

UNIVERSIDAD PERUANA DE CIENCIAS APLICADAS

FACULTAD DE INGENIERÍA

CARRERA DE INGENIERÍA ELECTRÓNICA

**DETECTOR DE EVENTOS REMOTOS BASADO EN TÉCNICAS
DE PROCESAMIENTO DIGITAL DE VÍDEO**

PROYECTO PROFESIONAL PRESENTADO POR:

ALAN YOSHIO AZAMA MAKISHI

TOMAS FERNANDO HUAMAN HUANCA

PARA OPTAR EL TÍTULO DE INGENIERO ELECTRÓNICO

Lima, Octubre del 2007

*“Si quieres ser sabio, aprende a interrogar razonablemente,
a escuchar con atención, a responder serenamente
y a callar cuando no tengas nada que decir.”*

(Johann Kaspar Lavater)

*A mis padres,
por motivarme a seguir adelante
y brindarme su apoyo incondicional.*

*A mi familia,
que siempre me ha aplaudido
en mis éxitos y aconsejado en mis derrotas.*

Yoshio

A dios,

*A mis padres, por su comprensión y
apoyo durante todo este tiempo.*

A mis hermanos, por sus sinceros consejos.

*A todos mis buenos amigos, gracias
por las gratas experiencias compartidas y
por permitirme ser parte de su mundo.*

Tomás

Agradecimientos

Quisiéramos dar un especial agradecimiento a las personas que colaboraron con la realización de este trabajo por habernos brindado su valioso tiempo y apoyo:

A nuestro asesor, Guillermo Kemper Vásquez, por su apoyo y orientación en el desarrollo de este proyecto profesional.

A nuestros profesores por haber compartido sus conocimientos y experiencias.

A nuestros amigos y compañeros, José Torres y Anderson Sinche, porque colaboraron muy de cerca en el desarrollo de nuestro proyecto; y también a nuestros grandes amigos, Gino, Walter, Julio, Roberto y Carlos, gracias por su amistad.

A todos mil gracias.

RESUMEN

El tema del presente proyecto profesional trata sobre las diferentes técnicas de procesamiento digital de imágenes aplicadas a un sistema de vigilancia, con cámaras IP conectadas en red. Estas técnicas de procesamiento hacen al sistema capaz de detectar automáticamente eventos remotos, tales como la detección de movimiento o el cambio de iluminación de una escena.

El desarrollo del proyecto está dividido en cinco capítulos. El primer capítulo presenta un panorama general del proyecto. El segundo capítulo explica los fundamentos teóricos utilizados. El tercer capítulo muestra el desarrollo de los algoritmos utilizados para la detección automática de eventos. El cuarto capítulo expone las características del diseño y la implementación del sistema. Finalmente, el último capítulo es para realizar un análisis de los resultados obtenidos.

Al finalizar el trabajo, luego de múltiples pruebas del sistema, se llegó a la conclusión de que la detección de eventos remotos utilizando técnicas de procesamiento digital de vídeo es un método de fácil implementación y superior a muchos sistemas actuales.

OBJETIVO GENERAL

- Desarrollar un sistema visual de seguridad de alta confiabilidad, capaz de detectar y verificar automáticamente eventos remotos, utilizando un conjunto de cámaras IP conectadas en red.

OBJETIVOS ESPECÍFICOS

- Diseñar la interfaz de conexión y calibración de las cámaras IP para poder visualizar las imágenes capturadas y determinar los objetos a ser detectados dentro de la escena.
- Implementar algoritmos de detección de movimiento y de cambio de iluminación, utilizando técnicas de procesamiento de imágenes que permitan alcanzar una baja probabilidad de error. Así como también, determinar la dirección y la velocidad de un objeto en movimiento.
- Implementar el registro de eventos con la capacidad de migrar a un archivo de Excel.
- Implementar las alarmas sonoras con un motor de voz en español, transformando texto a voz (TextToSpeech) de forma automática.
- Implementar los algoritmos de grabación y reproducción de imágenes capturadas y también de archivos de video en formato AVI.

ÍNDICE

INTRODUCCIÓN.....	I
CAPÍTULO 1. Panorama general del proyecto.....	1
CAPÍTULO 2. Fundamentos teóricos.....	6
2.1 Teoría de imágenes.....	6
2.1.1 Definición de Píxel.....	7
2.1.2 Modelos de color de un Píxel.....	8
2.1.3 Resolución de imágenes.....	12
2.1.4 Procesamiento digital de imágenes.....	13
2.2 Interpolación de imágenes.....	15
2.2.1 Tipos de interpolación.....	15
2.2.2 Interpolación bilineal.....	17
2.3 Cámaras IP.....	18
2.3.1 Esquema de una cámara IP.....	20

2.3.2 Aplicaciones comunes.....	21
2.4 Interconexión del sistema.....	22
2.4.1 Red LAN.....	22
2.4.2 Protocolo IP.....	23
2.5 Sistema de vigilancia con cámaras IP.....	24
2.5.1 Programa de gestión de vídeo.....	25
2.5.2 Ancho de banda consumido por una cámara IP.....	27
2.5.3 Capacidad requerida por el sistema.....	28
CAPÍTULO 3. Algoritmos desarrollados para el proyecto.....	30
3.1 Algoritmo de detección de movimiento.....	30
3.2 Algoritmo de detección de cambio de iluminación.....	34
3.3 Algoritmo de almacenamiento de vídeo.....	38
3.4 Algoritmo de almacenamiento de imágenes.....	42
3.5 Algoritmo de reproducción de imágenes.....	45
CAPÍTULO 4. Diseño e implementación del sistema.....	49
4.1 Esquema del sistema.....	50
4.2 Interfaz visual principal.....	52
4.2.1 Opciones de la ventana principal.....	53
4.3 Interfaz de calibración de objetos.....	58
4.4 Interfaz de vídeo y análisis de imágenes.....	61
4.4.1 Proceso de umbralización de la imagen.....	63

4.4.1 Cálculo de posición, velocidad y dirección del objeto.....	66
4.5 Interfaz de registro de eventos.....	70
4.6 Interfaz de zoom digital.....	72
4.7 Implementación de alarmas sonoras.....	74
CAPÍTULO 5. Resultados obtenidos.....	75
5.1 Conexión entre dispositivos de red.....	75
5.2 Velocidad de procesamiento del sistema.....	77
5.3 Probabilidad de detección de eventos remotos.....	79
5.4 Consideraciones importantes.....	82
5.5 Especificaciones finales del sistema.....	83
CONCLUSIONES.....	86
RECOMENDACIONES PARA TRABAJOS FUTUROS.....	88
TABLA DE ABREVIATURAS.....	90
BIBLIOGRAFIA.....	91
ANEXOS.....	94

ÍNDICE DE FIGURAS

Figura 1.1 Ejemplo de dos software de gestión convencionales para cámaras IP.....	3
Figura 2.1 Ampliación de un sector de una imagen.....	7
Figura 2.2 Composición de una imagen en RGB.....	8
Figura 2.3 Mezcla aditiva de colores del RGB y el modelo sustractivo del CMYK.....	9
Figura 2.4 De izquierda a derecha, imagen con sus componentes Y, I y Q.....	11
Figura 2.5 De izquierda a derecha, aumento de la resolución de una imagen.....	12
Figura 2.6 Ejemplos de aplicaciones del procesamiento digital de imágenes.....	14
Figura 2.7 Comparación entre los métodos de interpolación.....	16
Figura 2.8 Valor interpolado entre sus 4 vecinos más cercanos.....	17
Figura 2.9 Diferentes modelos de cámaras IP.....	18
Figura 2.10 Diagrama de bloques de una cámara IP.....	20
Figura 2.11 Ejemplo de monitorización remota.....	21
Figura 2.12 Esquema de la conexión del sistema.....	22

Figura 2.13 Estructura de la trama enviada mediante el protocolo IP.....	23
Figura 2.14 Empleo del sistema de vigilancia IP.....	25
Figura 2.15 Formas de visualización y administración de las cámaras IP.....	27
Figura 3.1 Diagrama de bloques para la detección de movimiento.....	31
Figura 3.2 Diagrama de flujo para la detección de movimiento.....	32
Figura 3.3 Diagrama de bloques para la detección de cambio de iluminación.....	35
Figura 3.4 Diagrama de flujo para la detección de cambio de iluminación.....	36
Figura 3.5 Diagrama de bloques del algoritmo para almacenar vídeo.....	39
Figura 3.6 Diagrama de flujo del algoritmo para almacenar vídeo.....	40
Figura 3.7 Diagrama de bloques del algoritmo para almacenar imágenes.....	43
Figura 3.8 Diagrama de flujo del algoritmo para almacenar imágenes.....	44
Figura 3.9 Diagrama de bloques del algoritmo para reproducir imágenes.....	46
Figura 3.10 Diagrama de flujo del algoritmo para reproducir imágenes.....	47
Figura 4.1 Esquema del sistema desarrollado.....	51
Figura 4.2 Ventana principal del programa desarrollado.....	52
Figura 4.3 Ventana de configuración de cámara IP.....	53
Figura 4.4 Monitorización de una cámara IP conectada.....	54
Figura 4.5 Descripción de las cámaras IP conectadas.....	56
Figura 4.6 Registro de eventos detectados.....	57

Figura 4.7 Modo de visualización secuencial.....	57
Figura 4.8 Ventana de calibración de objetos.....	58
Figura 4.9 Detección de movimiento en la cámara 1.....	59
Figura 4.10 Variación en la luminancia de los píxeles.....	60
Figura 4.11 Detección de cambio de iluminación en la cámara 1.....	61
Figura 4.12 Ventana inicial del analizador de vídeos.....	62
Figura 4.13 Proceso de binarización de la imagen.....	64
Figura 4.14 Variación de luminancia antes y después de la umbralización.....	64
Figura 4.15 Imágenes antes y después de la umbralización y la imagen binarizada.....	65
Figura 4.16 Variación el centro de masa (punto rojo).....	67
Figura 4.17 Espacio que cubre la cámara en función de la distancia de referencia.....	68
Figura 4.18 Ángulo calculado utilizando la posición actual y anterior del centro de masa.....	69
Figura 4.19 Ventana final del analizador de vídeos.....	70
Figura 4.20 Ventana de registro de los eventos detectados.....	71
Figura 4.21 Archivo de registro generado en Excel.....	72
Figura 4.22 Imagen amplificada utilizando interpolación bilineal.....	73
Figura 4.23 Componente SpVoice instalado en Builder C++.....	74
Figura 5.1 Red local de computadoras.....	76

ÍNDICE DE TABLAS

Tabla 2.1 Detalle de la imagen capturada.....	27
Tabla 2.2 Espacio requerido del sistema para una cámara.....	28
Tabla 4.1 Dirección del objeto según el cuadrante del ángulo obtenido.....	69
Tabla 5.1 Respuesta del sistema respecto a calidad visual.....	79
Tabla 5.2 Respuesta del sistema respecto a detección de movimiento.....	81
Tabla 5.3 Características finales obtenidas del programa desarrollado.....	83

INTRODUCCIÓN

En los últimos años, la importancia de los sistemas de vigilancia a través de cámaras se ha ido incrementando tanto para los negocios como para las aplicaciones domésticas, debido al desarrollo de nuevas tecnologías como el Internet, el cual permite a las personas conectarse simultáneamente a distintos lugares y así poder realizar el monitoreo de sus bienes.

Actualmente, existen sistemas de vigilancia de circuito cerrado (CCTV) y también sistemas de aplicación remota, los cuales en su mayoría están diseñados sólo para realizar monitoreo de cámaras y grabación de vídeo en forma manual. Sin embargo, algunos de ellos permiten la detección automática de eventos a través de sensores analógicos, lo cual hace que estos sistemas posean elevados costos.

Por otra parte, algunos de estos sistemas de vigilancia han comenzado a aplicar el procesamiento de imágenes en reemplazo de los sensores analógicos para así minimizar sus costos, pero ésta aún es una técnica muy poco difundida. Es por ello que surge la motivación de realizar un sistema de detección automática de eventos remotos aplicando técnicas de procesamiento digital de vídeo y de esta forma conseguir un sistema de baja probabilidad de error en la detección automática de eventos, sin la necesidad de adquirir cámaras muy sofisticadas que requieren de una gran inversión.

El desarrollo de este trabajo consta de cinco capítulos, de los cuales el primero presenta el panorama general de proyecto, en el cual se plantea el problema a resolver y la situación

actual de soluciones similares existentes en el mercado. El segundo capítulo está dedicado a mostrar en detalle los fundamentos teóricos utilizados en el desarrollo del proyecto, tales como la teoría de imágenes y su procesamiento, las características de funcionamiento de una cámara IP, la interconexión del sistema o algunas características básicas del sistema en sí, entre otros temas. El tercer capítulo explica el funcionamiento de los algoritmos de procesamiento de imágenes desarrollados durante el proyecto, como por ejemplo, los algoritmos de detección de eventos y también los algoritmos de almacenamiento y reproducción de imágenes. En el cuarto capítulo se detallan las características del diseño y de la implementación del sistema, así como las interfaces diseñadas, el registro de eventos, el zoom digital y las alarmas sonoras implementadas con un motor de voz en español. Finalmente, en el último capítulo se realiza un análisis de las pruebas y resultados obtenidos.

La metodología utilizada ha consistido en el desarrollo del programa cliente a base de prueba y error, por lo cual muchas veces se recurrió a diversas fuentes consultadas a través de Internet y también a fuentes académicas, todas estas especializadas.

Como alcance de este trabajo, se puede mencionar que durante el desarrollo del mismo se descubrió que cada vez más las empresas de seguridad están aplicando las técnicas de procesamiento digital de vídeo, pero aún no al nivel de personalización de nuestro sistema. Por otra parte, se contó con la limitación de realizar las pruebas de detección automática de eventos sólo en una red privada, ya que no se poseía los requisitos necesarios para trabajar en una red pública, el cual es un medio con mucho más tráfico de datos.

CAPÍTULO 1

PANORAMA GENERAL DEL PROYECTO

La tecnología de vídeo IP ofrece muchos beneficios ante las aplicaciones de vigilancia por circuito cerrado de TV (CCTV). A raíz de estos beneficios, se ha producido un importante crecimiento en la fabricación y desarrollo de cámaras IP, y se han desarrollado diversos programas de gestión que permiten la manipulación de estas cámaras. Por otra parte, aunque el rendimiento de las cámaras analógicas es comparable con el de las cámaras IP, éstas últimas ofrecen una perspectiva diferente.

Los avances recientes han hecho posible conectar de forma directa las cámaras a una red, y además han permitido a las personas tener cámaras en una cierta localización y observar el vídeo en tiempo real desde otro lugar a través de la red interna o por Internet. Mediante la configuración de la cámara, el acceso puede restringirse de tal manera que sólo las personas autorizadas puedan acceder a la cámara para observar las imágenes. Si se piensa instalar un sistema de video vigilancia, se debe tomar en cuenta que si se tiene la infraestructura necesaria para incorporar las cámaras IP a la red es recomendable utilizar las cámaras IP antes que

emplear el sistema tradicional de vigilancia de Circuito Cerrado de TV (CCTV) por las siguientes razones:

- Las cámaras IP permiten la observación y grabación de eventos mediante un acceso remoto, a diferencia de los sistemas CCTV que el monitoreo debe hacerse en el lugar de control. Además, las imágenes pueden visualizarse por Internet desde un navegador Web y en soluciones de seguridad más complejas se pueden visualizar a través de un software dedicado, mientras que para una cámara analógica, es necesario la adición de un servidor de vídeo para que las imágenes pueden ser visualizadas por Internet.
- La instalación de las cámaras IP es más flexible que la de un circuito CCTV porque la infraestructura se basa en una red local existente o nueva, y además el costo de instalación de las cámaras IP es inferior a la instalación de las cámaras analógicas.
- En el caso de una ampliación de la red de vigilancia, generalmente en los sistemas tradicionales CCTV se deben duplicar los sistemas de monitorización, mientras que en un sistema de cámaras IP, las cámaras deben conectarse a la red sin necesidad de invertir en nuevos sistemas de monitorización.
- Las cámaras IP contienen un componente CCD el cual posee un número de células fotoeléctricas que se expresan en píxeles, y representan la capacidad de resolución o detalle de la imagen. Por lo tanto resulta beneficioso instalar cámaras IP con un alto número de píxeles para obtener una mayor resolución.

Para emplear cámaras IP se debe tomar en cuenta que la calidad del vídeo digital depende del sensor de la cámara, la tecnología de compresión y del software de gestión que emplea el sistema de vídeo IP. Este último es muy importante puesto que sin un software de gestión adecuado no se aprovechará al máximo los beneficios del vídeo IP porque las aplicaciones desarrolladas en el software de gestión permiten el control, administración y visualización tanto de las imágenes en tiempo como del vídeo grabado a través de la red IP. El poder del circuito de TV IP no se encuentra en la cámara sino en el software de gestión puesto que éste es el que ofrece las aplicaciones para el usuario final. Por lo tanto, el rendimiento del sistema basado en tecnología IP depende de la interacción de las cámaras IP, del hardware de grabación de la red y del software de gestión. Cuando se trabaja con un sistema como éste, no existe una única solución, sino que se trabaja con un conjunto de componentes que se distribuyen a través de la red, siendo el componente principal el software de gestión que administra los otros componentes del sistema. Entonces se podría decir que las cámaras IP no son suficientes para un sistema IP, sino que el valor real se encuentra el software de gestión, y éste componente debería ser la prioridad al momento de decidir que solución de vídeo IP se piensa utilizar.



Fuente: www.videovigilancia.us

Figura 1.1 Ejemplo de un software de gestión convencional para cámaras IP.

La figura anterior muestra como ejemplo dos distintos software de gestión para el manejo de cámaras IP. Estos tipos de software son comunes en el mercado, ya que sus funciones básicas son las de monitorización de las cámaras y grabación por horario, más no así poseen la función de detección automática de eventos, debido a que esta última necesita de más algoritmos de procesamiento.

Bajo las premisas y condiciones mencionadas anteriormente se ha diseñado y desarrollado un sistema automático de detección de eventos remotos como valor agregado a los sistemas modernos de vigilancia que utilizan cámaras de vídeo. Para emplear el sistema se ha desarrollado una aplicación que permite realizar la detección de los eventos remotos a través del procesamiento digital de imágenes, ya que esto permite minimizar costos de hardware y de personal de vigilancia. Así mismo mediante la aplicación se pretende obtener reportes diarios de los eventos sucedidos en el área visualizada por las cámaras, ya que los operarios humanos

no son buenos en el visionado de monitores durante períodos largos. Además, tener un reporte de los eventos le permite al operario confirmar el evento y derivarlo al personal correspondiente.

En la actualidad la mayoría de las cámaras IP no tienen una buena reputación para las aplicaciones de seguridad y un software de gestión permite suplir estas deficiencias. Para mejorar el desempeño de las cámaras y emplear al máximo el potencial de los sistemas de vídeo vigilancia, se están diseñando y desarrollando cámaras IP que igualan o superan la calidad de una cámara analógica.

En conclusión, el principal valor de los sistemas de seguridad IP se encuentra en el software de gestión. Sin embargo, el software depende de otros componentes como las cámaras IP (rendimiento del sensor de imagen y de la tasa y calidad de compresión de la imagen), la computadora utilizada para la administración y el monitoreo (capacidad de procesamiento y almacenamiento), la red local (Wireless o cableado estructurado), el entorno y el ambiente en donde será empleado.

CAPÍTULO 2

FUNDAMENTOS TEÓRICOS

En este capítulo se describen los fundamentos teóricos que son la base del presente trabajo de tesis. Estos fundamentos ponen en claro algunos conceptos básicos sobre los cuales se trabajará, como por ejemplo el funcionamiento de las cámaras IP, la interconexión del sistema que se quiere desarrollar y sus características principales.

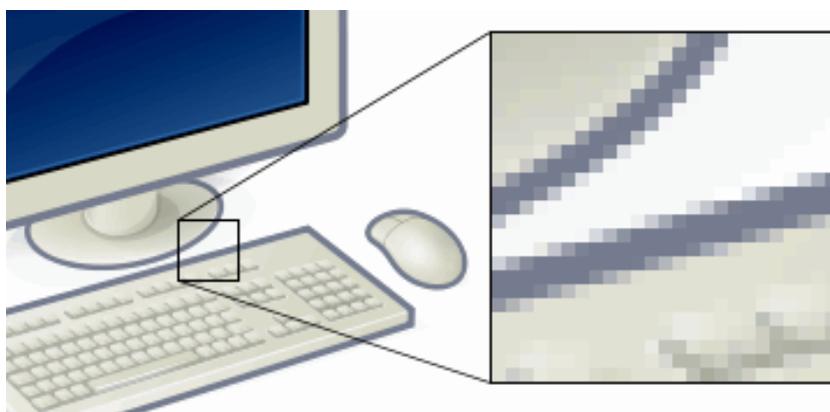
Por otra parte, además de analizar las características del sistema, también se explica detalladamente la teoría de imágenes, en la cual se comprenden conceptos tales como la definición de píxel, los modelos de color, la resolución y también algunos métodos de interpolación de imágenes que se utilizarán para implementar el zoom digital.

2.1 Teoría de imágenes

El trabajo con imágenes es una parte fundamental dentro del proyecto de tesis a realizar, ya que mediante el procesamiento de las mismas se logra la detección de los eventos predeterminados. Es por ello que en este capítulo se explicarán las características más importantes que poseen las imágenes.

2.1.1 Definición de Píxel

El píxel es la menor unidad en la que se descompone una imagen digital, ya sea una fotografía, un fotograma de vídeo o un gráfico. Los píxeles aparecen como pequeños cuadrados en color, en blanco o en negro, o en matices de gris. Las imágenes se forman como una matriz rectangular de píxeles, donde cada píxel forma un punto diminuto en la imagen total.



Fuente: es.wikipedia.org

Figura 2.1 Ampliación de un sector de una imagen.

Como se observa en la figura 2.1, cuando se amplifica una pequeña región de la imagen (zoom) se puede apreciar fácilmente a los píxeles que conforman esta imagen como si fueran pequeños cuadraditos. Los píxeles son tan diminutos que a simple vista uno no se puede percatar de su forma cuadrada.

En las imágenes de mapa de bits cada píxel se codifica mediante un conjunto de bits de longitud determinada (la llamada profundidad de color), por ejemplo, puede codificarse un

píxel con un byte, u 8 bits, de manera que cada píxel admite 256 variantes (2 dígitos por bit, elevados a la octava potencia, es decir, 2^8). En las imágenes de color verdadero, se suelen usar tres bytes para definir un color, es decir, en total se puede representar un total de 16'777,216 colores diferentes (2^{24} variantes)¹.

Para poder transformar la información numérica que almacena un píxel en un color se debe de conocer, además de la profundidad de color (el tamaño en bits del píxel), el modelo de color que se está usando, como por ejemplo el modelo de color RGB.

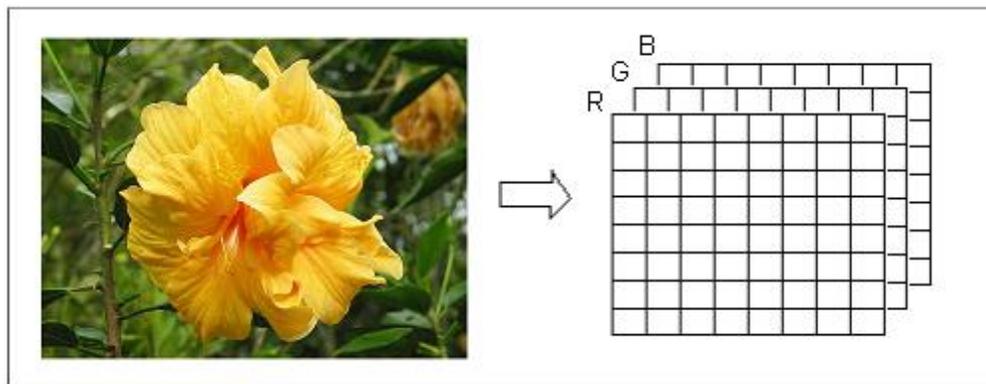


Figura 2.2 Composición de una imagen en RGB.

