] = detectar(frame);
    nueva_ubicacion();
    [tracks_asignados, tracks_no_asignados, detec_no_asignadas] =
asignar_tracks();
    actualiza_tracks_asignados();
    actualiza_tracks_no_asignados();
    borrar_tracks();
    nuevos_tracks();
    resultados();
    cont = ~isDone(obj.leer) && isOpen(obj.reproducir);
end

%%
function obj = objetos()

    obj.leer = vision.VideoFileReader(nombre_video);
    obj.reproducir = vision.VideoPlayer('Position', [20, 100, 650,
550]);
    obj.reproducir_fondo = vision.VideoPlayer('Position', [675, 100,
650, 550]);
    obj.detector = vision.ForegroundDetector('NumGaussians', 3,
'NumTrainingFrames', 40, 'MinimumBackgroundRatio', 0.7);
    obj.blob = vision.BlobAnalysis('BoundingBoxOutputPort', true,
'AreaOutputPort', true, 'CentroidOutputPort', true, 'MinimumBlobArea',
1000); %1000 para personas

    end

%%
function tracks = arreglo_tracks()

    tracks = struct('id', {}, 'cuadro', {}, 'kalmanFilter', {}, 'age',
{}, 'totalVisibleCount', {}, 'consecutiveInvisibleCount', {});

    end

%%
function frame = leer_frame()

    frame = obj.leer.step();

    end

%%
function [centros, cuadros, fondo] = detectar(frame)

```

```

fondo = obj.detector.step(frame);
fondo = imopen(fondo, strel('rectangle', [3,3]));
fondo = imclose(fondo, strel('rectangle', [15, 15]));
fondo = imfill(fondo, 'holes');
[~, centros, cuadros] = obj.blob.step(fondo);

end

%%
function nueva_ubicacion()

    for x = 1:length(tracks)
        cuadro = tracks(x).cuadro;
        nuevo_centros = predict(tracks(x).kalmanFilter);
        nuevo_centros = int32(nuevo_centros) - cuadro(3:4) / 2;
        tracks(x).cuadro = [nuevo_centros, cuadro(3:4)];
    end
end

%%
function [tracks_asignados, tracks_no_asignados, detec_no_asignadas] =
asignar_tracks()

    num_tracks = length(tracks);
    num_detec = size(centros, 1);
    costo = zeros(num_tracks, num_detec);

    for x = 1:num_tracks
        costo(x, :) = distance(tracks(x).kalmanFilter, centros);
    end

    costo_no_asignados = 20;
    [tracks_asignados, tracks_no_asignados, detec_no_asignadas] =
assignDetectionsToTracks(costo, costo_no_asignados);

end

%%
function actualiza_tracks_asignados()

    num_tracks_asignados = size(tracks_asignados, 1);

    for x = 1:num_tracks_asignados

        trackx = tracks_asignados(x, 1);
        detectx = tracks_asignados(x, 2);
        centro = centros(detectx, :);
        cuadro = cuadros(detectx, :);
        correct(tracks(trackx).kalmanFilter, centro);
        tracks(trackx).cuadro = cuadro;
        tracks(trackx).age = tracks(trackx).age + 1;
        tracks(trackx).totalVisibleCount =
tracks(trackx).totalVisibleCount + 1;
        tracks(trackx).consecutiveInvisibleCount = 0;
    end

end

```

```

%%
function actualiza_tracks_no_asignados()

    for x = 1:length(tracks_no_asignados)
        ind = tracks_no_asignados(x);
        tracks(ind).age = tracks(ind).age + 1;
        tracks(ind).consecutiveInvisibleCount =
tracks(ind).consecutiveInvisibleCount + 1;
        end

    end

%%
function borrar_tracks()

    if isempty(tracks)
        return;
    end

    const_perdido = 10;
    const_tiempo = 8;
    tiempo = [tracks(:).age];
    num_visibles = [tracks(:).totalVisibleCount];
    visibles = num_visibles ./ tiempo;
    perdidos = (tiempo < const_tiempo & visibles < 0.6) |
[tracks(:).consecutiveInvisibleCount] >= const_perdido;
    tracks = tracks(~perdidos);

    end

%%
function nuevos_tracks()

    centros = centros(detec_no_asignadas, :);
    cuadros = cuadros(detec_no_asignadas, :);

    for x = 1:size(centros, 1)

        centro = centros(x, :);
        cuadro = cuadros(x, :);

        kalman = configureKalmanFilter('ConstantVelocity', centro,
[200, 50], [100, 25], 100);
        nuevo_track = struct('id', num_track, 'cuadro', cuadro,
'kalmanFilter', kalman, 'age', 1, 'totalVisibleCount', 1,
'consecutiveInvisibleCount', 0);

        tracks(end + 1) = nuevo_track;
        num_track = num_track + 1;
    end

    end

%%
function resultados()

    frame = im2uint8(frame);

```

```

fondo = uint8(repmat(fondo, [1, 1, 3])) .* 255;
minimo_visible = 8;

if ~isempty(tracks)

    num_tracks_reales = [tracks(:).totalVisibleCount] >
minimo_visible;
    tracks_reales = tracks(num_tracks_reales);

    if ~isempty(tracks_reales)
        cuadros = cat(1, tracks_reales.cuadro);
        %numero = int32([tracks_reales(:).id]);
        %labels = cellstr(int2str(numero));
        labels = {''};

        frame = insertObjectAnnotation(frame, 'rectangle', cuadros,
labels);
        fondo = insertObjectAnnotation(fondo, 'rectangle', cuadros,
labels);
    end
end

obj.reproducir_fondo.step(fondo);
obj.reproducir.step(frame);
end

end

```