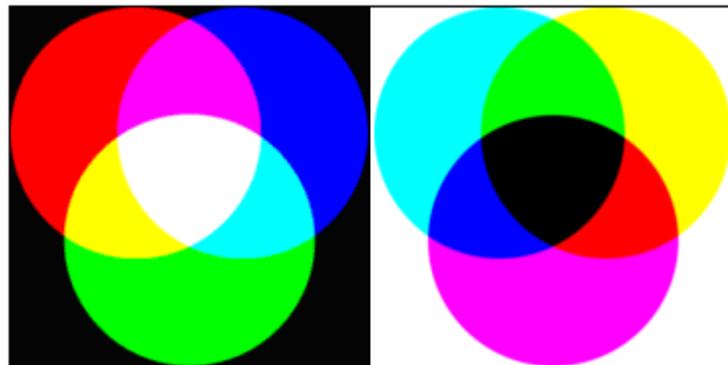
2.1.2 Modelos de color de un Píxel

Algunos de los modelos de color más conocidos que se utilizan en la composición del color de un píxel son el modelo de color RGB, CMYK y YIQ, los cuales serán descritos a continuación:

¹ Cfr. Wikipedia 2007

El modelo de color RGB (Red-Green-Blue) permite crear un color componiendo tres colores básicos: el rojo, el verde y el azul. En el modelo RGB es frecuente que se use un byte (8 bits) para representar la proporción de cada una de las tres componentes primarias. Así, de una manera estándar, la intensidad de cada una de las componentes se mide según una escala que va del 0 al 255. De esta forma, cuando una de las componentes vale 0, significa que esta no interviene en la mezcla y cuando vale 255 (2^8-1) significa que interviene aportando el máximo de ese tono.

Por lo tanto, el rojo se obtiene con (255,0,0), el verde con (0,255,0) y el azul con (0,0,255), obteniendo un color resultante monocromático. La combinación de dos colores en nivel 255 con un tercero en nivel 0 da lugar a tres colores intermedios: el amarillo (255,255,0), el cian (0,255,255) y el magenta (255,0,255).



Fuente: es.wikipedia.org

Figura 2.3 Mezcla aditiva de colores del RGB y el modelo sustractivo del CMYK.

El gráfico de la figura 2.3 muestra en su parte izquierda a la mezcla aditiva de colores del modelo RGB, en donde la combinación de sus tres componentes primarias compone al color blanco y la ausencia de sus componentes forma el color negro. En la derecha de la figura, se muestra la mezcla sustractiva del modelo CMYK.

Las ecuaciones siguientes muestran la forma de conversión del modelo RGB al modelo CMYK. Para convertir valores entre RGB y CMYK, se debe utilizar un valor CMY intermedio. Estos valores de color se deben representar como un vector, estos pueden variar entre 0.0 (color inexistente) y 1.0 (color totalmente saturado):

$$t_{CMYK} = \{C, M, Y, K\} \text{ es el cuádruplo del valor CMYK entre } [0,1]^4 \quad (2.1)$$

$$t_{CMY} = \{C, M, Y\} \text{ es el triple del valor CMY entre } [0,1]^3 \quad (2.2)$$

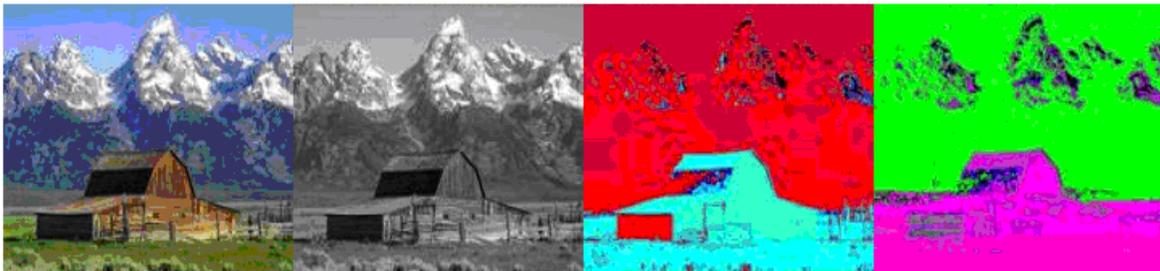
$$t_{RGB} = \{R, G, B\} \text{ es el triple del valor RGB entre } [0,1]^3 \quad (2.3)$$

El modelo CMYK (acrónimo de Cian, Magenta, Yellow y Key) es un modelo de colores sustractivo que se utiliza en la impresión a colores. Este modelo se basa en la mezcla de pigmentos de los colores Cian (C), Magenta (M), Amarillo (Y) y Negro (K) para crear otros más.

La mezcla de colores CMY ideales es sustractiva (pues imprimir cian, magenta y amarillo en fondo blanco resulta en el color negro). El modelo CMYK trabaja en base a la absorción de la luz. Los colores que se ven son de la parte de la luz que no es absorbida. El cian es el opuesto

al rojo, lo que significa que actúa como un filtro que absorbe dicho color $(-R +G +B)$. Magenta es el opuesto al verde $(+R -G +B)$ y amarillo el opuesto al azul $(+R +G -B)$.

El modelo YIQ define un espacio de color, usado antiguamente por el estándar de televisión NTSC. La componente “Y” representa la información de luminancia y es el único componente utilizado por los televisores de blanco y negro. “I” y “Q” representan la información de crominancia o del color. La luminancia (Y) viene dada por el brillo de un objeto, pudiendo producir dos objetos con tonalidades diferentes la misma sensación lumínica. La señal de luminancia es la cuantificación de esa sensación de brillo.



Fuente: www.answers.com

Figura 2.4 De izquierda a derecha, imagen con sus componentes Y, I y Q.

En la figura 2.4, de izquierda a derecha, se muestra primero la imagen en color real, luego se muestra la imagen únicamente con sus valores de luminancia Y (brillo de la imagen). Esta imagen muestra la información de crominancia I (naranja-azul) y la que sigue utiliza los valores de crominancia Q (púrpura-verde). La matriz de conversión del modelo RGB al modelo YIQ se muestra en la siguiente ecuación:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.595716 & -0.274453 & -0.321263 \\ 0.211456 & -0.522591 & 0.311135 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (2.4)$$

La representación YIQ se emplea a veces en transformaciones de procesamiento digital de imágenes en color. Por ejemplo, aplicando una ecualización del histograma directamente a los canales en una imagen RGB se alterarían los colores unos en relación con otro, resultando una imagen con colores que no tienen sentido. En vez de ello, si la ecualización del histograma es aplicada al canal Y de la representación YIQ de la imagen, sólo se normalizan los niveles de brillo de la imagen².

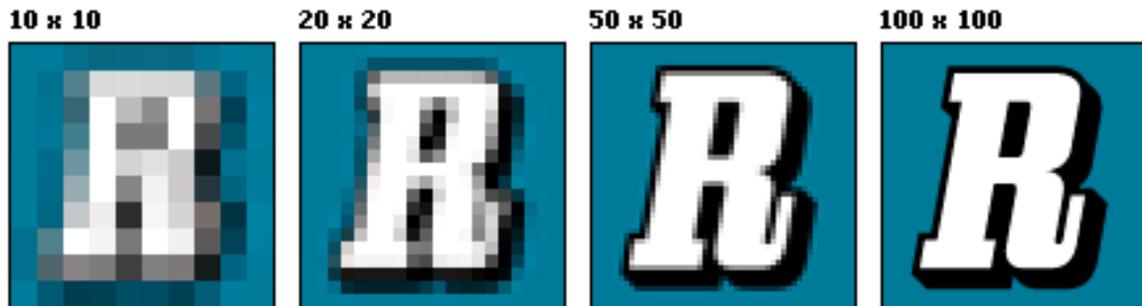
2.1.3 Resolución de imágenes

La resolución de imágenes describe cuánto detalle puede observarse en una imagen. El término es comúnmente utilizado en relación a imágenes de fotografía digital, pero también se utiliza para describir cuán nítida es una imagen de fotografía convencional. Tener mayor resolución se traduce en obtener una imagen con más detalle o calidad visual.

Para las imágenes digitales almacenadas como mapa de bits, la convención es describir la resolución de la imagen con dos números enteros, donde el primero es la cantidad de columnas de píxeles y el segundo es la cantidad de filas de píxeles. La convención que le sigue en popularidad es describir el número total de píxeles en la imagen (usualmente expresado como

² Cfr. Wikipedia 2007

la cantidad de megapíxeles), que puede ser calculado multiplicando la cantidad de columnas de píxeles por la cantidad de filas de píxeles.



Fuente: es.wikipedia.org

Figura 2.5 De izquierda a derecha, aumento de la resolución de una imagen.

Como se muestra en la figura 2.4, el primer cuadro de la izquierda posee una resolución de 10x10 (10 columnas y 10 filas de píxeles) en la que la imagen es bastante borrosa y como se observa en las imágenes siguientes hacia la derecha, a medida que la resolución aumenta también aumenta la calidad y la nitidez de la imagen³.

2.1.4 Procesamiento digital de imágenes

El procesamiento digital de imágenes es el conjunto de técnicas que se aplican a las imágenes digitales con el objetivo de mejorar la calidad o facilitar la búsqueda de información. En los últimos años el procesamiento digital de imágenes ha sido ampliamente utilizado por diversas disciplinas tales como: Medicina, Biología, Física e Ingeniería.

³ Cfr. Wikipedia 2007

Mediante el procesamiento digital de imágenes es posible manipular imágenes digitales en un computador con el fin de obtener información objetiva de la escena captada por una cámara.

Algunas de las tareas fundamentales del procesamiento digital de imágenes son:

- Mejoramiento de una imagen digital con fines interpretativos.
- Toma de decisiones de manera automática de acuerdo a los datos contenidos en la imagen digital.
- Representar eficazmente una imagen digital (compresión).



Fuente: es.wikipedia.org

Figura 2.6 Ejemplos de aplicaciones del procesamiento digital de imágenes.

La figura 2.6 muestra tres ejemplos donde es utilizado el procesamiento digital de imágenes. En el primer ejemplo el procesamiento es aplicado para lograr la detección de fallas en la soldadura, mientras que el segundo ejemplo, muestra la restauración de imágenes a partir de

dos o más imágenes distorsionadas y por último, se muestra la detección de bordes utilizando filtros digitales⁴.

Como algunas aplicaciones típicas también se pueden mencionar: la detección de presencia de objetos, la inspección visual automática, la medición de características geométricas y de color de objetos, la clasificación de objetos, la restauración de imágenes y el mejoramiento de la calidad de las imágenes.

2.2 Interpolación de imágenes

La interpolación de imágenes es el proceso por el cual se determinan valores de píxeles desconocidos a partir de otros ya conocidos mediante la aplicación de algoritmos concretos. Por ejemplo, si se recalibra una imagen de manera que contenga el doble del número de píxeles que tenía originalmente, se tiene que obtener valores para los nuevos píxeles a través de la interpolación.

Es importante tener en cuenta que la interpolación de una imagen a un tamaño concreto (imagen amplificada) nunca conseguiría la misma calidad de una imagen en su tamaño original, ya que en este último caso la información de píxeles es real y no inventada⁵.

2.2.1 Tipos de interpolación

Algunos de los métodos de interpolación más conocidos son:

⁴ Cfr. Wikipedia 2007

⁵ Cfr. *Transformaciones geométricas* 2005

- Interpolación de orden cero o por réplica.
- Interpolación bilineal.
- Interpolación bicúbica.

La interpolación por vecino más cercano adapta una superficie constante a través de los valores de intensidad. El valor de un píxel interpolado es el valor del píxel más cercano.

La interpolación bilineal ajusta una superficie lineal a través de los valores de intensidad existentes. El valor de un píxel interpolado es una combinación de los valores de los 4 píxeles más cercanos. La interpolación bilineal es lineal, más rápida y ocupa menos procesamiento que la interpolación bicúbica.

La interpolación bicúbica ajusta una superficie cúbica a través de los valores de intensidad existentes. El valor de un píxel interpolado es una combinación de los valores de los 16 píxeles más cercanos. Este método produce una superficie mucho más suave que la interpolación bilineal. Para usar la interpolación bicúbica, la primera y segunda derivada de la superficie deben ser continuas⁶.



Fuente: Transformaciones geométricas (García Mateos, Ginés)

Figura 2.7 Comparación entre los métodos de interpolación.

⁶ Cfr. *Transformaciones geométricas*

En la figura 2.7 se compara gráficamente la diferencia en calidad visual obtenida al ampliar una imagen con los tres métodos de interpolación mencionados anteriormente.

2.2.2 Interpolación bilineal

La interpolación bilineal consiste en aplicar dos interpolaciones lineales. La primera interpola la imagen horizontalmente en las filas existentes. Mientras que, la segunda interpola la imagen verticalmente en todo el espacio.

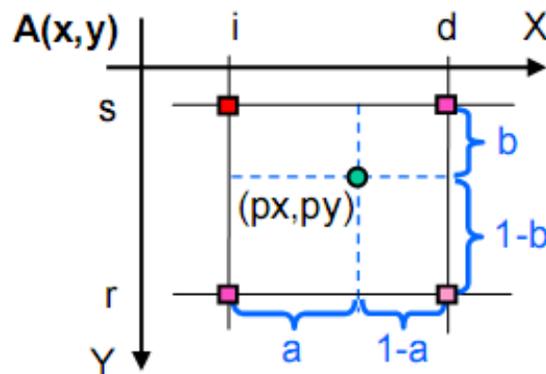


Figura 2.8 Valor interpolado entre sus 4 vecinos más cercanos.

Como se muestra en la figura 2.8, en este método de interpolación el píxel a interpolar, en la posición (px,py) , forma 4 regiones con sus 4 vecinos más próximos (al trazar una línea horizontal y vertical). Estas 4 regiones indican la influencia que tiene cada vecino en el píxel interpolado, por ejemplo el píxel de la esquina superior izquierda “posición: (s,i) ” tendrá una influencia tal como el área formada en la región más opuesta a él. En otras palabras, mientras el píxel a interpolar este más cerca a uno de sus 4 vecinos, éste tendrá mayor influencia en el valor del nuevo píxel.

A continuación se muestra la ecuación final para obtener el valor del píxel interpolado:

$$A^3(px, py) = (1-a) \times (1-b) \times A(s, i) + a \times (1-b) \times A(s, d) + (1-a) \times b \times A(r, i) + a \times b \times A(r, d) \quad (2.5)$$

La interpolación bilineal mejora la interpolación del vecino más próximo, pero produce un efecto de “zonas rectangulares”. A pesar de esto, este tipo de interpolación consume mucho menos procesamiento que la interpolación bicúbica y la calidad de su ampliación es aceptable. Es por ello, que se decidió implementar este tipo de interpolación para realizar el zoom digital a las imágenes capturadas durante la detección automática de eventos⁷.

2.3 Cámaras IP

Una cámara IP es una videocámara especialmente diseñada para enviar las señales de video y/o audio, a través de Internet desde un explorador o a través de un concentrador en una Red Local. Para cumplir con este propósito, una cámara IP contiene su propia dirección IP, un servidor web (una interfaz TCP/IP) para gestionar la comunicación en la red y un circuito de compresión.



Fuente: www.ipcam.com.mx

⁷ Cfr. *Transformaciones geométricas*

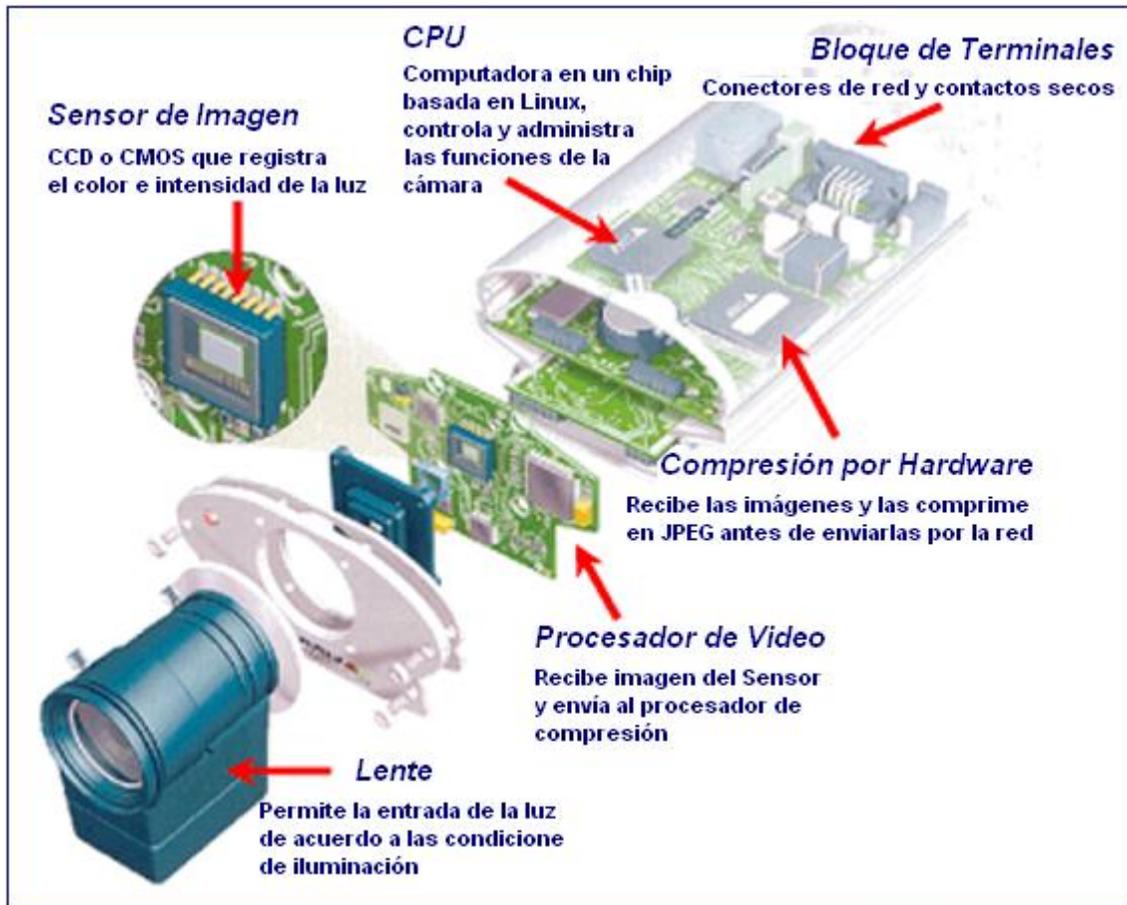
Figura 2.9 Diferentes modelos de cámaras IP.

En la figura superior se muestran los modelos de cámaras IP más comunes del mercado. Entre estos modelos se encuentran: las cámaras fijas, las cuales no realizan movimiento y necesitan de una conexión física Ethernet para transmitir las imágenes a los clientes; las cámaras inalámbricas, que sólo requieren de un “*Access Point*” que las comunique con Internet y de su fuente de alimentación; también están las cámaras tipo domo, las cuales además de poseer movimiento vertical y horizontal también poseen una cubierta para trabajar en ambientes exteriores; y por último están las cámaras con movimiento y un control PTZ agregado.

Las cámaras IP más avanzadas también pueden equiparse con muchas otras funciones de valor añadido como integrar aplicaciones de detección de movimiento, envío de correos, grabación de imágenes o secuencias. En algunos modelos pueden incluir entradas para alarmas, entrada/salida de video analógico y salida de relé⁸.

2.3.1 Esquema de una cámara IP

⁸ Cfr. Configurar Equipos 2002



Fuente: www.gscssoftware.com

Figura 2.10 Diagrama de bloques de una cámara IP.

Como se observa en la figura 2.10, una cámara IP posee un esquema compuesto básicamente por los siguientes componentes⁹:

- La cámara de video tradicional (lentes, sensores, procesador digital de imagen).
- Un sistema de compresión de imagen (para poder comprimir las imágenes captadas por la cámara a formatos adecuados como: JPEG, MJPEG, MPEG Y MPEG4).

⁹ Cfr. GSCS Software 1997

- Un sistema de procesamiento (CPU y un módulo Wireless). Este sistema de procesamiento se encarga de la gestión de las imágenes y del envío al módem.

2.3.2 Aplicaciones comunes

Estas cámaras comúnmente son utilizadas para vigilancia de estacionamientos, lobbies de hoteles, áreas de recepción de materiales, bodegas, control de acceso de personal, vigilancia de propiedades privadas (condominios, barcos, etc.). Por otra parte, cuando se utilizan para monitorización es usual que monitorean el control de acceso a data centers, trabajos de construcción, líneas de producción, enfermería, cuidado de niños, colas en cajas, etc.

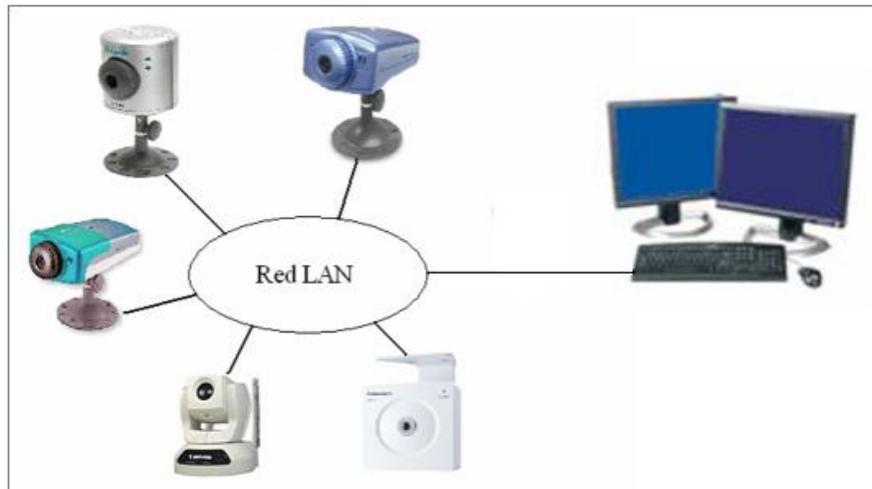


Figura 2.11 Ejemplo de monitorización remota.

En la figura 2.11 se muestra un ejemplo de cómo se efectúa la monitorización remota, en una red local (LAN), a través de la cual se acceden a los distintos tipos de cámaras IP.

2.4 Interconexión del sistema

El tipo de conexión requerida para el sistema es del tipo LAN (Red de Área Local o simplemente Red Local), ya que permite que dos o más dispositivos de red se comuniquen. En este caso los dispositivos a conectar en la red local vendrían a ser las cámaras IP.

2.4.1 Red de Área Local (LAN)

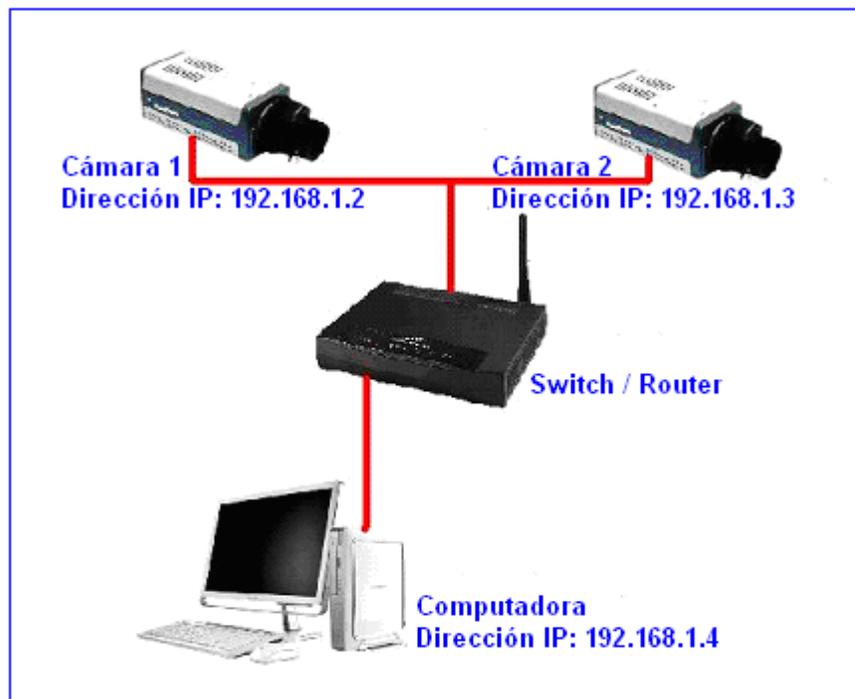


Figura 2.12 Esquema de la conexión del sistema.

Una red local es la interconexión de varios ordenadores y periféricos. Un ejemplo de una red local es la que se muestra en la figura 2.12, en la que cada dispositivo de red lleva una

dirección IP privada asignada automáticamente, la cual sirve para identificar exclusivamente a cada uno de los dispositivos dentro de esta red local. La extensión de la conexión tipo LAN, está limitada físicamente a un edificio o a un entorno de unos pocos kilómetros. Su aplicación más extendida es la interconexión de ordenadores personales y estaciones de trabajo (en oficinas, fábricas, etc.) para compartir recursos e intercambiar datos y aplicaciones.

2.4.2 Protocolo IP

En la capa de Internet, el Protocolo IP se encarga de la comunicación de una máquina a otra proporcionando un servicio de entrega de paquetes sin conexión. El protocolo IP especifica el formato de los datos que viajarán a través de Internet, para lo cual realiza cuatro funciones fundamentales:

- Selecciona la ruta para el envío de los datos.
- Implementa las reglas que definen cómo y de qué manera se deben procesar los paquetes de datos.
- Proporciona un método eficaz para identificar las máquinas en la red.
- Implementa el (ICMP) un sistema de información y corrección de errores.

Datagrama de petición o respuesta (bytes)	$2048 = 2000 \text{ (datos)} + 40 \text{ cab. IP (20 x 2 fragmentos)} + 8 \text{ cab. ICMP}$
1 ^{er} Fragmento (bytes)	$1500 = 20 \text{ cabecera IP} + 8 \text{ cabecera ICMP} + 1472 \text{ datos}$
2 ^{do} Fragmento (bytes)	$548 = 20 \text{ cabecera IP} + 528 \text{ datos}$

Fuente: <http://www4.ujaen.es/~mdmolina>

Figura 2.13 Estructura de la trama enviada mediante el protocolo IP.

En la figura 2.13 se muestra la estructura de la trama enviada cuando se utiliza el protocolo IP. Es trama está compuesta de 2048 bytes divididos en dos fragmentos, de los cuales 2000 bytes son para los datos, 40 bytes sirven para la cabecera IP y por último 8 bytes funcionan como cabecera ICMP. El primer fragmento transporta 20 de los 40 bytes de la cabecera IP, también lleva los 8 bytes de la cabecera ICMP y 1472 bytes de datos. Mientras que el segundo fragmento transporta los otros 20 bytes de la cabecera IP y los 528 bytes restantes de datos¹⁰.

2.5 Sistema de vigilancia con cámaras IP

Un sistema de cámaras IP consiste en la utilización de cámaras de vigilancia distribuidas de forma estratégica que permiten el control de las actividades que acontecen o se desarrollan dentro del ambiente supervisado. Si el entorno o ambiente está equipado con una red IP, entonces ya se cuenta con la infraestructura necesaria para incorporar las cámaras IP. Dado que las cámaras IP se conectan directamente a la red existente a través de un puerto Ethernet, entonces la instalación del sistema de cámaras de vigilancia IP es más rápida y económica.

La ventaja de los sistemas de cámaras de vigilancia IP frente a los sistemas analógicos es que se puede realizar la supervisión desde cualquier computadora que esté conectada en la LAN, además cabe la posibilidad de acceder al sistema a través de Internet, por lo tanto se puede realizar un mejor control puesto que hay más personas atentas a detectar algún evento o hecho

¹⁰ Cfr. Dolores Molina González 2006

sospechoso. Otra ventaja importante es que el sistema es escalable pudiéndose iniciar con una configuración mínima de cámaras y a medida que se considere adecuado incorporar nuevas al sistema existente.

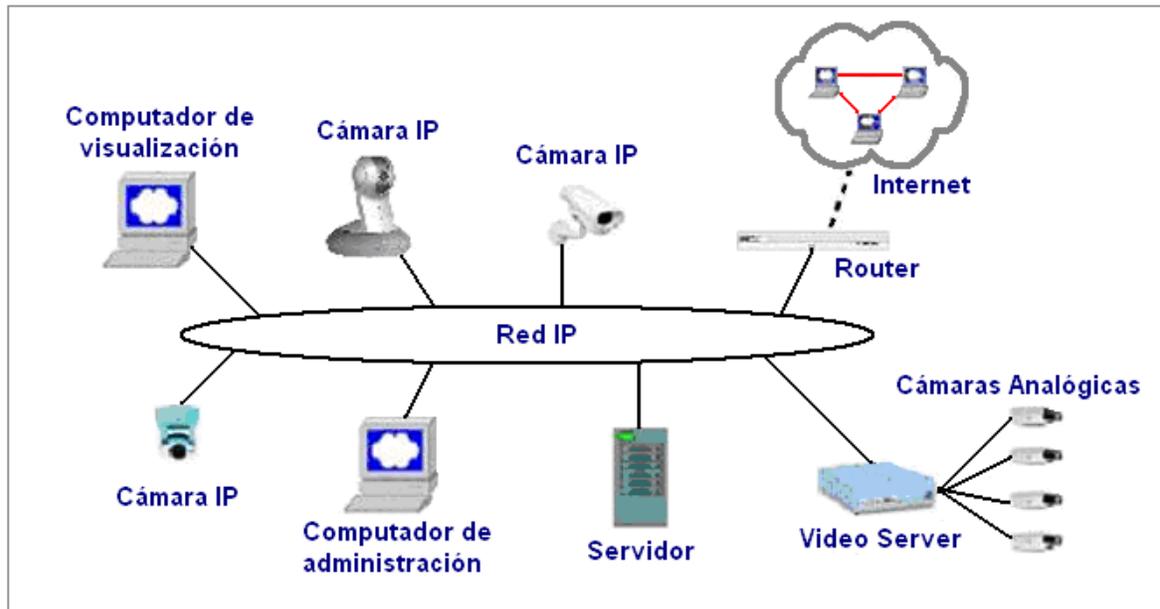


Figura 2.14 Empleo del sistema de vigilancia IP.

En la figura 2.14 se muestra una red IP, en la cual se encuentran ubicados distintos dispositivos de red para vigilancia, tales como cámaras IP, servidores, cámaras analógicas con un Video Server de interfaz

En un sistema de cámaras de vigilancia IP, las cámaras son instaladas en lugares ubicados a una altura adecuada para permitir una visión panorámica de la zona a controlar y supervisar, esto también reduce la posibilidad de la destrucción de la cámara o de su reorientación¹¹.

¹¹ Cfr. IndigoVision 2006

2.5.1 Programa de gestión de vídeo

El programa de gestión de vídeo funciona sobre un servidor Unix/Linux o Windows, que establece la base para la grabación, análisis y monitoreo de vídeo. En la actualidad se encuentran disponibles una amplia gama de programas de gestión de vídeo que se basan en las necesidades de los usuarios.

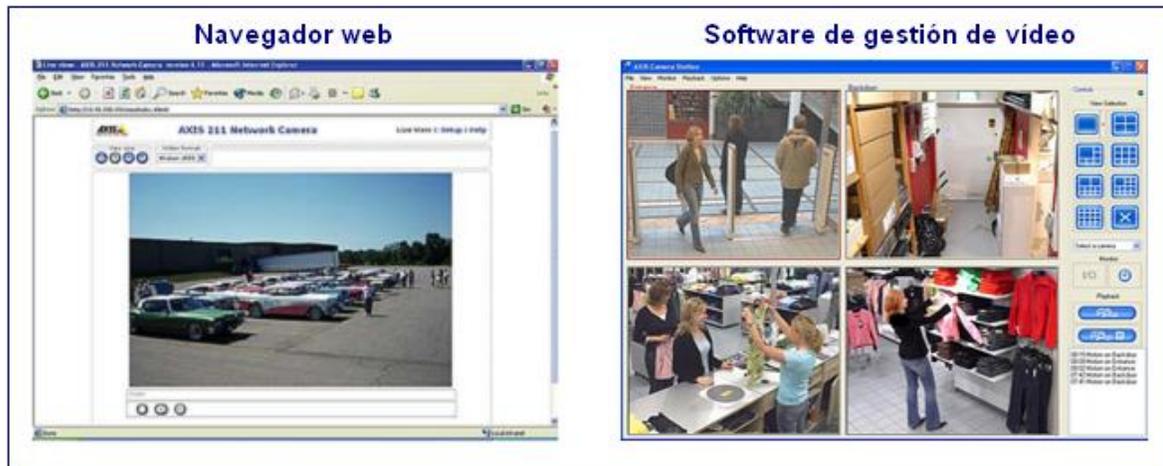
La diferencia entre un navegador web y un programa de gestión es que un navegador web proporciona la visualización para las aplicaciones de vídeo IP, empleando la interfaz web integrada en las cámaras IP o en los servidores de vídeo, en casos en que se pueden visualizar una o pocas cámaras de forma simultánea. Mientras que un programa de gestión de vídeo permite visualizar diversas cámaras al mismo tiempo.

Por lo tanto, un programa de gestión de vídeo ofrece visualización en directo, almacenamiento y recuperación de secuencias de imágenes de vídeo. Un programa avanzado proporciona e incluye las siguientes características¹²:

- Visualización en forma simultánea y grabación de vídeo de múltiples cámaras.
- Diversos modos de grabación: grabaciones continuas, grabaciones programadas, por activación de alarmas y por detección de movimiento.
- Capacidad para el manejo y control de gran cantidad de datos a altas velocidades.
- Múltiples funciones de búsquedas para los eventos grabados.

¹² Cfr. Axis 2004

- Permite el acceso remoto a través del navegador web, con un programa cliente.
- Control de cámaras PTZ y Domos.
- Soporte de sistema de audio en tiempo real (full dúplex).
- Establece y administra las funciones de gestión de alarmas (notificaciones de alarmas, ventanas desplegadas o correo electrónico).
- Vídeo inteligente.



Fuente: www.axis.com

Figura 2.15 Formas de visualización y administración de las cámaras IP.

2.5.2 Ancho de banda consumido por una cámara IP

La fórmula para calcular el consumo de ancho de banda es:

$$\text{AnchodeBanda} = \text{Tamaño_imagen(bytes)} \times \text{Frecuencia_captura(fps)} \quad (2.5)$$

Tabla 2.1 Detalle de la imagen capturada.

Tiempo	Cantidad de cuadros	Tamaño de la imagen
1 seg.	5	25 KB

El cálculo estimado del espacio requerido por el sistema se realizará en base a los siguientes parámetros¹³:

- Formato de imagen: 320 píxeles x 240 píxeles
- Frecuencia de captura: 5 fps (cuadros por segundo)
- Tamaño de la imagen: 25 KBytes

Por lo tanto, el ancho de banda consumido es: $25 \text{ KB} \times 5 \text{ fps} = 125 \text{ Kbps}$.

2.5.3 Capacidad requerida por el sistema

El sistema ante una detección de un evento comienza a grabar continuamente, lo que ocupará un determinado espacio (en bytes) en la computadora según el tiempo que permanezca la grabación. El cálculo estimado de este espacio requerido por el sistema se realiza en base a los siguientes parámetros¹⁴:

- Formato de imagen: 320 píxeles x 240 píxeles
- Cantidad de eventos: 1
- Cantidad de cámaras: 1
- El espacio requerido por el sistema está calculado en base a la compresión mínima de las imágenes en formato JPEG.

¹³ Cfr. Ydeon 2007

¹⁴ Cfr. Ydeon 2007

Tabla 2.2 Espacio requerido del sistema para una cámara.

Tiempo	Cantidad de cuadros	Espacio requerido
1 seg.	5	25 KB
1 min.	300	1.5 MB
1 hora	18000	90 MB
24 horas	432000	2.16 GB

Los resultados mostrados en la tabla superior han sido obtenidos considerando que el sistema de detección esta grabando continuamente con una sola cámara conectada. En caso haya más de una cámara conectada, para obtener el espacio requerido por el sistema durante la grabación, se tendrá que multiplicar el número de cámaras conectadas con el espacio requerido obtenido para una sola cámara.

CAPÍTULO 3

ALGORITMOS DESARROLLADOS PARA EL PROYECTO

El desarrollo de algoritmos de procesamiento de imágenes viene a ser una de las partes más importantes dentro del proyecto de tesis, ya que gracias a estos algoritmos se logra realizar la detección automática de eventos. A continuación, se explicará detalladamente las características de funcionamiento de los algoritmos utilizados.

3.1 Algoritmo de detección de movimiento

Para llevar a cabo la detección de movimiento se ha decidido utilizar como variable principal de análisis a la componente de luminancia de las imágenes (Y), ya que esta variable permite conocer la intensidad de luz de cada píxel en una imagen. La variación de esta intensidad de luz es la que permite conocer cuando un objeto ha modificado su posición. Su cálculo se realiza a partir de los valores de RGB de cada píxel de la imagen, mediante la siguiente matriz de transformación:

$$\begin{bmatrix} Y \\ I \\ Q \end{bmatrix} = \begin{bmatrix} 0.299 & 0.587 & 0.114 \\ 0.595716 & -0.274453 & -0.321263 \\ 0.211456 & -0.522591 & 0.311135 \end{bmatrix} \begin{bmatrix} R \\ G \\ B \end{bmatrix} \quad (3.1)$$

Es necesario que antes de aplicar el algoritmo de detección de movimiento, el usuario determine el número de objetos a analizar dentro de la imagen y también que seleccione las

regiones de estos objetos. El algoritmo que se muestra líneas abajo no recorrerá toda el área de la imagen, sino que solamente recorrerá las áreas de los objetos seleccionados para así minimizar el tiempo de procesamiento.

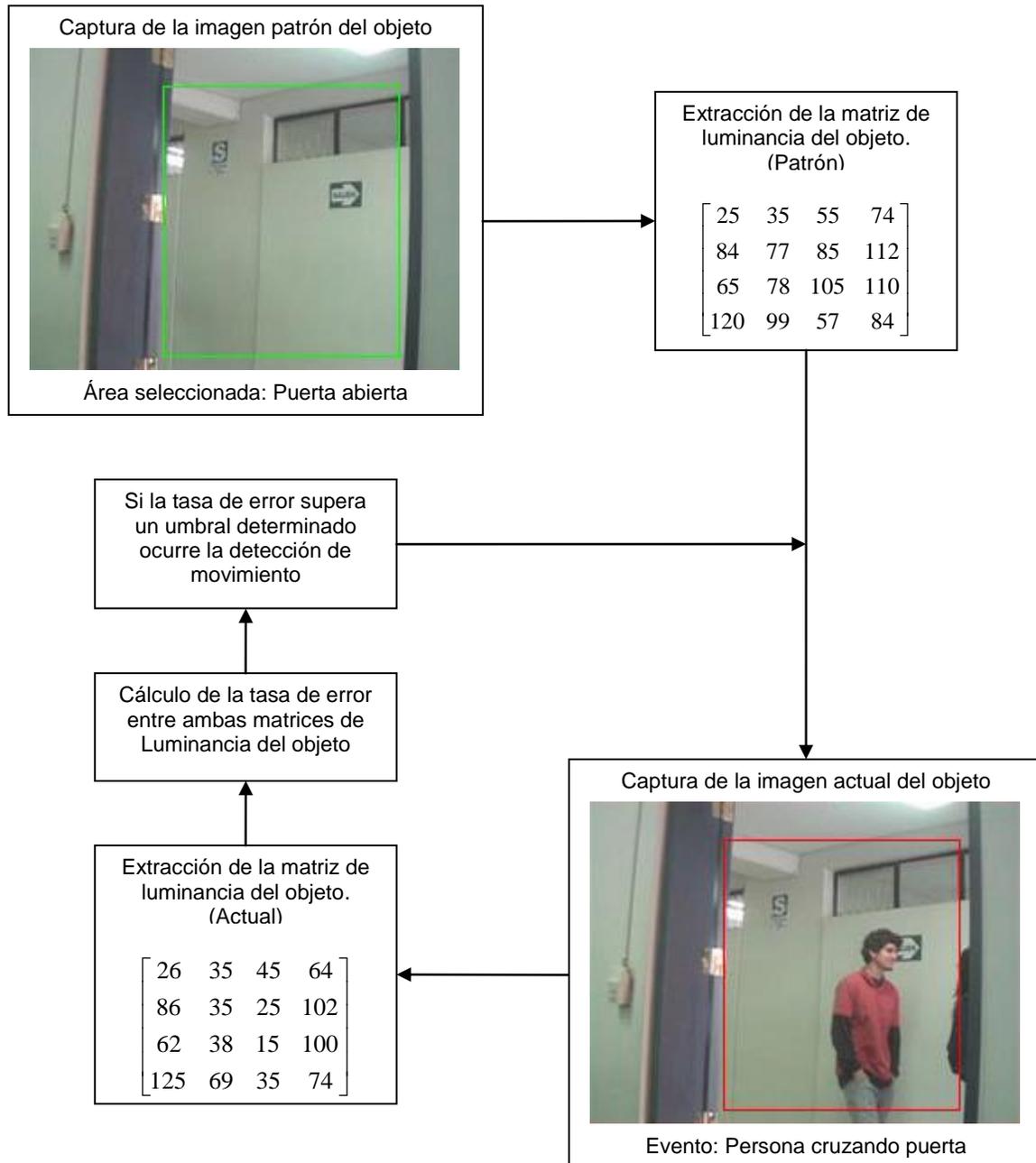


Figura 3.1 Diagrama de bloques para la detección de movimiento.

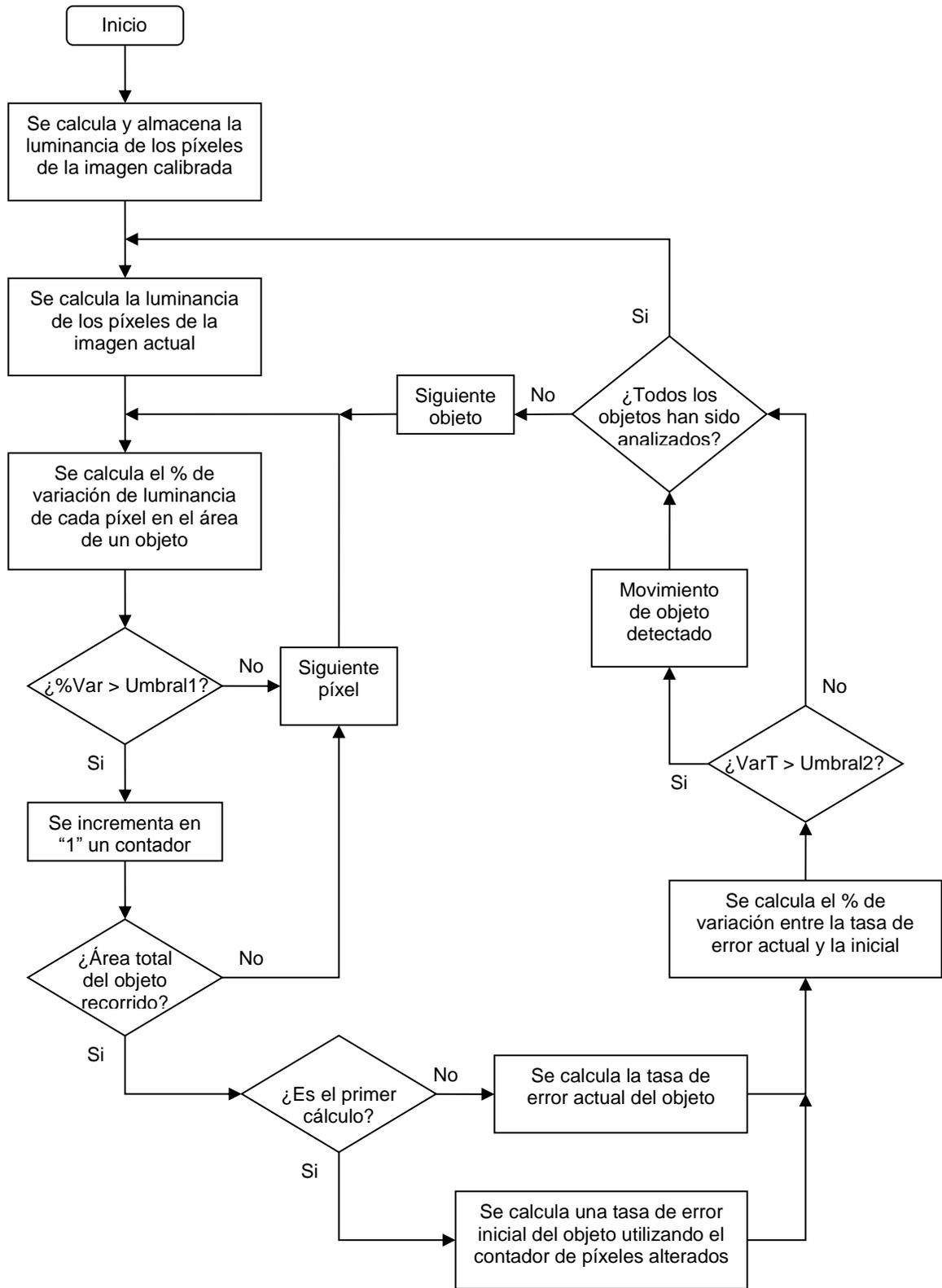


Figura 3.2 Diagrama de flujo para la detección de movimiento.

El algoritmo mostrado funciona de la siguiente manera: primero calcula y almacena el valor de la componente de luminancia de todos los píxeles de la imagen calibrada en una matriz $Y_{cal}(x,y)$. Esta imagen calibrada se captura inicialmente al momento que el usuario determina los objetos a detectar, y sirve como una imagen patrón, ya que en ella aún no se han registrado alteraciones en los píxeles.

Luego, se requiere realizar el mismo proceso de cálculo de los valores de la componente de luminancia de cada píxel, pero esta vez de la imagen actual, la cual es recibida en tiempo real. Dichos valores de luminancia son almacenados en la matriz $Y_{act}(x,y)$ y una vez obtenidos los valores de luminancia de los píxeles en ambas matrices se procede a recorrer el área de todos los objetos seleccionados calculando el porcentaje de variación que ha tenido cada píxel.

$$\%Var(x, y) = \left(\frac{Y_{act}(x, y) - Y_{cal}(x, y)}{255} \right) \times 100\% \quad (3.2)$$

Si este porcentaje de variación supera un primer umbral ($Umbral1$) igual al 15% entonces se considera que dicho píxel se ha alterado y se incrementa en una unidad el valor de un contador de píxeles alterados (existe uno para cada objeto). El valor del primer umbral ha sido calibrado a 15%, ya que para porcentajes menores es posible que el objeto no se haya alterado y que el porcentaje de variación de luminancia generado haya sido causado por el ruido que se genera al capturar la imagen. Luego, se prosigue con el siguiente píxel, y si ya se analizaron todos los píxeles del objeto entonces se calcula su tasa de error, utilizando el valor acumulado del contador de píxeles alterados de dicho objeto.

$$Tasa_{error} = \left(\frac{\text{Contador de píxeles alterados}}{M \times N} \right) \times 100\% \quad (3.3)$$

Si es la primera vez que se calcula la tasa de error, entonces ésta es guardada como un valor de tasa de error inicial. Es necesario guardar la tasa de error inicial, ya que no se puede suponer que sea cero a pesar que la imagen no presente alteraciones, debido a que el ruido que se acopla al capturar la imagen puede hacer que se genere una tasa de error baja o alta dependiendo del contraste de la imagen capturada.

Una vez que se posee una tasa de error inicial almacenada, las siguientes tasas de error que se calculen se van a actualizar en tiempo real. De esta manera, cuando la diferencia ($VarT$) entre la tasa de error inicial y la tasa de error actual supere un pequeño umbral ($Umbral2$) igual al 2% de error se considerará que el objeto analizado se ha movido.

$$VarT = Tasa_{actual} - Tasa_{inicial} \quad (3.4)$$

Después de esta comparación, si aún quedan más objetos por analizar dentro de la imagen, entonces se prosigue con el siguiente objeto, de lo contrario se repite el procedimiento para todos los objetos actualizando el valor de luminancia de todos los píxeles de la imagen y analizando las regiones de todos los objetos seleccionados.

3.2 Algoritmo de detección de cambio de iluminación

El algoritmo desarrollado para la detección de cambio de iluminación es bastante similar al algoritmo de detección de movimiento, ya que también se utiliza como variable principal de

análisis a la componente de luminancia de las imágenes (Y). A diferencia del algoritmo de detección de movimiento, ahora no se va a analizar el área de cada objeto seleccionado, sino que se analizará el área de toda la imagen, ya que un cambio de iluminación en la escena debería afectar a la mayoría de píxeles de la imagen.

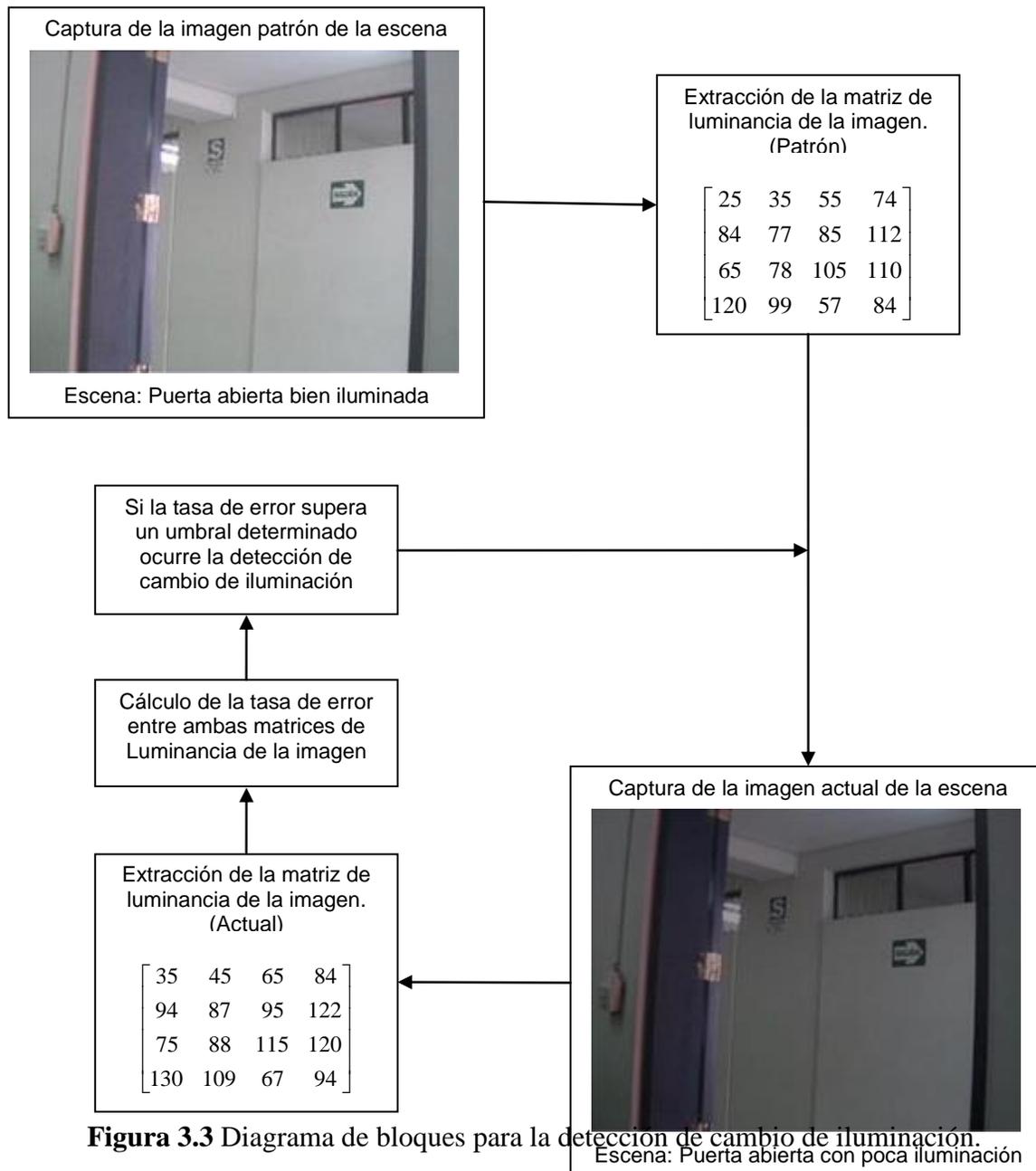


Figura 3.3 Diagrama de bloques para la detección de cambio de iluminación.

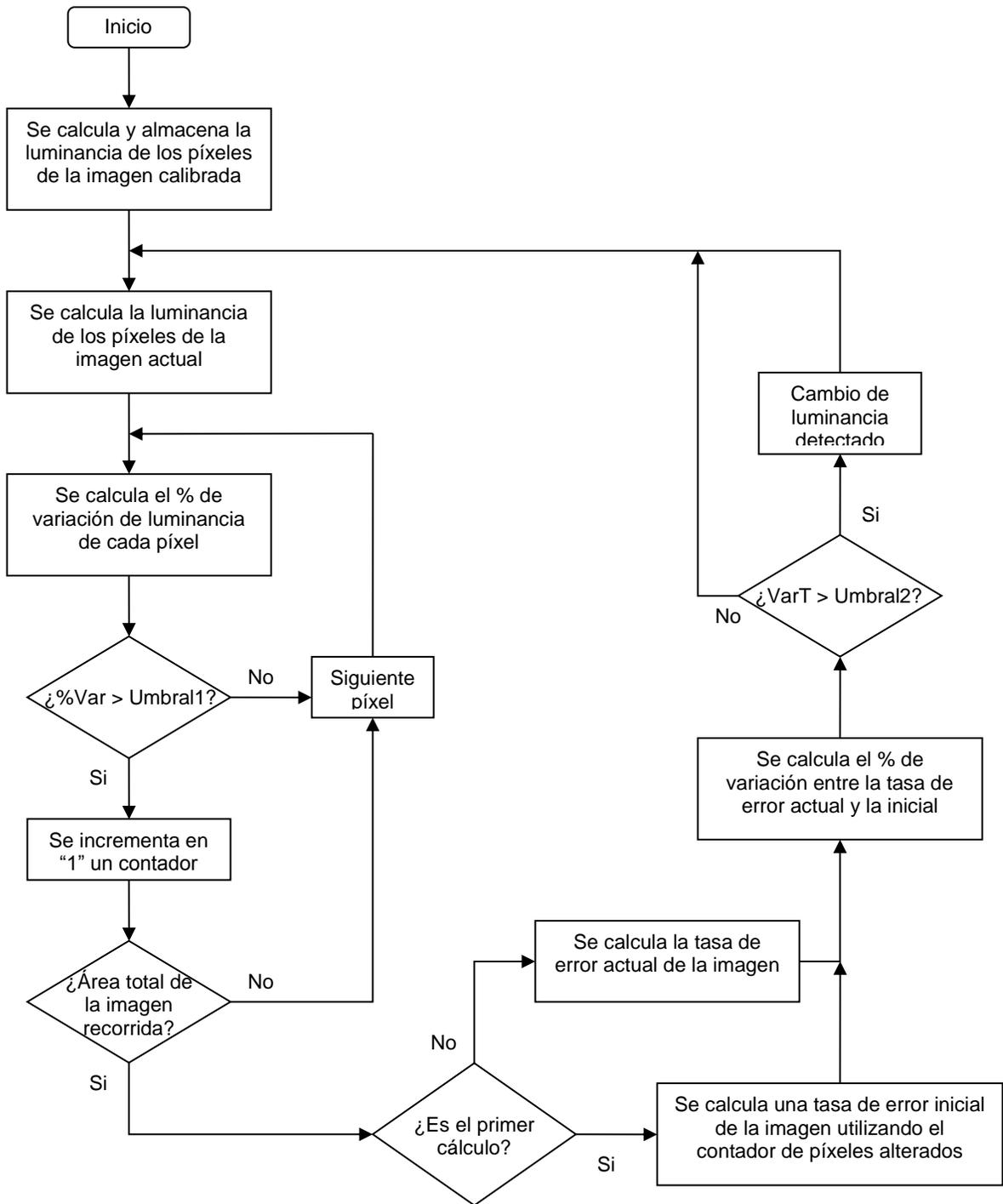


Figura 3.4 Diagrama de flujo para la detección de cambio de iluminación.

El desarrollo de este algoritmo, de igual forma que el algoritmo de detección de movimiento, comienza con el cálculo y almacenamiento del valor de la componente de luminancia de todos los píxeles de la imagen calibrada en una matriz $Y_{cal}(x,y)$ y también de los píxeles de la imagen actual en una matriz $Y_{act}(x,y)$. Y una vez obtenidos los valores de luminancia de los píxeles en ambas matrices se procede a recorrer el área de toda la imagen actual calculando el porcentaje de variación que ha tenido cada píxel, utilizando para ello la fórmula dada en la ecuación (3.2). Los valores de luminancia de ambas matrices se calculan recorriendo el área de la imagen, y aplicando la siguiente ecuación de conversión a las componentes RGB de cada píxel:

$$Y(x, y) = 0.299 \times R(x, y) + 0.587 \times G(x, y) + 0.114 \times B(x, y) \quad (3.5)$$

Para este tipo de detección de cambio de iluminación se considera que si el porcentaje de variación supera un primer umbral ($Umbral1$) igual al 5% entonces dicho píxel se ha alterado y se incrementa en una unidad el valor de un contador de píxeles alterados (existe uno para cada objeto). El valor del primer umbral ha sido calibrado a 5%, ya que como se trata de detectar cambio de iluminación el umbral no debe ser muy elevado puesto que un cambio de iluminación sólo modifica en un pequeño rango el valor de luminancia de los píxeles. Luego, se prosigue con el siguiente píxel, y si ya se analizaron todos los píxeles de la imagen entonces se calcula su tasa de error, utilizando la fórmula dada en la ecuación (3.3) y reemplazando en ésta el valor acumulado del contador de píxeles alterados de la imagen.

Al igual que en la detección de movimiento, si es la primera vez que se calcula la tasa de error, entonces ésta es guardada como una tasa de error inicial. De lo contrario, las siguientes tasas de error que se calculen se van a actualizar en tiempo real. De esta manera, cuando la diferencia (VarT) entre la tasa de error inicial y la tasa de error actual supere un umbral alto (Umbral2) igual al 75% de error se considerará que la imagen capturada de la escena presenta un cambio de iluminación. Este segundo umbral es muy elevado ya que para suponer que la escena ha sufrido un cambio de iluminación la mayoría de los píxeles de la imagen actual deberían haberse alterado por lo menos en un pequeño rango. Luego de analizar si existe la detección se procede a actualizar el valor de luminancia de todos los píxeles de la imagen actual para poder reiniciar el proceso de detección.

3.3 Algoritmo de almacenamiento de vídeo

El algoritmo desarrollado para el almacenamiento de vídeo permite generar archivos de vídeo en formato AVI siempre y cuando se detecte algún evento. Esta característica permite que no sea necesario tener un modo de grabación constante, lo cual requeriría de mucho espacio de almacenamiento. De esta manera, el sistema desarrollado es mucho más eficiente e incluso inteligente, ya que sólo realiza la grabación de vídeo en el momento que se detecta algún evento ocurrido.

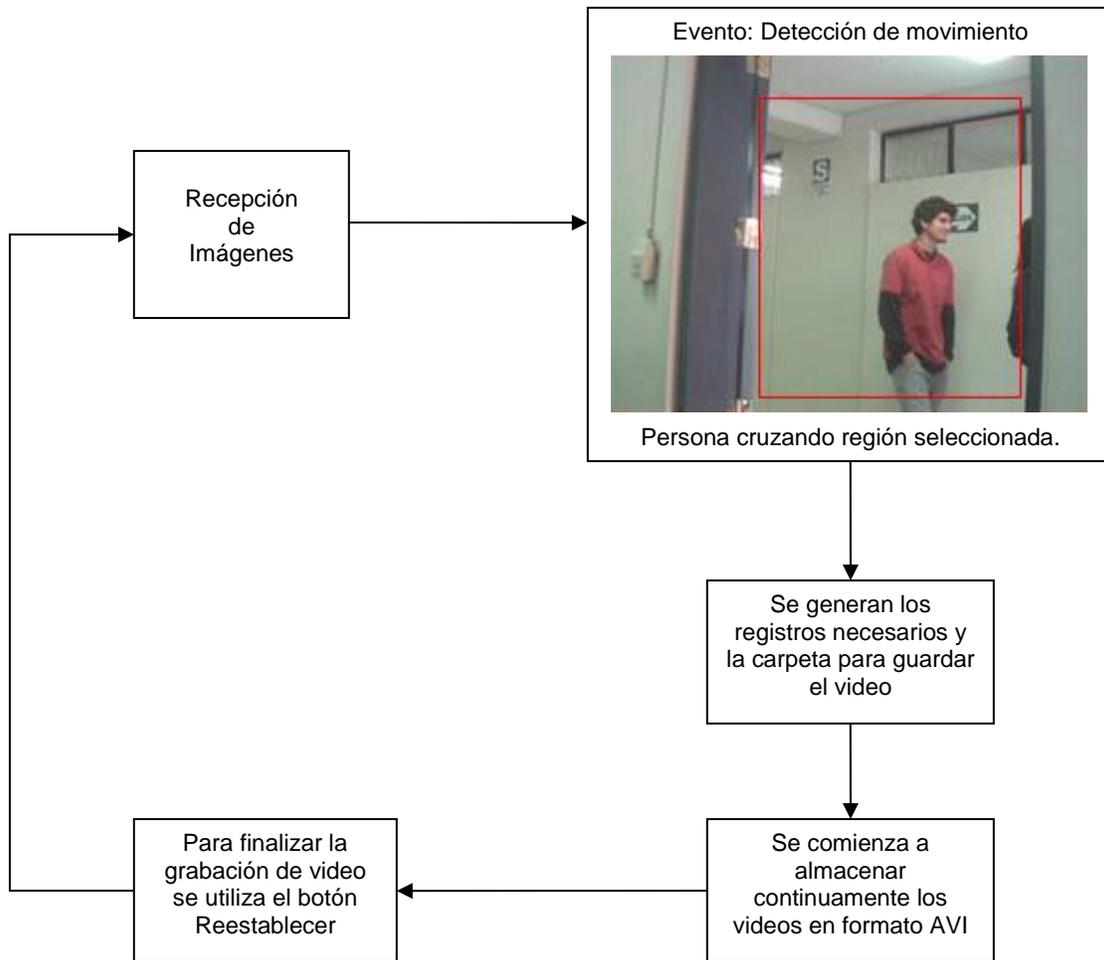


Figura 3.5 Diagrama de bloques del algoritmo para almacenar vídeo.

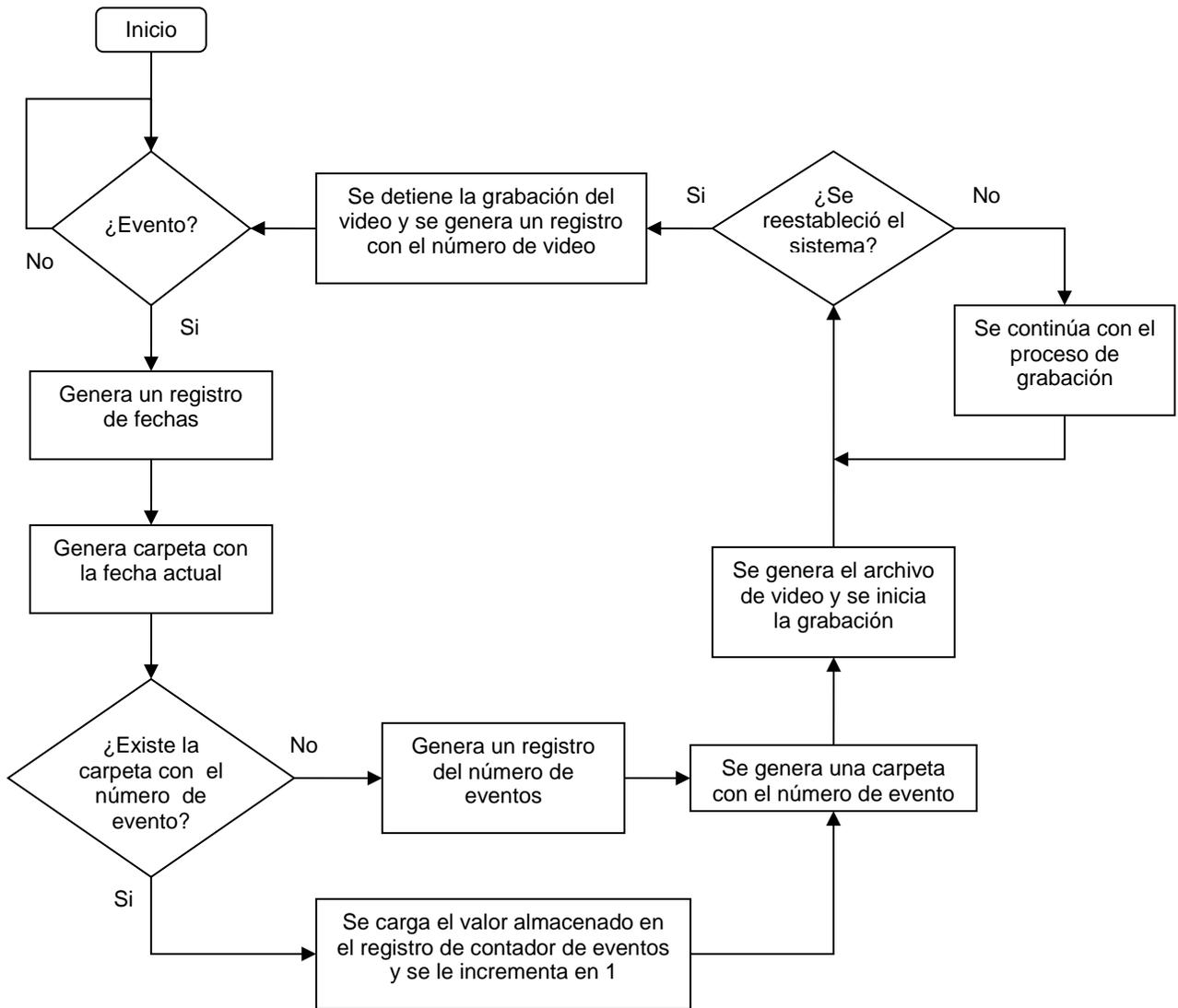


Figura 3.6 Diagrama de flujo del algoritmo para almacenar vídeo.

El algoritmo mostrado consiste en estar verificando constantemente la ocurrencia de eventos, ya que de no ocurrir ninguno no se podría iniciar la secuencia del algoritmo. Al momento de detectarse un evento, lo primero que realiza el algoritmo es verificar si existe un archivo de registro en el cual esté almacenado la fecha actual, ya que de no existir dicho registro entonces se debe generar uno y además crear una carpeta que lleve de nombre la fecha actual (fecha en la que ocurrió el evento detectado). En este caso, como no existía el registro con la fecha actual entonces también se debe generar un archivo de registro del número de eventos en el cual se ingresa el valor de 1, ya que correspondería al primer evento detectado en la fecha actual. De lo contrario, en el caso que ya haya existido un registro con la fecha actual, esto indicaría que ha ocurrido un evento anteriormente en esta fecha, por lo que el algoritmo ya no generaría un registro con el número de eventos, sino que solamente incrementaría en una unidad al valor que está en el registro del número de eventos para luego cargar dicho valor al contador de eventos del programa.

Luego de conocer el número de evento correspondiente a la detección se procede a crear una carpeta con el nombre del número de evento. En esta carpeta es en la que se va a generar el archivo en formato AVI y grabará continuamente mientras no se restablezca el sistema. Una vez que el usuario se percate del evento ocurrido, éste debe presionar el botón de *Reestablecer Sistema* para que así el algoritmo detenga la grabación y reinicie las variables de detección para así poder continuar con la detección de otros eventos.

3.4 Algoritmo de almacenamiento de imágenes

El algoritmo desarrollado para el almacenamiento de imágenes es básicamente el mismo que el algoritmo para almacenar vídeo con la principal diferencia de que en lugar de guardar sólo un archivo de vídeo en formato AVI en este caso se guardan múltiples imágenes capturadas en formato JPEG, las cuales al reproducirse en el mismo tiempo en que han sido capturadas dan la impresión de ser un vídeo continuo. Este algoritmo captura imágenes en vez de vídeo por la razón principal de que es menos complejo analizar imágenes en formato JPEG que analizar un sólo archivo de vídeo, en el cual la información de píxeles está comprimida.

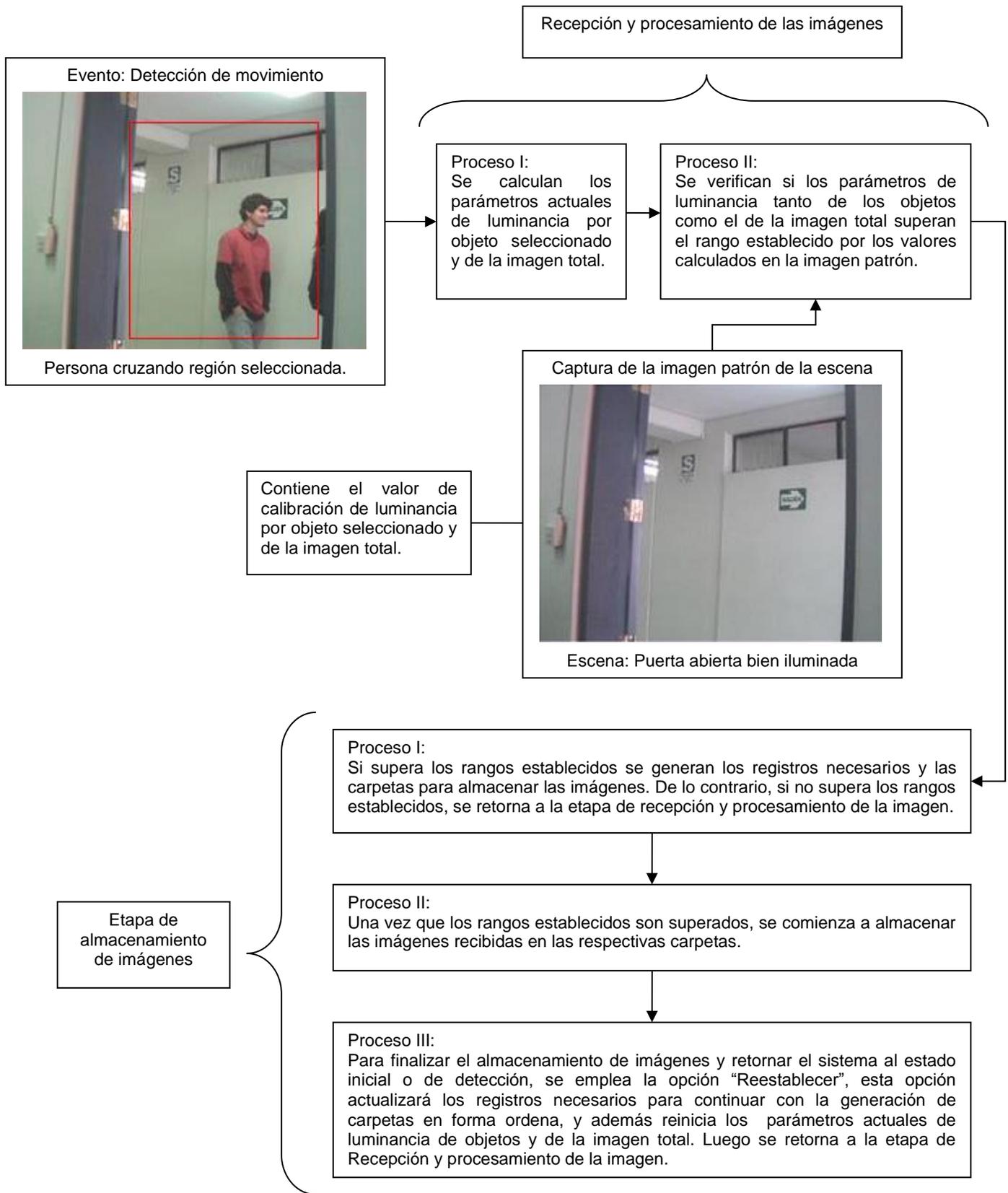


Figura 3.7 Diagrama de bloques del algoritmo para almacenar imágenes.

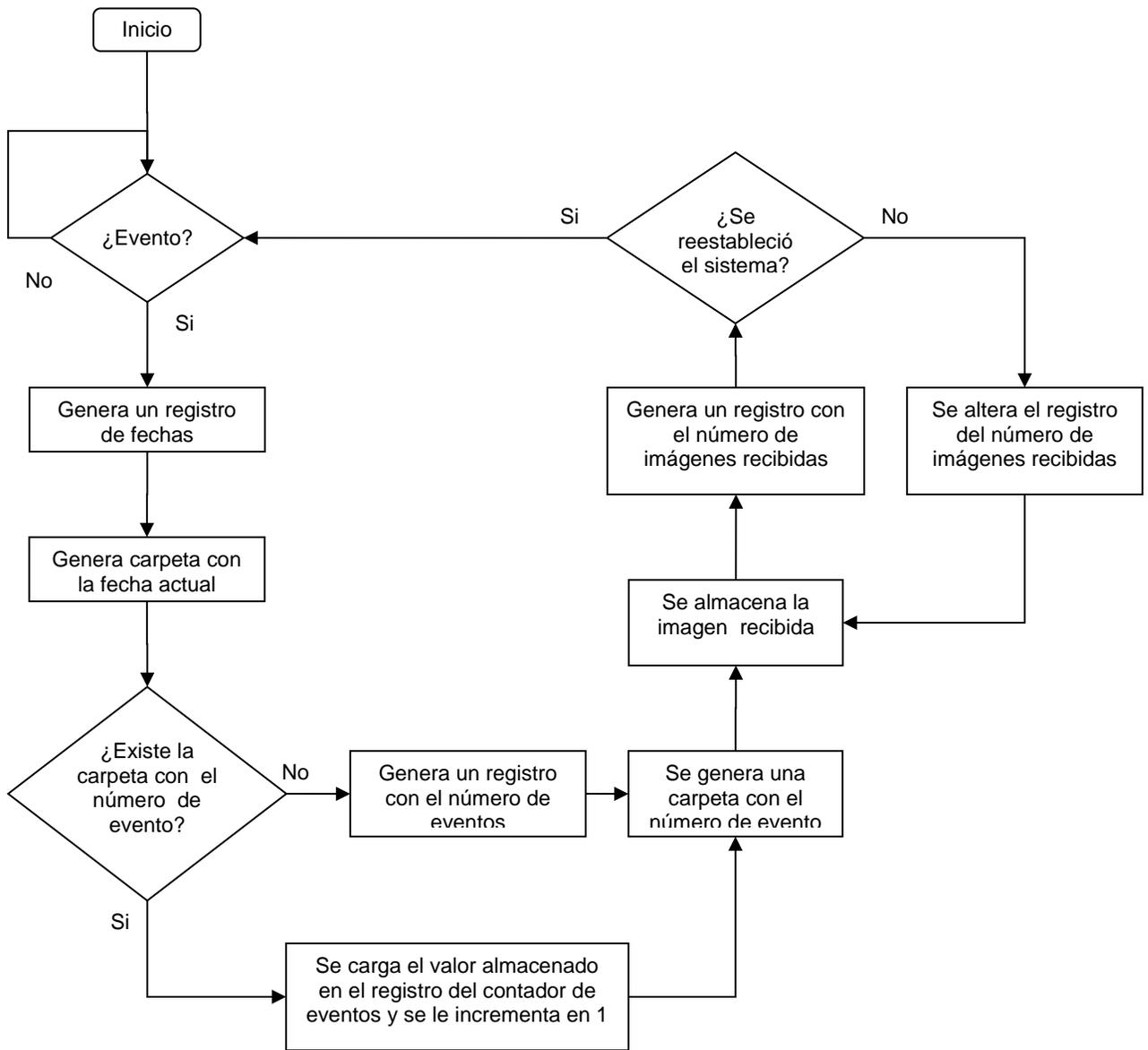


Figura 3.8 Diagrama de flujo del algoritmo para almacenar imágenes.

Este algoritmo sigue los mismos pasos que el algoritmo para almacenar vídeo excepto en la parte en la que se debería generar el archivo de vídeo, en lugar de eso se captura una imagen en formato JPEG y también se genera un archivo de registro en el que se guarda el número de imágenes capturadas. Luego, si aún no se ha reestablecido el sistema entonces el valor del registro que guarda el número de imágenes capturadas se incrementa en una unidad y se procede a capturar la imagen correspondiente. Este bucle de captura de imágenes seguirá realizándose mientras el usuario no presione el botón de *Reestablecer Sistema*, el cual detiene la captura de imágenes y actualiza las variables para poder continuar nuevamente con el testeo constante de eventos, y así iniciar una nueva secuencia de capturas al momento de ocurrir otro evento.

3.5 Algoritmo de reproducción de imágenes

El algoritmo de reproducción de imágenes se ha desarrollado con la intención de visualizar las imágenes capturadas durante un evento ocurrido, de tal forma que la visualización de estas imágenes de la impresión de ser un vídeo continuo. Esta reproducción de imágenes es una alternativa diferente a la reproducción de vídeo, ya que la reproducción de vídeo sólo permite visualizar la grabación del evento. Sin embargo, este modo de reproducción además de permitir la visualización del evento ocurrido también permite que el usuario disponga de otras opciones, como realizar el análisis imagen por imagen de la grabación, logrando de esta manera obtener detalles adicionales como la velocidad promedio, ubicación y dirección de un objeto en movimiento dentro de la imagen, y además, también se puede realizar un zoom digital de la imagen seleccionada en el cuadro de reproducción.

El algoritmo consiste en analizar una serie de condiciones antes de reproducir las imágenes almacenadas. Primero, se comprueba si existe el archivo de registro de fechas, el cual registra las fechas en las que se han grabado imágenes de algún evento ocurrido. De existir dicho registro, se comprueba si es que están creadas las carpetas correspondientes a las fechas guardadas y si es que existen los archivos de registro con el número de eventos ocurridos en cada fecha. Luego de esto, se verifica la existencia de carpetas de los eventos ocurridos en cada fecha, cada una con su respectivo registro del número de imágenes capturadas que posee. De no cumplirse cualquiera de estas condiciones de existencia, entonces no existiría imagen alguna para reproducir.

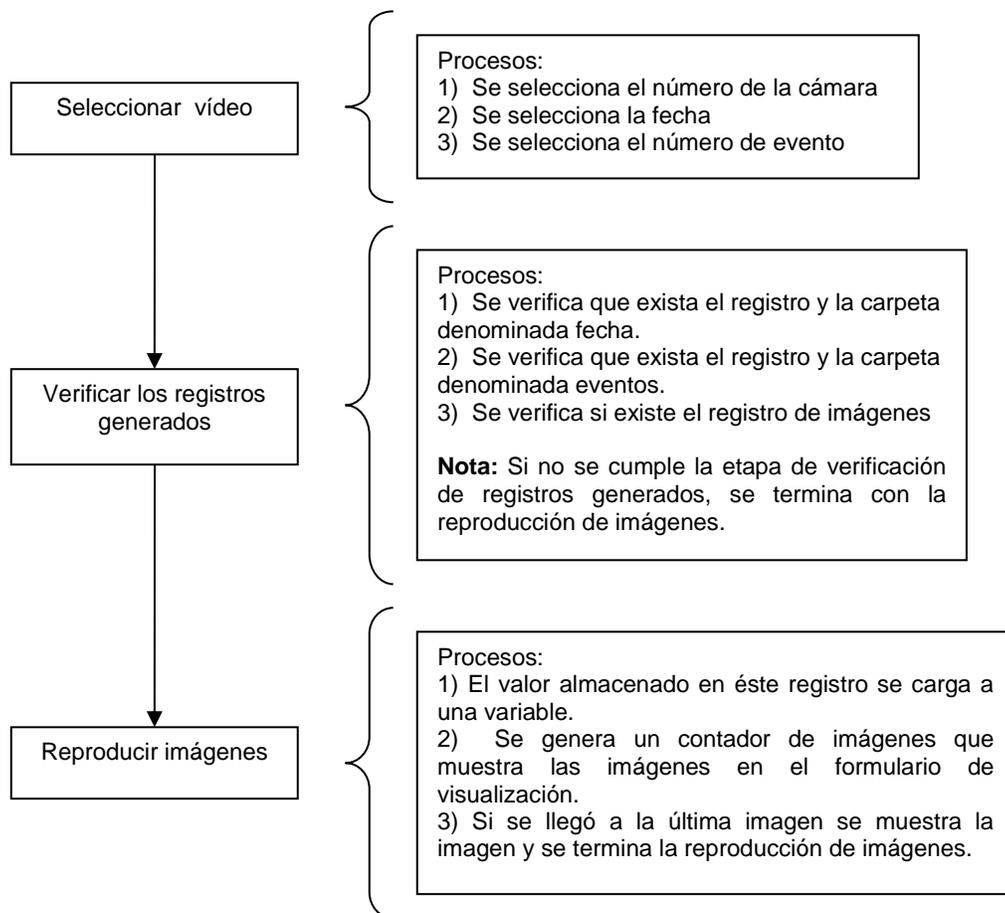


Figura 3.9 Diagrama de bloques del algoritmo para reproducir imágenes.

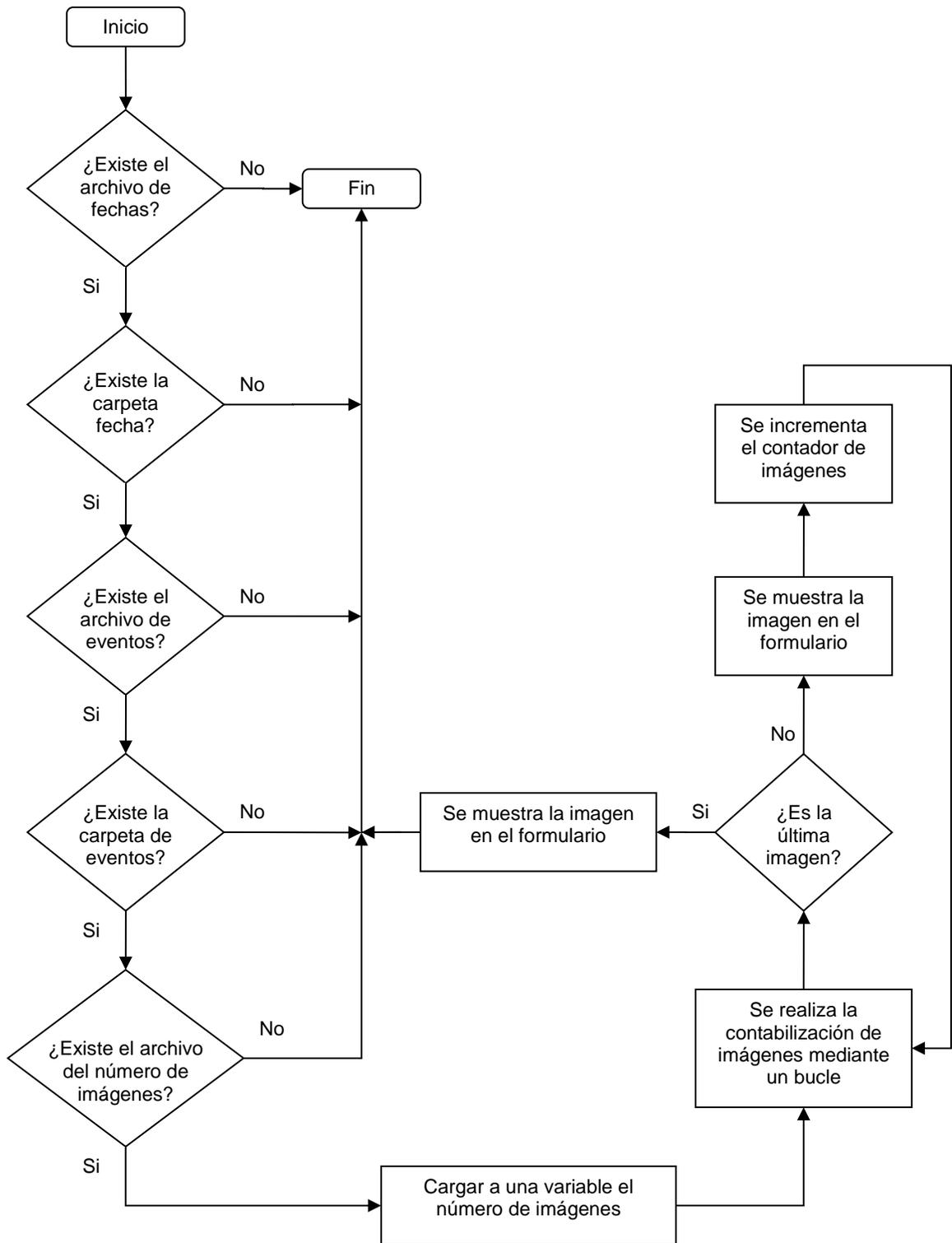


Figura 3.10 Diagrama de flujo del algoritmo para reproducir imágenes.

Una vez que se ha comprobado la existencia de las carpetas y archivos de registro correspondientes a los eventos ocurridos anteriormente, el algoritmo carga a una variable la cantidad de imágenes del evento que el usuario ha seleccionado para visualizar. Y luego de conocer este número de imágenes el algoritmo muestra en un cuadro de visualización la primera imagen almacenada, para luego incrementar el contador de imágenes en una unidad y así realizar consecutivamente la visualización de imágenes hasta que se haya mostrado la última imagen correspondiente al evento seleccionado.

CAPÍTULO 4

DISEÑO E IMPLEMENTACIÓN DEL SISTEMA

El diseño del sistema se ha desarrollado específicamente para conseguir los objetivos trazados al inicio del proyecto de tesis, los cuales plantean diseñar una interfaz que permita visualizar múltiples cámaras IP en una red local y que pueda analizar las imágenes recibidas de estas cámaras para así poder detectar automáticamente eventos remotos.

Por otra parte, la implementación del sistema consiste en implementar uno a uno los algoritmos desarrollados para la detección de eventos. Estos algoritmos estarán ubicados en diversas partes de la interfaz de usuario desarrollada y se ejecutarán cuando se cumplan ciertas condiciones preestablecidas que el usuario debe de llevar a cabo. A lo largo de este capítulo se detallarán todas las partes de la interfaz de usuario en las cuales se muestra la implementación de los algoritmos de detección.

4.1 Esquema del sistema

El sistema consiste de un conjunto de cámaras IP, las cuales pertenecen a una red local o en su defecto se conectan a través de Internet utilizando una IP pública fija. Estas cámaras actúan como servidores, ya que están a la espera de que algún programa cliente se conecte a ellas para comenzar a enviar la información de las imágenes que van capturando.

En este proyecto, se ha desarrollado un software cliente al cual se deben ingresar manualmente las direcciones IP de las cámaras, para que éste comience a visualizarlas en una interfaz propia, a través de la cual se acceden a múltiples opciones de configuración que permiten analizar las imágenes recibidas. Durante este análisis de las imágenes es que se ejecutan los algoritmos de detección de eventos y ante la ocurrencia de alguno se disparan las alarmas, las cuales pueden ser sonoras mediante un motor de voz o alarmas visuales remarcando en rojo a la cámara que ha detectado el evento. Además, también se almacenan los datos del evento ocurrido en una tabla de registro, la cual permite migrar sus datos a un archivo de Excel.

En la figura 4.1, mostrada en la página siguiente, se observa el diagrama de bloques del sistema desarrollado, el cual está dividido en dos partes, transmisión y recepción. La parte de transmisión está formada por la cámara IP, la cual captura una cierta cantidad de imágenes por segundo y las transmite a través de su interfaz Ethernet que le permite conectarse con otros dispositivos en red. Por otra parte, la recepción está formada por la computadora que tiene instalado el software de administración para manipular las imágenes recibidas por las cámaras IP conectadas, las cuales serán monitoreadas y procesadas para poder detectar eventos automáticamente y generar alarmas y reportes.

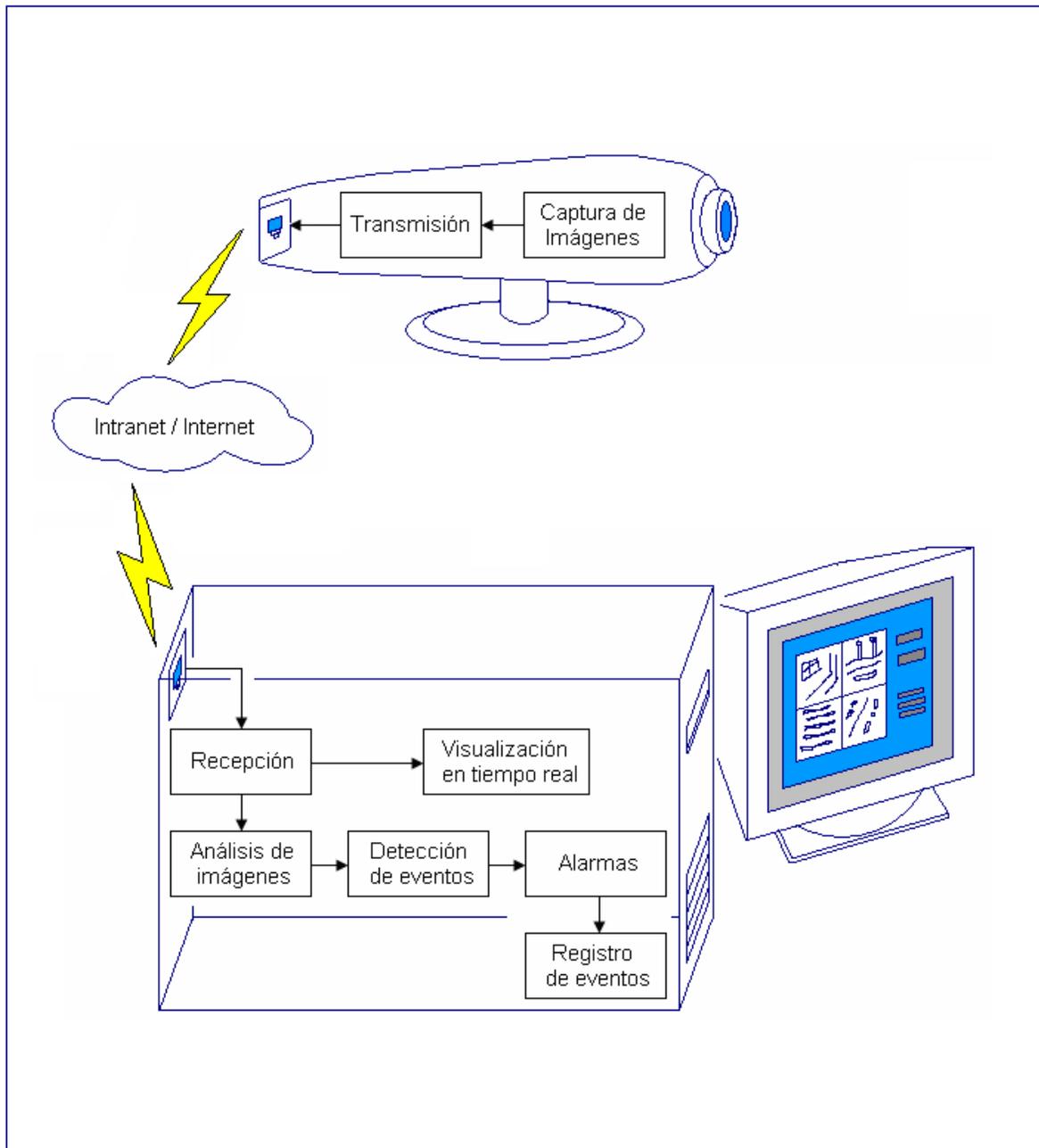


Figura 4.1 Esquema del sistema desarrollado.

4.2 Interfaz visual principal

La interfaz visual principal desarrollada para el proyecto consta de cuatro cuadros de visualización ubicados a la izquierda de la ventana. Estos cuadros permiten conectarse como máximo a cuatro cámaras IP en simultáneo. Se determinó que sólo debería ubicarse 4 cuadros, ya que el procesamiento para analizar las imágenes recibidas es bastante largo y colocar más de cuatro cuadros de visualización afectaría en la respuesta del sistema.

En la figura 4.2, los botones de las opciones para manipular la interfaz visual están ubicados en la parte inferior derecha de la ventana, y debajo de ellos se encuentra el botón para salir del programa. Por otra parte, en la zona superior derecha se encuentran visibles los modelos de cámaras IP con los que el sistema desarrollado es compatible.

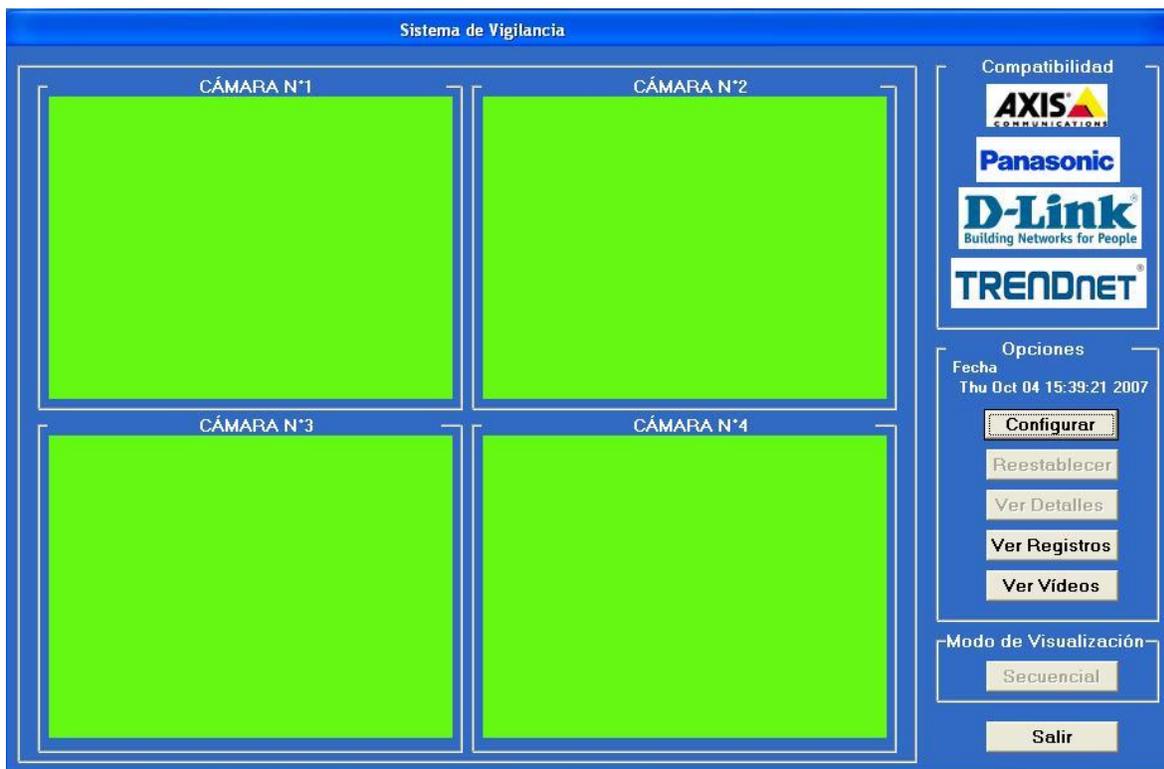


Figura 4.2 Ventana principal del programa desarrollado.

4.2.1 Opciones de la ventana principal

Dentro del cuadro de opciones de la ventana principal, el primer botón habilitado es el botón *Configurar* y sirve para configurar las cámaras IP. Una vez presionado este botón, aparece una ventana de configuración en la cual se debe seleccionar uno de los cuatro cuadros de la ventana principal para recibir en él las imágenes de la cámara que se quiere conectar. Luego, se debe ingresar la dirección IP de la cámara a monitorear y también se puede ingresar opcionalmente el modelo de la cámara IP y la ubicación de la misma.

Luego de llenar los datos de la cámara IP se presiona el botón *Conectar* e inmediatamente en el cuadro de la izquierda de la ventana de configuración se comenzarán a transmitir las imágenes recibidas de la cámara conectada, como se observa en la figura 4.3. De no efectuarse la visualización de las imágenes se deberá a que no se ha realizado la conexión, probablemente a causa de una dirección IP errónea o a fallas de la cámara.



Figura 4.3 Ventana de configuración de cámara IP. Esta ventana de configuración también posee otros botones como *Desconectar*, el cual desconecta la cámara IP visualizada. También

existe el botón *Limpiar* que vacía el contenido de todos los datos ingresados de la cámara; el botón *Calibrar* que sirve para la detección de eventos, lo que será explicado posteriormente en este capítulo; el botón *Configurar Cámara* que sirve para realizar una configuración avanzada de la cámara, ya que muestra a través de un browser todas las opciones de la cámara IP como la velocidad de envío de imágenes, la tasa de compresión, entre muchas otras opciones; y por último, se encuentra el botón *Regresar* que sirve para volver a la ventana principal del programa.

En la figura siguiente se muestra la ventana principal del programa en la cual se ha realizado la conexión de una cámara IP en el cuadro de la cámara N°1. En estas circunstancias sólo se estaría cumpliendo la función de monitoreo, más no así la de detección de eventos, ya que sólo se habría realizado la conexión a la cámara IP y aún no se habría calibrado los objetos a detectar en dicha cámara.

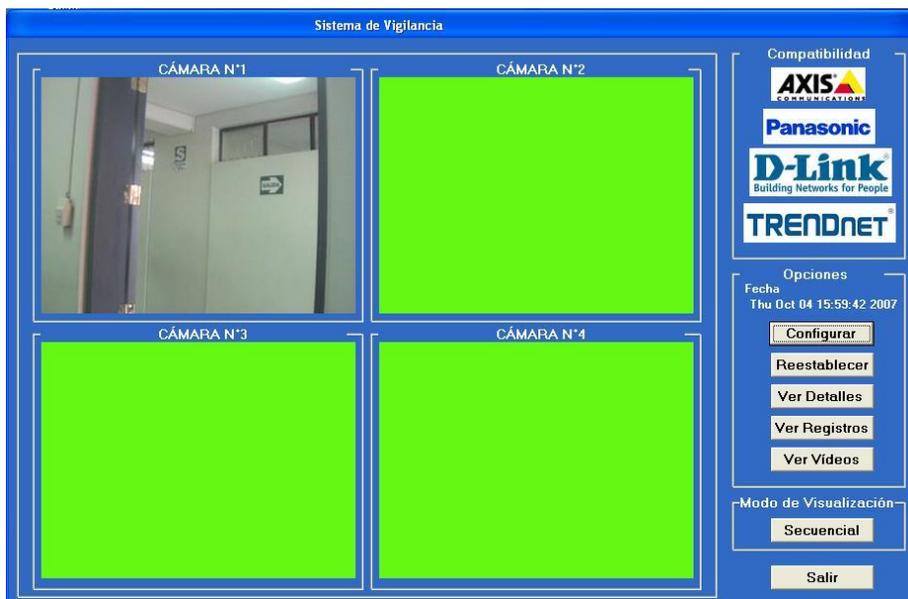


Figura 4.4 Monitorización de una cámara IP conectada.

El siguiente botón de la ventana principal a analizar es el botón *Reestablecer*, cuya accesibilidad inicialmente esta deshabilitada, ya que solamente se habilita luego de que alguna cámara esté realizando el proceso de detección de eventos. Esto se debe a que la función del botón *Reestablecer*, como su nombre lo indica, es precisamente la de reestablecer los valores iniciales de las variables que se usan durante el proceso de detección luego de que haya ocurrido alguno.

Luego de una detección, las alarmas se encienden y la cámara se pone en rojo, comenzando con la grabación continua de vídeo e inhabilitando la detección de otros eventos. Es por esa razón que el botón *Reestablecer* es muy necesario, ya que le permite al usuario seguir detectando nuevos eventos y terminar con la grabación del último evento detectado. De esta forma, el sistema vuelve a un estado tal como si los objetos a detectar dentro de la cámara estuvieran recién calibrados.

Otro botón de la ventana principal que inicialmente está deshabilitado es el botón *Ver Detalles*, esto debido a que su función principal es la de mostrar en un nuevo formulario (ver figura 4.5) toda la información registrada de las cámaras IP conectadas, por lo que permanece inhabilitado hasta que se ingrese información de alguna cámara IP. La razón de ser de este botón es la de permitir al usuario visualizar la información de todas sus cámaras más ordenadamente, ya que muestra tanto el modelo, la dirección IP y la ubicación de las cámaras conectadas. Además, también permite al usuario ingresar un pequeño gráfico en el que se puede visualizar la distribución de las cámaras IP, para así poder realizar un mejor control de

las mismas y también analizar ubicaciones estratégicas, posibles cambios de localización o un incremento de cámaras para zonas con poca seguridad.

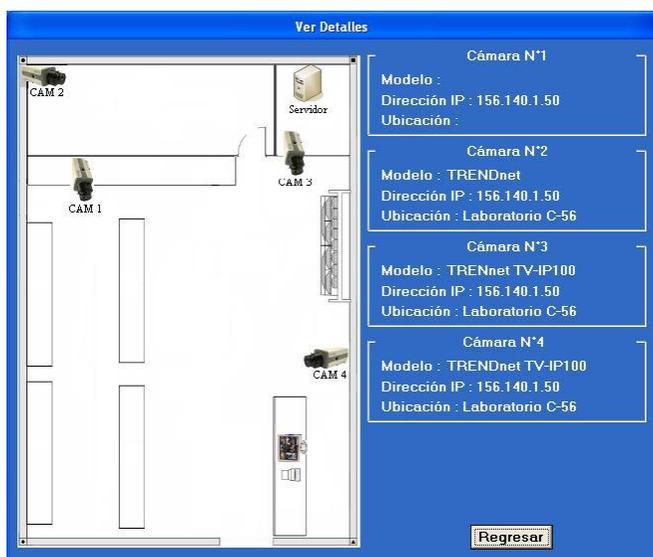


Figura 4.5 Descripción de las cámaras IP conectadas.

El siguiente botón en la ventana principal es el botón *Ver Registros*, el cual sirve para visualizar la información de los eventos registrados desde la ejecución del programa. Este nuevo formulario de registro muestra ordenadamente el número de cámara en la que ocurrió cada evento detectado, el objeto que dio origen al evento que pudo ser un objeto de la escena o la cámara misma, el tipo de evento detectado tal como un movimiento o un cambio de iluminación del ambiente de la cámara, la fecha y la hora exacta del evento.

Además, también dentro de este formulario de registro (ver figura 4.6) existe un botón para generar un archivo de Excel con el cual se puede migrar la información de todos los eventos detectados. Este archivo generado puede ser utilizado para trasladar la información fácilmente

entre computadoras o para imprimir dicha información, también sería posible migrar esta información a una base de datos para almacenarla.

N°Cámara	Objeto	Evento	Fecha	Hora
CAM 1	Objeto1	Movimiento	4/9/2007	15:50:42
CAM 1	Objeto1	Movimiento	4/9/2007	15:54:45
CAM 1	Objeto1	Movimiento	4/9/2007	15:55:50
CAM 1	Objeto1	Movimiento	4/9/2007	15:59:54
CAM 1	Objeto1	Movimiento	4/9/2007	16:0:9
CAM 1	Camara1	iluminacion	4/9/2007	16:04:19

Figura 4.6 Registro de eventos detectados.

El último botón del cuadro de opciones en la ventana principal es el botón *Ver Vídeos*, el cual muestra los vídeos almacenados. Existe también en la ventana principal un botón que permite cambiar el modo de visualización múltiple de 4 cámaras por un modo de visualización secuencial (ver figura 4.7) en el que se visualiza cámara por cámara.

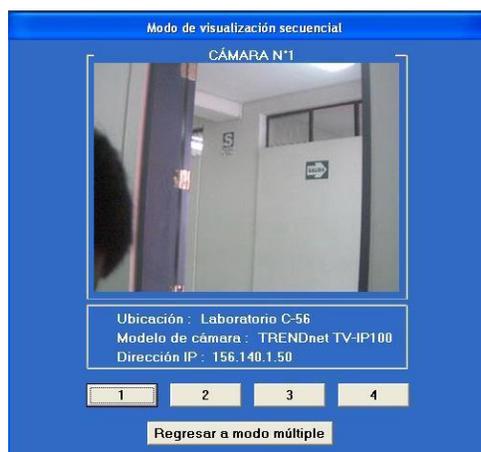


Figura 4.7 Modo de visualización secuencial.

4.3 Interfaz de calibración de objetos

Para acceder a la interfaz de calibración de objetos es necesario primero conectar la cámara IP en la ventana de configuración. Luego en esta misma ventana se presiona el botón *Calibrar* con el que la ventana de configuración se extiende hacia abajo, mostrando así las opciones para seleccionar los objetos que se desean analizar en la imagen capturada.

Primero, es necesario indicar el número de objetos a seleccionar, luego se sombrea una región en la imagen capturada haciendo clic en ella y arrastrando sin soltar el *mouse* para formar una región rectangular con un borde rojo (ver figura 4.8) y después se ingresa el nombre que se quiere dar al objeto del que se acaba de seleccionar su área.

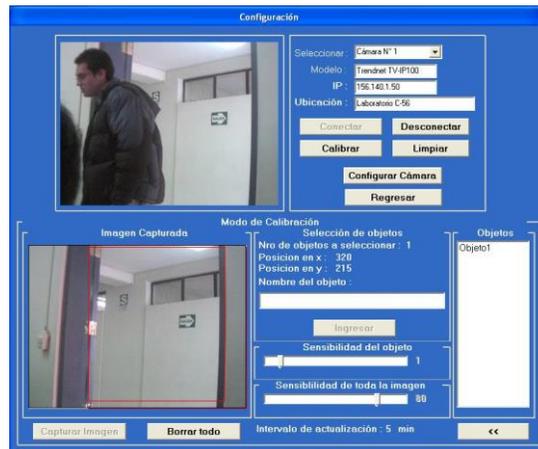


Figura 4.8 Ventana de calibración de objetos.

Existen también 2 *sliders* los cuales muestran los valores de sensibilidad del objeto y de toda la imagen. Estos valores de sensibilidad pertenecen al algoritmo de detección de movimiento desarrollado, el cual fue mostrado en el capítulo anterior, y son estos valores los que corresponden al primer y segundo umbral del algoritmo de detección. Estos valores están

calculados para una imagen de contraste promedio, pero se pueden modificar a criterio del usuario, ya que para imágenes distintas podría generar falsas detecciones.

Luego de ajustar los valores de sensibilidad se presiona el botón *Ingresar* y automáticamente el objeto se registra en el cuadro de la derecha, lo que indica que una vez que se regrese a la ventana principal se va a ejecutar el algoritmo de detección de movimiento. Y cuando el algoritmo detecte una variación significativa en el área del objeto, se pondrá en rojo el borde de la cámara (ver figura 4.9) y se ejecutarán las alarmas.

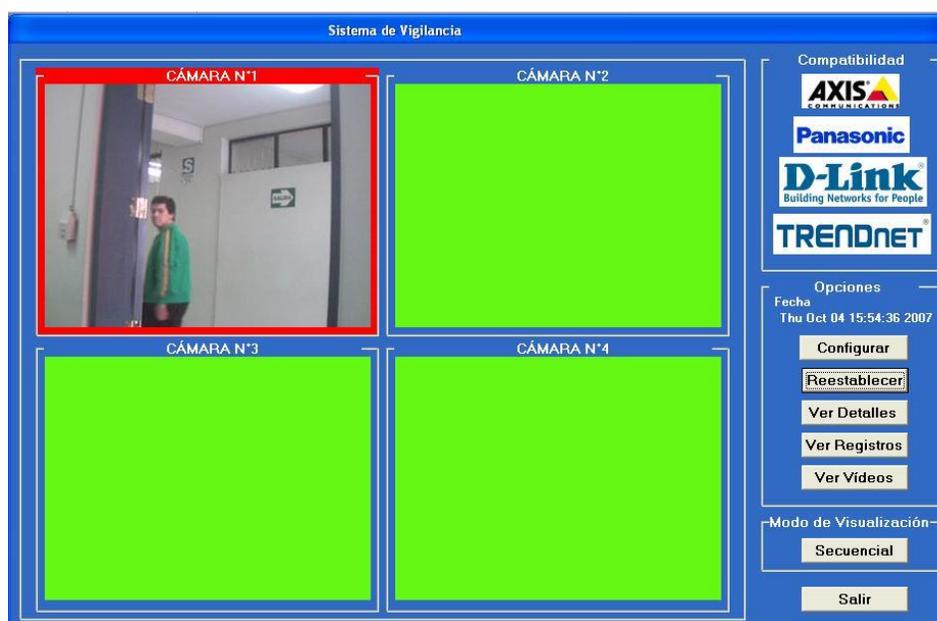


Figura 4.9 Detección de movimiento en la cámara 1.

Después de que se ha realizado la primera calibración de un objeto no sólo se ejecutará el algoritmo de detección de movimiento, sino que también automáticamente se dará inicio al algoritmo de detección de cambio de iluminación. Al igual que el algoritmo de detección, el algoritmo de cambio de iluminación también posee 2 umbrales de sensibilidad, pero a diferencia del otro algoritmo, éste no permite ajustar estos valores de sensibilidad, ya que han sido calculados para detectar pequeñas variaciones de luminancia en un gran sector de la

imagen. En el caso de que no se esté ejecutando el algoritmo de detección de movimiento para ningún objeto, entonces tampoco se estaría ejecutando el algoritmo de detección de cambio de iluminación.

El cambio de iluminación puede ser detectado debido a que cuando la iluminación de un ambiente varía, los píxeles de las imágenes capturadas también varían aumentando o disminuyendo su valor de luminancia (ver figura 4.10). El análisis de estos píxeles modificados permite la detección del cambio de iluminación. El gráfico inferior muestra la imagen capturada de una cámara antes y después de haber detectado una alteración.

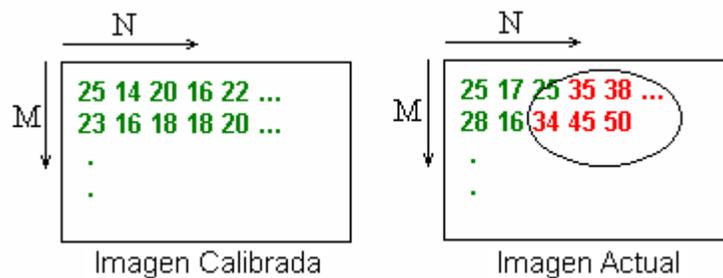


Figura 4.10 Variación en la luminancia de los píxeles.

Cuando ocurre de cambio de iluminación, el cuadro de la cámara en la que ocurrió la detección se resalta en color amarillo (ver figura 4.11) para indicar que ha ocurrido dicho evento. Esto permite al usuario diferenciar fácilmente entre los distintos tipos de eventos.

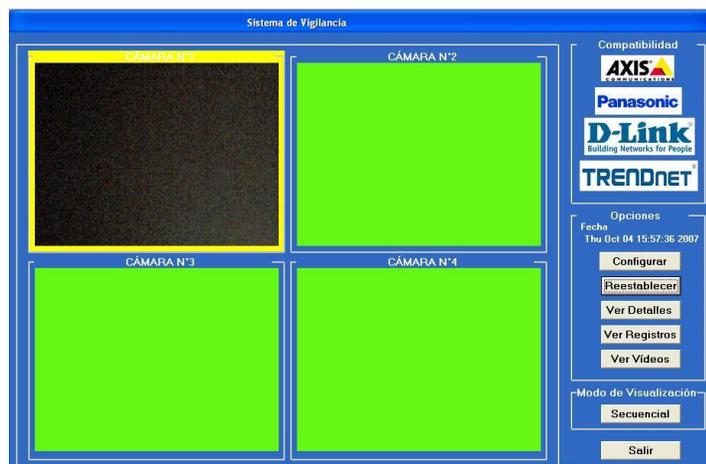


Figura 4.11 Detección de cambio de iluminación en la cámara 1

4.4 Interfaz de vídeo y análisis de imágenes

El acceso a la interfaz de vídeo se hace en la ventana principal a través del botón *Ver Vídeos*, el cual muestra una ventana en la cual están registrados todos los videos grabados durante la detección de eventos. Estos vídeos pueden haber sido almacenados en fechas anteriores y aun así se pueden visualizar ya que el algoritmo utilizado para su reproducción no los elimina. Cabe mencionar, que para realizar el registro de vídeos se ha ejecutado el algoritmo de almacenamiento de vídeo mostrado en el capítulo anterior, el cual también muestra el algoritmo para reproducir los vídeos almacenados.

Para realizar la reproducción de un vídeo es necesario primero seleccionar el número de cámara en la que se va a buscar el vídeo, luego se selecciona la fecha en que se grabó el mismo, después de esto se elige uno de los vídeos que aparecen listados en el cuadro de la parte inferior y finalmente se presiona el botón *Play* para visualizar el vídeo. Por otra parte, esta ventana de visualización permite realizar un análisis más detallado del vídeo seleccionado, para esto se presiona el botón *Analizar Vídeo*, el cual expande las opciones de la ventana a una nueva interfaz, como se ve en la figura 4.12.



Figura 4.12 Ventana inicial del analizador de vídeos.

Las nuevas opciones pertenecen al analizador de vídeo, el cual utiliza el algoritmo de almacenamiento de imágenes desarrollado en el capítulo anterior para guardar imágenes consecutivas en forma de vídeo, las cuales son analizadas con el objetivo de determinar ciertas características de los objetos ubicados dentro de la imagen. Estas características pueden ser la posición, velocidad o dirección del objeto en movimiento dentro de la imagen y para obtenerlas es necesario analizar las imágenes almacenadas.

Cabe mencionar que, el software considera que sólo existe un objeto en movimiento en toda la imagen, de manera que si existieran más objetos en movimiento los parámetros calculados como la posición, la velocidad, etc. estarían en base a suponer que todos estos objetos forman parte de uno solo. El primer paso que realiza el software, para el cálculo de los parámetros mencionados anteriormente, es identificar al objeto en movimiento dentro de la imagen, para ello se realiza el proceso de umbralización de la imagen, con el cual se extraen todos los píxeles pertenecientes al objeto en movimiento.

4.4.1 Proceso de umbralización de la imagen

La umbralización de una imagen, nos permite dado un umbral definido, separar lo que es fondo de los que es objeto, siempre y cuando el fondo y el objeto tengan sus niveles de luminancia agrupados en dos modos dominantes. En este caso, para identificar al objeto en movimiento se contrastarán dos imágenes de la misma escena capturadas en diferentes instantes de tiempo, de manera que en la primera imagen no aparece el objeto en movimiento, por lo que ésta servirá como imagen patrón, esta imagen es capturada justo antes de que el objeto ingrese a la escena, mientras que, la segunda imagen a considerar es la imagen actual, la cual es capturada en tiempo real.

Para realizar el proceso, primero se obtienen los valores de luminancia de todos los píxeles de ambas imágenes, para luego calcular el porcentaje de error que ha tenido cada píxel. Este porcentaje de error se compara con un umbral de error predefinido y se determina que aquellos píxeles que superen el umbral pertenecen al objeto extraño (en otras palabras, aquellos píxeles que hayan modificado significativamente su valor de luminancia), por lo que se les tomará en cuenta como píxeles del objeto y a los que no superen el umbral serán considerados píxeles del fondo. En la figura 4.13, se puede observar a las dos imágenes que se van a contrastar para identificar los píxeles del objeto.

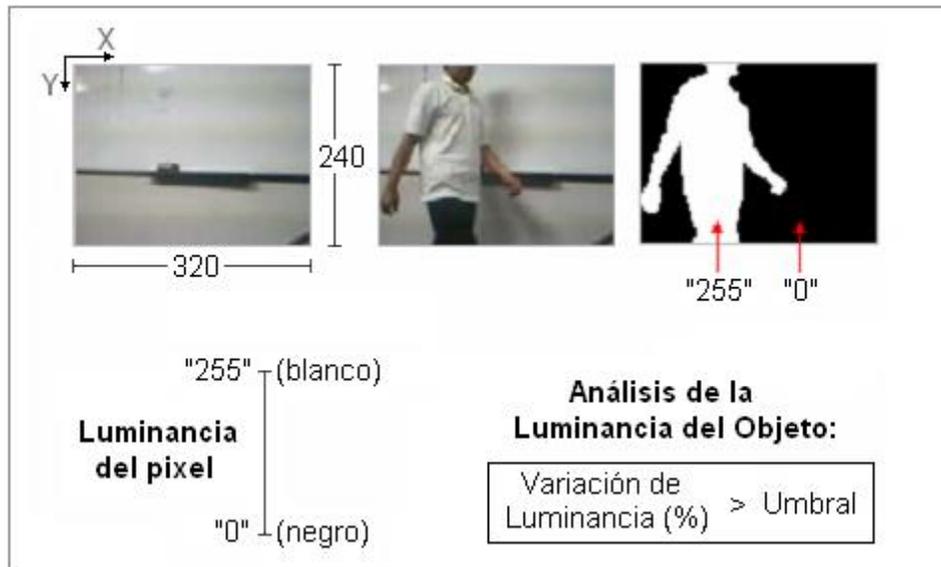


Figura 4.13 Proceso de binarización de la imagen.

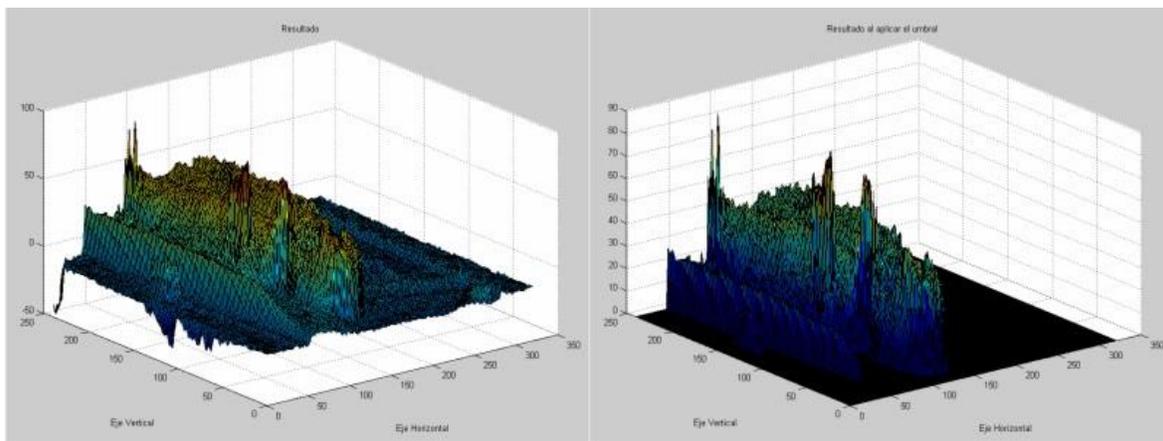


Figura 4.14 Variación de luminancia antes y después de la umbralización.

En la figura 4.14, a la izquierda se puede observar el porcentaje de variación de luminancia de cada píxel en una superficie de 320 x 240 píxeles con una variación de 0 a 100 %. Normalmente, si en las dos imágenes contrastadas para obtener este porcentaje de variación de luminancia no hubiera un objeto extraño, la gráfica obtenida hubiera sido una superficie plana

bordeando los valores de 0 %. Sin embargo, como se ve hay una zona de gran variación de luminancia con forma de una montaña. Por otra parte, a la derecha de la figura se muestra el porcentaje de variación de luminancia de cada píxel luego de haber aplicado el proceso de umbralización, en otras palabras, los píxeles que se observan son los que pertenecen al objeto y los del fondo tienen un valor de 0%.

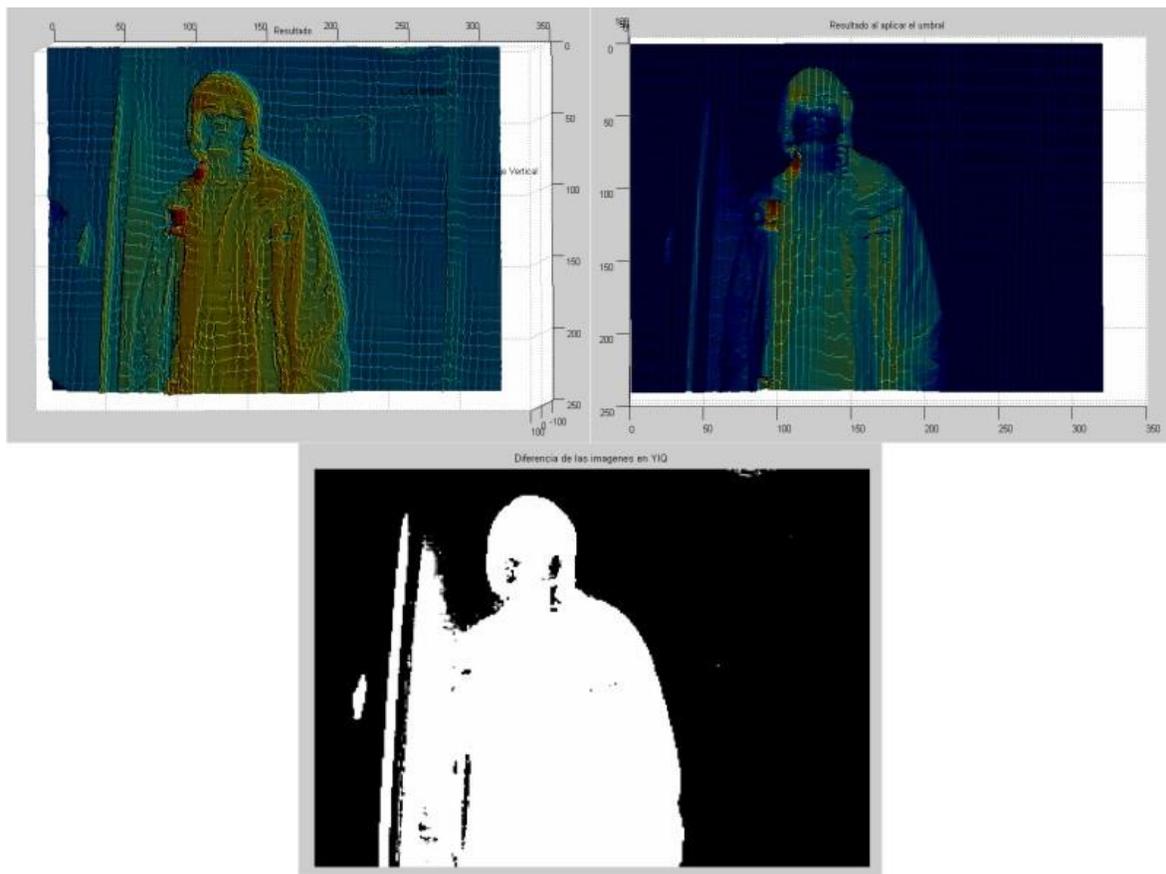


Figura 4.15 Imágenes antes y después de la umbralización y la imagen binarizada.

En la parte superior de la figura 4.15, se observa en vista de planta, el porcentaje de variación de luminancia de la imagen antes y después de la umbralización. Al estar en vista de planta, nos permite visualizar como este porcentaje grafica claramente al objeto extraño en la escena.

Usualmente luego de realizar el proceso de umbralización se realiza la binarización de la imagen, donde al área deseada (el objeto) se pone en blanco y al área restante (el fondo) de color negro. Para esto, a aquellos píxeles que superen el umbral predefinido se les asigna un valor de luminancia de ‘255’ y a los demás píxeles restantes un valor de luminancia de ‘0’. Es así que luego de realizar el proceso de binarización se obtiene una imagen negra en la que hay una región blanca perteneciente al objeto extraño (ver parte inferior de la figura 4.15).

4.4.2 Cálculo de posición, velocidad y dirección del objeto

Para realizar el cálculo de todos estos parámetros pertenecientes al “*Analizador de imágenes capturada*” es necesario primero obtener el centro de masa del objeto extraño a la escena original. Para ello, se hace uso de la imagen binarizada obtenida y se procede a calcular el centro de masa del objeto extraño, para lo que se calcula el promedio equivalente a las coordenadas (x,y) de todos los píxeles con luminancia ‘255’, los cuales pertenecen a la región del objeto en movimiento. La fórmula resultante para obtener estas coordenadas del centro de masa (xc,yc) del objeto extraño se muestra a continuación, donde I_B representa a la imagen binarizada:

$$x_c = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} x \times I_B(x, y)}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I_B(x, y)}; y_c = \frac{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} y \times I_B(x, y)}{\sum_{x=0}^{M-1} \sum_{y=0}^{N-1} I_B(x, y)} \quad (4.1)$$

En la figura 4.14 se muestra una secuencia de 5 imágenes en las que un individuo atraviesa la región capturada por la cámara. Se puede observar que en la primera imagen aún no aparece el individuo, ya que es capturada exactamente antes de que ocurra la detección y es precisamente debido a que no tiene ninguna alteración por lo que es utilizada como imagen patrón. Las imágenes capturadas posteriores a la imagen patrón son binarizadas con referencia a ésta y es

así que se obtiene una imagen binarizada por cada imagen capturada. De esta forma, se puede obtener el centro de masa del objeto extraño en todo momento y con él realizar los cálculos de los parámetros deseados.

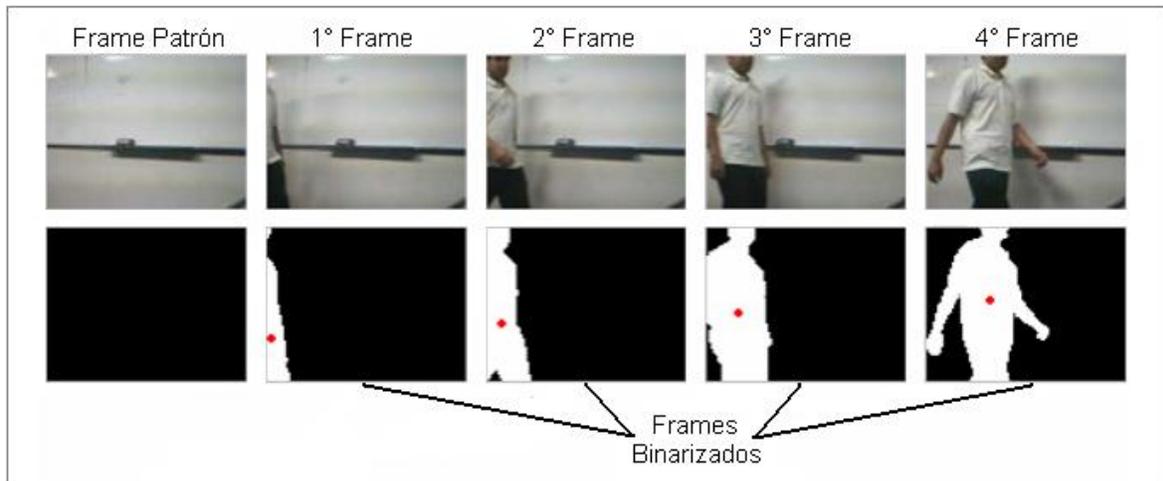


Figura 4.16 Variación el centro de masa (punto rojo).

Para calcular el primer parámetro deseado que es la velocidad instantánea del objeto, se necesita la distancia euclidiana entre 2 puntos (centros de masa) y el tiempo transcurrido que vendría a ser el periodo de captura de cada imagen, entonces ya se podría conocer la velocidad del objeto (en píxeles/seg.). Luego para obtener esta velocidad en metros/seg. se calcula una equivalencia de píxeles a metros, la cual es obtenida a partir la distancia que cubre la cámara (320 píxeles), la cual depende de la distancia de referencia que hay de la cámara al objeto y del ángulo de apertura de la cámara.

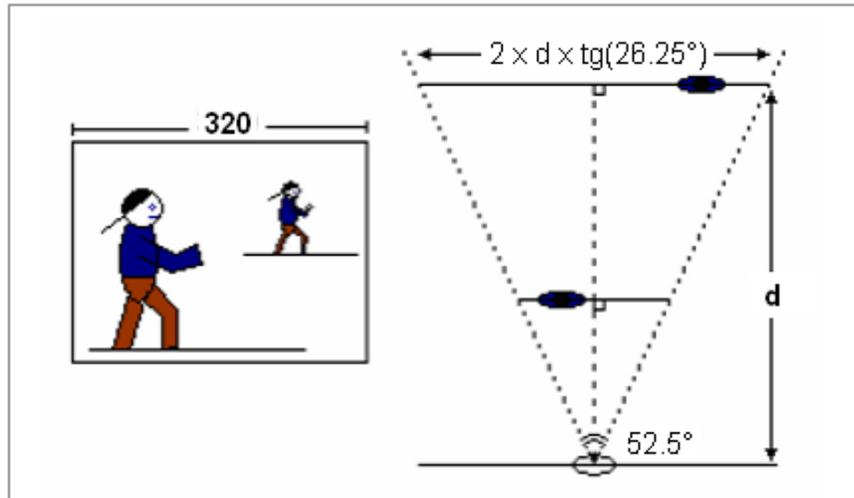


Figura 4.17 Espacio que cubre la cámara en función de la distancia de referencia.

Como se muestra en la figura de arriba, los 320 píxeles de área que cubre la cámara son equivalentes (en metros/seg.) a una cantidad que está en función de la distancia de referencia entre la cámara y la trayectoria del objeto, y también del ángulo de 52.5° de apertura de la cámara. La equivalencia obtenida se muestra en la ecuación (4.3), y con esta equivalencia se logró obtener la fórmula para calcular la velocidad del objeto (en metros/seg.), la cual es mostrada en la ecuación (4.4).

$$d = \sqrt{(x_{c1} - x_{c0})^2 + (y_{c1} - y_{c0})^2} \quad , \quad T = 0.2 \text{seg} \quad (4.2)$$

$$320 \text{ píxeles} \equiv 2 \times d_{ref} \times \text{tg}(26.25^\circ) \text{ metros} \quad (4.3)$$

$$V = \frac{d}{T} (\text{píx} / \text{seg}) = \frac{d}{T} \times \frac{2 \times d_{ref} \times \text{tg}(26.25^\circ)}{320} (\text{metros} / \text{seg}) \quad (4.4)$$

Luego para obtener la dirección en la que se desplaza el objeto a través de la imagen, se calcula el ángulo hacia donde se mueve el objeto en cada instante de captura, y comparando este ángulo obtenido con la tabla 4.1 se puede definir el cuadrante en el que se encuentra el objeto y así determinar la dirección que éste lleva a cada instante dentro de la imagen capturada.

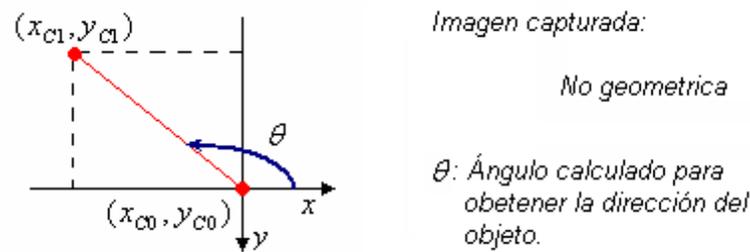


Figura 4.18 Ángulo calculado utilizando la posición actual y anterior del centro de masa.

$$\theta = \text{arctg} \left(\frac{y_{c0} - y_{c1}}{x_{c1} - x_{c0}} \right) \quad (4.5)$$

Cabe resaltar, que la formula para calcular el ángulo con el que se desplaza el objeto, no se da simplemente como el arco tangente de la diferencia de ordenadas entre la diferencia de abscisas, sino que la diferencia de ordenadas esta invertida, ya que por ser una imagen no geométrica, la ordenada crece hacia abajo y no hacia arriba.

Tabla 4.1 Dirección del objeto según el cuadrante del ángulo obtenido.

Ángulo	Dirección
$0 < \theta < 30^\circ ; 330^\circ < \theta < 360^\circ$	derecha
$30^\circ < \theta < 60^\circ$	diagonal superior derecha

$60^\circ < \theta < 120^\circ$	Arriba
$120^\circ < \theta < 150^\circ$	diagonal superior izquierda
$150^\circ < \theta < 210^\circ$	Izquierda
$210^\circ < \theta < 240^\circ$	diagonal inferior izquierda
$240^\circ < \theta < 300^\circ$	Abajo
$300^\circ < \theta < 330^\circ$	diagonal inferior derecha

Finalmente, en la ventana para analizar vídeo se muestran los parámetros calculados (ver figura 4.16), primero se muestra la posición, velocidad y dirección del objeto extraño a cada instante y luego, en otro cuadro, muestra la velocidad inicial, final y promedio de todo el trayecto del objeto.



Figura 4.19 Ventana final del analizador de vídeos.

4.5 Interfaz de registro de eventos

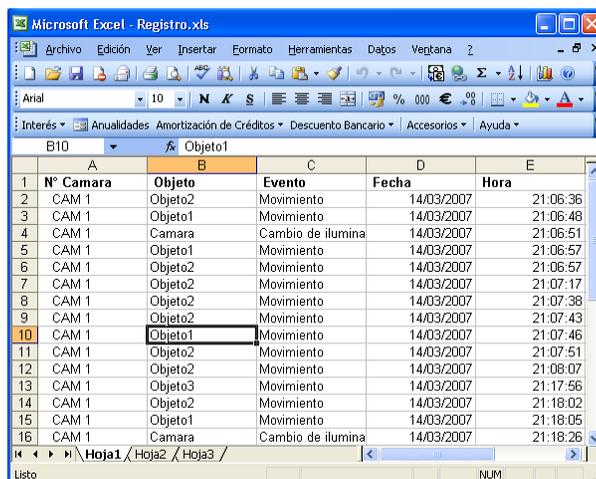
El programa posee un formulario para ver los eventos registrados, el cual aparece al presionar el botón *Ver Registros* ubicado en la ventana principal. Este formulario posee una tabla que muestra los eventos detectados sólo durante el tiempo de ejecución del programa (ver figura 4.17), y entre las características que se muestran está el número de cámara en la que ocurrió el evento; el objeto que causó el evento que pudo ser inclusive la cámara misma; el tipo de evento detectado, si fue movimiento o cambio de iluminación; y finalmente la fecha y la hora del evento.



Figura 4.20 Ventana de registro de los eventos detectados.

Adicionalmente, el formulario que muestra los eventos registrados posee un botón que permite migrar la información de la tabla a un archivo de Excel (ver figura 4.18), el cual se genera en el *disco D* de la computadora dentro de una carpeta llamada *Registro*.

La opción de poder migrar la información es muy importante, ya que el formulario pierde los eventos registrados al cerrarse el programa y este archivo de Excel le da al usuario la facilidad de poder imprimir la información, trasladarla fácilmente a otra computadora o incluso almacenarla en una base de datos.



	A	B	C	D	E
	N° Camara	Objeto	Evento	Fecha	Hora
1	CAM 1	Objeto2	Movimiento	14/03/2007	21:06:36
2	CAM 1	Objeto1	Movimiento	14/03/2007	21:06:48
3	CAM 1	Camara	Cambio de ilumina	14/03/2007	21:06:51
4	CAM 1	Objeto1	Movimiento	14/03/2007	21:06:57
5	CAM 1	Objeto2	Movimiento	14/03/2007	21:06:57
6	CAM 1	Objeto2	Movimiento	14/03/2007	21:07:17
7	CAM 1	Objeto2	Movimiento	14/03/2007	21:07:38
8	CAM 1	Objeto2	Movimiento	14/03/2007	21:07:43
9	CAM 1	Objeto2	Movimiento	14/03/2007	21:07:46
10	CAM 1	Objeto1	Movimiento	14/03/2007	21:07:51
11	CAM 1	Objeto2	Movimiento	14/03/2007	21:08:07
12	CAM 1	Objeto2	Movimiento	14/03/2007	21:17:56
13	CAM 1	Objeto3	Movimiento	14/03/2007	21:18:02
14	CAM 1	Objeto2	Movimiento	14/03/2007	21:18:05
15	CAM 1	Objeto1	Movimiento	14/03/2007	21:18:26
16	CAM 1	Camara	Cambio de ilumina	14/03/2007	

Figura 4.21 Archivo de registro generado en Excel.

4.6 Interfaz de zoom digital

Para poder realizar un zoom digital a las imágenes de vídeo capturadas fue necesario utilizar el método de interpolación bilineal explicado en el capítulo 1. El mismo que fue implementado para amplificar cualquier imagen al doble de su tamaño original, es así que el zoom implementado varía en múltiplos de dos (2X, 4X hasta 8X).

El acceso al formulario para aplicar zoom digital es a través del botón *Zoom* que se encuentra en el formulario del analizador de vídeo. Al presionar dicho botón aparece el formulario para amplificar imágenes mostrando a la imagen seleccionada con una amplificación del doble de

su tamaño original (ver figura 4.19). Este nuevo formulario permite al usuario seguir amplificando la imagen ya amplificada, para eso simplemente se debe hacer clic con el *mouse* sobre la imagen en el punto que se desea centrar la imagen a amplificar. Luego de esto, el valor de amplificación pasará de 2X a 4X, y así se puede seguir amplificando hasta obtener un valor de amplificación equivalente a 8 veces la imagen original (8X).

Por otra parte, si lo que se quiere es deshacer un nivel de zoom y volver a la imagen anterior, sólo se debe presionar el botón (-), el cual reduce en un nivel de amplificación a la imagen. Adicionalmente, este formulario permite restaurar la imagen amplificada hasta un nivel de zoom de 2X, guardar la imagen amplificada y volver al formulario anterior.

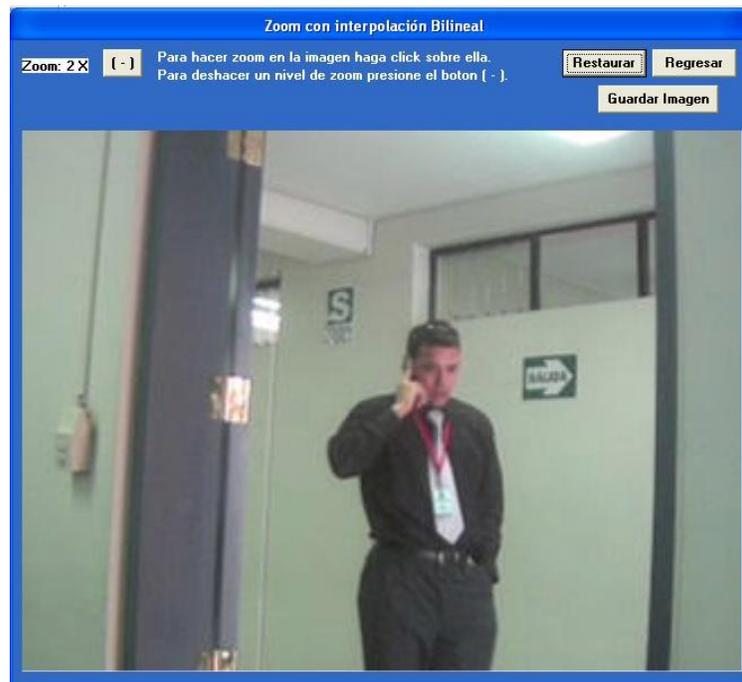


Figura 4.22 Imagen amplificada utilizando interpolación bilineal.

4.7 Implementación de alarmas sonoras

Para la implementación de las alarmas sonoras se ha realizado la conversión de texto a voz haciendo uso de tres aplicaciones: el MSagent.exe, el lhttspe.exe y el spchapi.exe, las cuales son tecnologías desarrolladas por Microsoft. De los tres componentes, el lhttspe.exe (Lernout & Hauspie TTS3000 TTS engine - Spanish) es el motor de voz en español y tiene la capacidad de modificar la tonalidad de la voz, consiguiendo así emular la voz grave del hombre y la voz aguda de la mujer.

Al instalar las aplicaciones mencionadas en el *Builder C++* se genera el componente *SpVoice*, el cual es mostrado en el gráfico de abajo y tiene la forma de una boca humana (ver figura 4.20). A este componente aparte de calibrarle la tonalidad de la voz, sólo es necesario asignarle una variable dinámica de texto para que pueda reproducirla constantemente. Y es con la ayuda de este componente es que se ha implementado un software de detección capaz de hacer sonoros todos los movimientos que el usuario realice dentro del programa y también alarmar al usuario cuando se detecte algún evento.

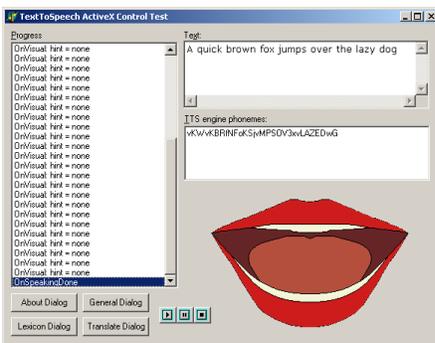


Figura 4.23 Componente *SpVoice* instalado en *Builder C++*.

CAPÍTULO 5

PRUEBAS Y RESULTADOS

En este capítulo se comentará brevemente los resultados obtenidos luego de haber desarrollado el sistema de detección de eventos remotos. El análisis de estos resultados ha sido dividido áreas diferentes como por ejemplo, la conexión entre dispositivos de red, la velocidad de procesamiento de imágenes, la probabilidad de detección de eventos remotos y las especificaciones finales del sistema.

5.1 Conexión entre dispositivos de red

El primer paso al desarrollar el proyecto fue el de conseguir la comunicación entre una cámara IP y el software desarrollado, para ello se valió de un componente ActiveX llamado Gif89a, el cual viene con el software de la cámara IP TRENDnet (ver figura 5.1). Este componente sólo permite la visualización de las imágenes capturadas, más no así la manipulación de sus píxeles, por lo que se hizo necesario transformar la imagen al formato de mapa de bits (BMP), para así llevar a cabo el procesamiento de dichas imágenes.

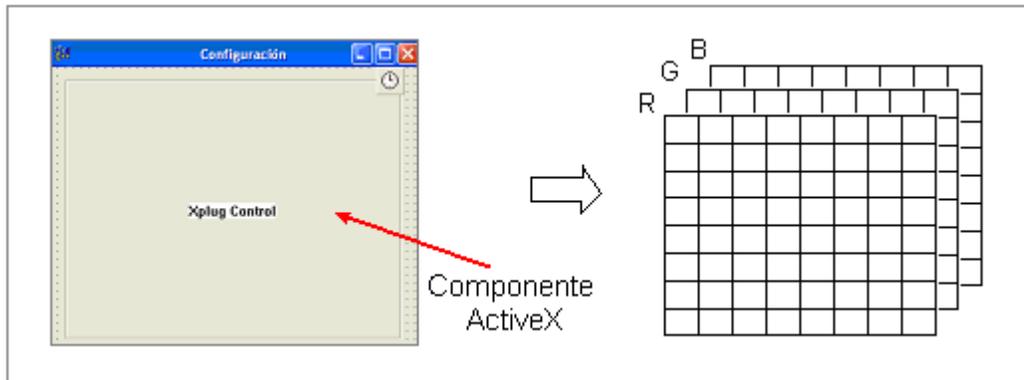


Figura 5.1 Componente para conexión a las cámaras IP.

Cabe mencionar, que en este componente ActiveX no se encontró variable alguna para sincronizar la visualización de las imágenes con la recepción correcta de éstas, es por eso que en el inicio, fue motivo de muchos errores, ya que al no estar sincronizados ambos procesos a veces sucedía que se intentaba visualizar una imagen incompleta generando un error que inhabilitaba al programa. Aunque se logró solucionar este inconveniente usando unos marcadores para sincronizar (por software), esto no es 100% confiable, ya que cuando el procesamiento de las imágenes se hace complejo genera mucho retardo, el cual desestabiliza el sistema generando el mismo error de sincronización.

Es por eso que al desarrollar el programa se trató de no realizar mucho procesamiento para no sobrecargarlo. También es importante recalcar, que el utilizar el componente ActiveX del software de la cámara IP TRENDnet no asegura que también funcione correctamente con otros modelos y marcas de cámaras IP, debido a que mayormente cada dispositivo tiene

características y capacidades diferentes. Por lo mismo, sólo se puede asegurar que el software desarrollado es compatible con la cámara IP modelo TV-IP100 de TRENDnet, aunque probablemente también sea compatible con algunos modelos más.

5.2 Velocidad de procesamiento y visualización del sistema

Se debe tener en cuenta que hay diferencias entre la velocidad de visualización y la velocidad de procesamiento. La velocidad de visualización indica la cantidad de cuadros que se pueden observar en un segundo, mientras que la velocidad de procesamiento indica la cantidad de cuadros que se pueden procesar en un segundo, y esta última permite definir cuántas imágenes se pueden procesar para realizar la detección de eventos.

Tanto para la velocidad de visualización y para la velocidad de procesamiento se debe tener en cuenta la calidad de la imagen y su tasa de compresión, debido a que esto influye directamente en las velocidades. Si se tiene una alta calidad con una baja tasa de compresión, la imagen sería muy buena pero la velocidad de visualización disminuiría considerablemente porque se estaría utilizando un gran ancho de banda, y la velocidad de procesamiento también disminuiría porque la imagen a procesar tendría más detalle. Si se tienen una baja calidad y una alta tasa de compresión, la imagen sería de muy baja calidad pero la velocidad de visualización aumentaría porque se estaría utilizando un menor ancho de banda, y la velocidad de procesamiento aumentaría porque la imagen ya no presentaría muchos detalles.

Con los casos mencionados se debe emplear el factor costo – beneficio, y tratar de compensar la velocidad de visualización y la velocidad de procesamiento. Para realizar esto se debe compensar la calidad de la imagen con la tasa de compresión porque para un sistema de detección de eventos se debe tener una imagen clara y además tener la posibilidad de procesar una buena cantidad de imágenes.

Teniendo en cuenta al factor de procesamiento de la imagen, se estima que el tiempo mínimo de respuesta o tiempo mínimo de procesamiento del sistema para procesar y realizar la detección en una imagen es aproximadamente de 200 milisegundos, debido a que es el tiempo que demora el buffer de recepción en recibir la imagen en forma completa, pasarla a formato JPEG y procesarla. Si se disminuye este tiempo, el sistema puede procesar la imagen y detectar un evento a una mayor velocidad pero el tiempo de estabilidad del sistema se vuelve menor, es decir, al tratar de procesar y detectar a una mayor velocidad, la probabilidad de que el buffer no recepcione completamente la imagen aumenta y por ende se presente un problema en el sistema debido a que no se cuenta con la imagen completa para realizar la detección. Si se aumenta este tiempo, el sistema se vuelve más estable pero la probabilidad de detección disminuye, puesto se están procesando pocas imágenes y la probabilidad de la imagen procesada no presente alteraciones es alta.

Al tener un tiempo mínimo de respuesta, el procesamiento de las imágenes a 7, 15 ó 30 fps no sería posible porque los valores de tiempo de procesamiento serían muy pequeños y el sistema no podría procesar las imágenes a estas velocidades. Esto no impide realizar la visualización de las imágenes a 7, 15 ó 30 fps debido a que se sigue procesando la imagen con el tiempo mínimo de procesamiento, pero se están perdiendo muchas imágenes. Se debe tomar en cuenta

que al tener muchas imágenes por segundo aumenta la probabilidad de saturar el buffer de recepción.

Para el sistema se ha optado por emplear una tasa de visualización de 7 fps puesto que emplear una tasa de visualización de 5 fps significa tener una alta sincronización entre las cámaras y la computadora. La sincronización entre las cámaras y la computadora se puede perder porque la tasa de transmisión de la cámara no es constante, y también porque el procesamiento de la imagen puede tomar más tiempo si es que la imagen es de alta calidad o si ésta presenta muchos detalles. Al emplear una tasa de visualización de 7 fps se corrige en un mayor grado la pérdida de imágenes o la posibilidad de saturar el buffer de recepción, y además porque se evita el problema de la sincronización.

El programa de detección desarrollado ha sido programado para recibir las imágenes capturadas de las cámaras IP a una velocidad de procesamiento de cinco imágenes por segundo (5 fps). Sin embargo, esta tasa de transferencia es sólo teórica, ya que la tasa real obtenida es variable debido a diversos factores que ralentizan el sistema.

Entre estos factores se encuentra el tiempo de procesamiento de las imágenes que hay entre una recepción y otra. Este tiempo de procesamiento asignado para realizar la detección de eventos retarda ligeramente el funcionamiento del sistema cuando se trata de una sola cámara, pero cuando el sistema opera con cuatro cámaras IP el retardo se hace más notable para el usuario. A continuación, se muestra una tabla con datos aproximados obtenidos a partir de las múltiples pruebas de detección de movimiento, realizadas en el laboratorio.

Tabla 5.1 Respuesta del sistema respecto a calidad visual.

Tasa de visualización	Tiempo de actualización de la imagen (milisegundos)	Calidad visual
5 fps	200	Pausada (retardo considerable)
7 fps	143	Semi-pausada (retardo ligero)
15 fps	67	Normal
30 fps	33	Buena

El sistema desarrollado ha sido probado ante distintas tasas de recepción de imágenes (frames por segundo) obtenidas en la configuración de la cámara IP. Como se muestra en la tabla 5.1, a medida que la tasa de visualización del sistema aumenta la calidad visual para el usuario es mucho mejor, debido a que al haber mayor cantidad de imágenes el video se hace más suavizado. Sin embargo, también se observa que a medida que aumenta la calidad visual del sistema se necesita un menor tiempo en el procesamiento, debido al aumento de las imágenes. Esto hace inestable al sistema pudiendo generar un error de sincronización o de saturación del buffer de recepción e incluso afectando la efectividad del programa al momento de detectar automáticamente los eventos. Por estas razones, se optó por definir la tasa de visualización de imágenes a 7 fps, para que de esta forma el retardo sea mínimo, a pesar de no obtenerse la mejor calidad visual posible.

Por otra parte, el sistema trabaja en buenas condiciones en una red local a pesar del retardo por procesamiento. Sin embargo, cuando el sistema es aplicado a través de Internet, los resultados son diferentes, ya que el tráfico de la red global es mucho mayor y esto causa que

reproducción de imágenes sea incluso menor a tres imágenes por segundo (3 fps). Es por eso, que el sistema está planificado para trabajar preferentemente sólo a través de una red local.

5.3 Probabilidad de detección de eventos remotos

La probabilidad de detección de eventos remotos obtenida por el sistema es relativamente buena. Pero al utilizar múltiples cámaras se ralentiza el sistema y la probabilidad de error en la detección de eventos aumenta, ya que al disminuir la cantidad de imágenes procesadas por segundo, es muy probable que algún objeto atravesase la escena sin ser detectado.

En la tabla 5.2 se muestran algunos datos aproximados, de acuerdo a los resultados de las simulaciones del programa hechas en el laboratorio. Para realizar las pruebas se instalaron las cámaras en puntos estratégicos del laboratorio, vigilando puertas y diversos objetos, los cuales se emplearon como objetos de prueba para simular eventos y registrar la efectividad de la detección. De estos datos se puede observar que el aumentar demasiado la tasa de visualización de imágenes podría perjudicar la probabilidad en la detección automática de eventos. Por lo que, fue conveniente elegir una tasa de visualización de 7 fps y una tasa de procesamiento de 5 fps para obtener una buena probabilidad de detección.

Tabla 5.2 Respuesta del sistema respecto a detección de movimiento.

Tasa de procesamiento	Tiempo de actualización de la imagen (milisegundos)	Probabilidad de detección
5 fps	200	Detección óptima
7 fps	143	No es posible, tiempo de procesamiento es menor al mínimo requerido.

15 fps	67	No es posible, tiempo de procesamiento es menor al mínimo requerido.
30 fps	33	No es posible, tiempo de procesamiento es menor al mínimo requerido.

Durante las pruebas se observó que si las cámaras enfocan a zonas con mucha variación de iluminación como ambientes externos o ventanas, etc. la probabilidad de realizar una falsa detección se incrementa. Si las cámaras enfocan en zonas con muy poca iluminación, la probabilidad de detectar eventos disminuye. Por otra parte, si un objeto extraño cruza las regiones de los objetos vigilados a una velocidad superior a la de visualización y de procesamiento, entonces este objeto no será detectado por el sistema.

En resumen, mientras el sistema trabaje en una red local, la probabilidad de efectuar correctamente la detección de eventos remotos es aceptable. Es necesario que las cámaras estén ubicadas estratégicamente para que se pueda visualizar un área grande dentro de cada escena y de esta manera hacer que cualquier evento que se detecte sea visualizado en la mayor cantidad de imágenes posible para que así la probabilidad de error en la detección de eventos sea mínima.

5.4 Consideraciones importantes

Como anteriormente se ha mencionado, el sistema ha sido desarrollado para trabajar remotamente a través de una red local (LAN), aunque también tiene la capacidad para trabajar con cámaras IP conectadas a Internet. Sin embargo, para funcionar a través de Internet el

sistema requiere que las cámaras IP a conectarse posean una dirección IP pública fija, lo cual es un servicio que brindan los proveedores de Internet a un costo adicional.

Cabe mencionar, que las empresas dedicadas a la implementación de sistemas de vigilancia, para trabajar a través de Internet usualmente utilizan sus servidores a los cuales se conectan las cámaras IP, de esta manera los usuarios pueden visualizar sus cámaras IP sin la necesidad de tener que utilizar ningún programa, sino simplemente a través de un browser. Sin embargo, mantener un servidor también representa un elevado costo, el cual para un proyecto como el que se ha desarrollado no era factible, eso además de la complejidad que significa desarrollar un sistema similar.

Es debido a los factores arriba mencionados, que durante la elaboración del proyecto se definió que éste debía ser realizado preferentemente para funcionar a través de una red local (LAN), ya que aparte de los inconvenientes para operar en Internet también sucedía que la efectividad del sistema se iba a reducir considerablemente.

También se debe tener en cuenta que para el cálculo de la velocidad del objeto se asume que éste se mueve en forma paralela a la cámara. Además la distancia de la cámara al objeto puede variar por lo que la velocidad del objeto calculada es aproximada.

5.5 Especificaciones finales del sistema

A continuación, se muestra en una tabla todas las especificaciones obtenidas para el programa ya finalizado:

Tabla 5.3 Características finales obtenidas del programa desarrollado.

DATOS TÉCNICOS DEL PROGRAMA DE DETECCIÓN DE EVENTOS	
Compatibilidad	TRENDnet TV-IP100
Ratio visualización	30 fps
Ratio de procesamiento de imágenes	10 fps (aproximadamente)
Sistema Operativo	Windows XP – Service pack 2
Máximo número de cámaras conectadas	4 cámaras IP
Soporte PTZ	Control de cámaras PT o PTZ (No disponible) Zoom digital (hasta 8X)
Interfaz de usuario	Entorno visual desarrollado en Builder C++
Programación de gradación ante alarma	Detección de movimiento Cambio de iluminación
Visualización en vivo	Multi-cámara (4 cuadros de visualización) Video-rondas (Secuencial)
Notificación de alarmas	Aviso visual de alarmas Aviso sonoro de alarmas Registro de eventos en Excel
Búsqueda de grabaciones	Búsqueda por fecha y número de cámara
Reproducción	Reproductor de video en formato AVI Analizador de vídeo imagen por imagen en formato JPEG

Exportación de video e imágenes	Exportación de video en formato AVI Exportación de imágenes en formato JPEG
Administración	Gestión de administrador
Idiomas	Español
Procesador	Intel Pentium 4 3.4Ghz
Memoria	512 MB RAM
Resolución de la tarjeta VGA	800x600 ó superior
Interconexión del sistema	Red de Área Local (LAN o Intranet)
Protocolo de comunicación	TCP / IP
Entradas / Salidas	Puerto RJ-45 LAN Salida de audio
Configuración de cámaras IP	Acceso mediante un browser a las propiedades avanzadas de la cámara IP
Selección de objetos	Hasta 10 objetos por cámara Ajuste de sensibilidad de detección por objeto Ajuste de sensibilidad de cambio de iluminación por cámara
Reinicio de sistema	Mediante el botón <i>Reestablecer</i>
Audio interactivo	Seguimiento de las acciones del usuario mediante un motor de voz en español
Analizador de vídeo	Posición, velocidad y dirección de un objeto en la grabación de vídeo
Descripción de los dispositivos conectados	Mediante el botón <i>Ver Detalles</i>

CONCLUSIONES

- Se demostró que realizar la detección de eventos remotos utilizando procesamiento digital de vídeo es factible y de hecho la mejor alternativa para detectar eventos automáticamente, ya que permite un control óptimo de los píxeles de las imágenes capturadas.
- Se optó por desarrollar un sistema de detección de eventos remotos aplicado a una red local, debido a los problemas que representa trabajar a través de Internet, como son la conexión (direccionamiento IP) y la velocidad de procesamiento.
- Si el sistema se encuentra trabajado en una red local, la probabilidad de efectuar correctamente la detección de eventos remotos es aceptable, ya que el flujo de información no ralentiza mucho el sistema.

- Se estableció que a pesar de haber definido valores apropiados para el nivel de los umbrales en la detección de eventos, es necesario que el usuario pueda ajustar el nivel de sensibilidad según sea el contraste de las imágenes capturadas.
- El algoritmo de grabación de vídeo desarrollado permite grabar una cantidad infinita de vídeos e imágenes, lo cual debe ser controlado de forma manual por el usuario, ya que podría llegar a saturar los recursos de la computadora.
- Para calcular la velocidad de un objeto en movimiento (en metros/segundos) es necesario conocer el ángulo de apertura de la cámara y la distancia aproximada que hay entre la cámara y la trayectoria del objeto, ya que con ellos se puede hallar un equivalente en metros de los 320 píxeles del ancho de la imagen.
- Es conveniente que las cámaras estén ubicadas estratégicamente de tal forma que cualquier evento que se detecte sea visualizado en la mayor cantidad de imágenes posible, para que así la probabilidad de error en la detección de eventos sea mínima.

RECOMENDACIONES
PARA TRABAJOS FUTUROS

En esta última parte del proyecto se han elaborado algunas propuestas para posibles trabajos futuros que se realicen con el afán de mejorar este proyecto. A continuación, se mencionan algunas recomendaciones específicas:

- Para añadir al programa la capacidad de trabajar con múltiples marcas y modelos de cámaras, es necesario primero definir estas marcas y modelos con las que se desea operar para luego agregar al programa sus respectivas librerías de componentes. Además, es necesario colocar una interfaz para que el usuario pueda seleccionar la marca y el modelo de cámara que va a utilizar.

- Para que el sistema de detección pueda visualizarse mediante Internet, el programa desarrollado debe ser modificado a una aplicación Web para poder ser cargado en un servidor, al cual puedan conectarse los usuarios vía Internet.
- Uno de los problemas principales durante el desarrollo del proyecto fue la sincronización con la cámara IP, para subsanar este problema se recomienda analizar la estructura de la trama de datos enviada por la cámara IP, para así optimizar la recepción de imágenes y mantener la tasa de recepción de éstas.
- En el programa desarrollado para determinar la velocidad y la dirección de un objeto en la imagen se empleó el centroide de los píxeles variantes en toda la imagen, lo cual asume el movimiento de varios objetos dentro de la imagen como el de un sólo objeto. Para optimizar este proceso se recomienda modificar el programa para que seleccione una región de interés y así identificar el objeto a analizar.
- Si se desea ampliar las capacidades del programa (detección de otros eventos), se debe tener en cuenta que el consumo de recursos aumenta mientras más procesamiento se incluya en el programa.
- Si se emplean cámaras con funciones PT o PTZ se puede desarrollar un seguidor de movimiento utilizando el cálculo de la dirección del objeto.

- Se puede desarrollar como valor agregado al programa desarrollado, el envío de mensajes de texto vía telefonía celular, en caso el disco de almacenamiento se llene.

TABLA DE ABREVIATURAS

IP:	Término informático de protocolo de Internet.
CCTV:	Circuito Cerrado de Televisión.
CCD:	Dispositivo de cargas eléctricas interconectadas.
RGB:	Hace referencia a la composición del color en términos de la intensidad de los colores primarios (Rojo, Verde y Azul).
CMYK:	Modelo de colores sustractivo que se utiliza en la impresión a colores (Cyan, Magenta, Amarillo y Negro).
CMY:	Modelo de colores sustractivo que se utiliza en la impresión a colores (Cyan, Magenta y Amarillo).

BIBLIOGRAFÍA

1. HOGAR DIGITAL

- 2007 (http://www.technoimport.com.co/camaras_de_internet.htm)
Contiene información relevante acerca del uso, aplicación y ventajas de las cámaras IP.

2. YDEON

- 2007 (<http://www.ydeon.com>)
Consideraciones para el diseño del sistema.

3. WIKIPEDIA

- 2007 (<http://es.wikipedia.org/wiki/P%C3%ADxel>)
Contiene información relevante sobre la definición del píxel.

4. DARIOLA.NET

- 2006 (http://www.dariola.net/?page_id=23)
Página web orientada a analizar la práctica artística en la Internet como soporte audiovisual, y al intercambio de recursos y herramientas para artistas.

5. INDIGOVISION

- 2006 (<http://www.indigovision.com/files/international/IP-Cameras%20-%20Apr06Spanish-Revised.pdf>)
Resumen sobre la tecnología de cámaras IP, y explica los desafíos a los que ésta enfrenta en las comparaciones de rendimiento.

6. DEPARTAMENTO DE SISTEMAS INFORMÁTICOS

- 2006 (<http://www.info-ab.uclm.es/asignaturas/300204/Temas/Tema1.pdf>)
Compresión de Datos Multimedia.

7. JAP SISTEMAS

- 2006 (<http://jap-sistemas.com.ar/Archivos/CamarasVigilancia.pps>)
Uso de las cámaras IP en los Sistemas IP.

8. MOLINA GONZÁLEZ, Dolores

- 2006 Departamento de Ingeniería Electrónica, de Telecomunicación y Automática.
(<http://www4.ujaen.es/~mdmolina/>)
Contiene información sobre informática.

9. GARCÍA MATEOS, Ginés

- 2005 Transformaciones geométricas (Interpolación Bilineal).
(<http://dis.um.es/~ginesgm/files/doc/pav/tema4.pdf>)

10. CLUB DELPHI

- 2005 (<http://www.clubdelphi.com/foros/archive/index.php/t-19264.html>)

Información sobre archivos Excel en Builder C++.

11. UNIVERSIDAD DE VALENCIA (UV)
2005 (http://www.uv.es/~acuna/tema_4.htm)
Operaciones de procesado de imagen.
12. GLOSARIO DE TÉRMINOS Y DEFINICIONES DE VIDEO VIGILANCIA
2004 (<http://www.glossary-of-terms.net/glossary-of-video-surveillance-terms.html>)
Contiene definiciones de los términos de video vigilancia
13. AXIS COMMUNICATIONS
2004 (<http://www.axis.com/>)
Compañía dedicada a la tecnología de información que brinda soluciones de video en red para instalaciones profesionales.
14. VOXDATA COMUNICACIONES IP / CÁMARAS IP
2003 (<http://www.voxdata.com.ar/voxcamip.html>)
Contiene información sobre los requerimientos, condiciones de uso de las cámaras IP, así como su definición.
15. MICROSOFT AGENT
2003 (<http://www.microsoft.com/msagent/>)
Componente para la conversión de texto a voz.
16. CONFIGURAR EQUIPOS
2002 (<http://www.configurarequipos.com/doc511.html>)
Contiene información acerca de las cámaras IP y sus aplicaciones comunes.
17. IPRONET
2002 (<http://www.ipronet.es/faq.php?faq=62>)
Compañía especializada en soluciones globales para Video IP.
18. TRENDNET
2001 (<http://www.trendnet.com/sp/products/TV-IP100.htm>)
Empresa dedicada a la oferta de una línea completa de soluciones de red de alta calidad asequibles tanto para los pequeños y medianos negocios como para usuarios domésticos.
19. GSCS Software
1997 Sistemas de seguridad, vigilancia y monitoreo remoto
(<http://www.gscssoftware.com/>)
Contiene información sobre las cámaras IP.
20. GONZÁLES, Rafael y WOODS, Richard
1996 Tratamiento digital de imágenes
Addison-Wesley Iberoamericana S.A.

ANEXO

MANUAL DE USO DEL SISTEMA

Para emplear el sistema se debe seguir con las siguientes instrucciones:

1. Para ingresar al sistema debe ubicar el archivo ejecutable denominado “Pr_SistemaVigilancia.exe” en la ruta en el cual instaló el sistema.

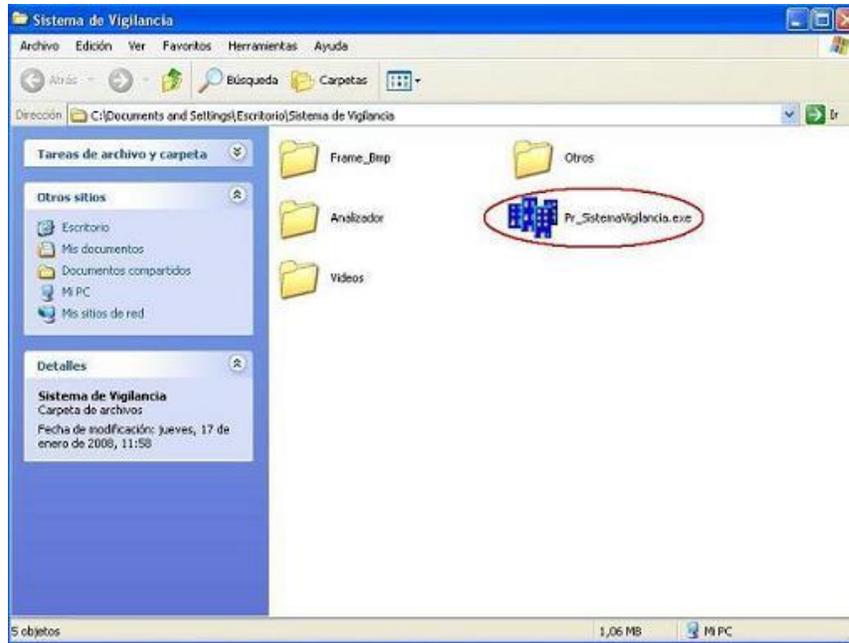


Figura. Localización del archivo de aplicación del sistema.

2. Al ingresar al sistema, se accederá a la siguiente ventana de presentación denominada ventana principal.

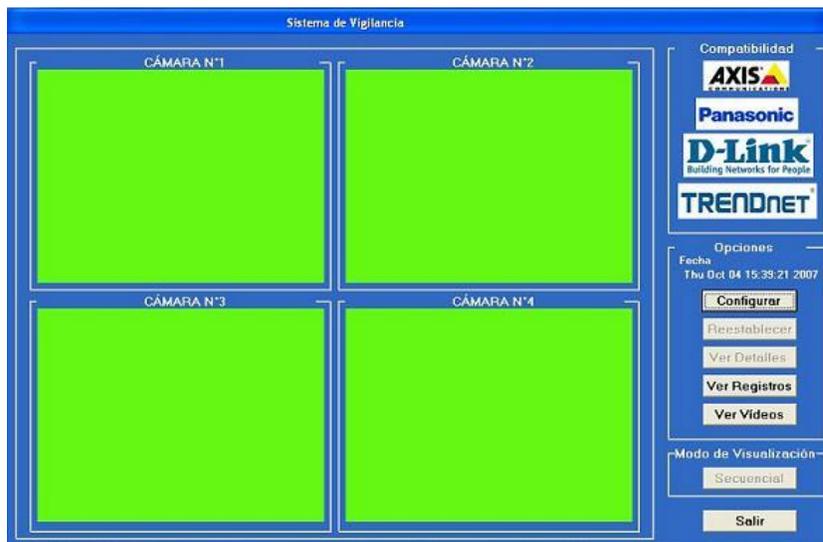


Figura. Ventana principal del sistema.

3. Para acceder a las cámaras IP y configurar el sistema para el modo de detección de eventos, debe seleccionar la opción denominada “Configurar”.

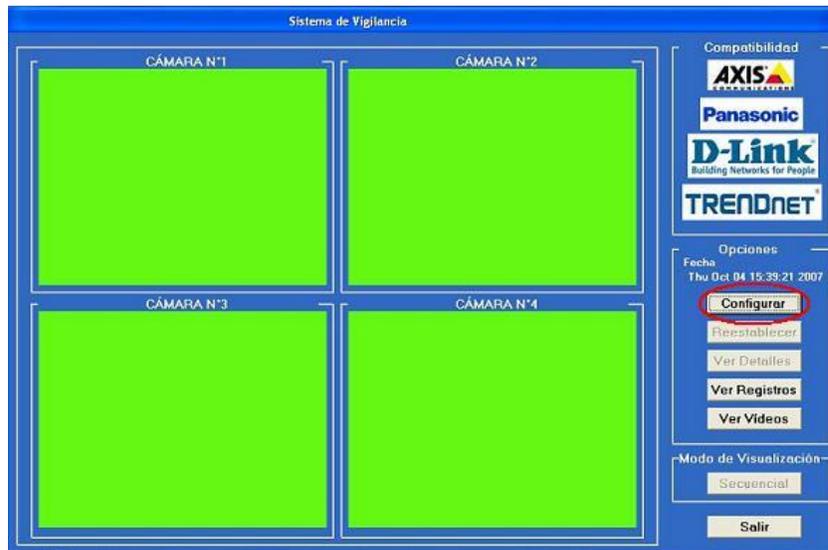


Figura. Botón de configuración seleccionado.

4. Al seleccionar la opción “Configurar”, se mostrará la siguiente ventana. Aquí se deberá seleccionar el número de cámara (los número de las cámaras varían entre 1 y 4) y llenar los datos requerido (dirección IP de la cámara y colocar una descripción de la cámara en los espacios destinados al campo “Modelo” y “Ubicación”). Una vez ingresados los parámetros requeridos seleccione la opción “Conectar”. Si desea desconectar la cámara seleccione la opción “Desconectar”.



Figura. Botón Conectar deshabilitado porque ya se realizó la conexión.

5. Para acceder al “Modo de Calibración” debe seleccionar la opción “Calibrar”. Para calibrar el sistema se debe realizar los siguientes pasos:
 - a) Capturar la imagen de referencia presionando el botón “Capturar Imagen”.
 - b) Una vez obtenida la imagen de referencia, se debe seleccionar la cantidad de objetos que se van a monitorear (el número de objetos que se pueden monitorear varían entre 1 y 10).
 - c) Luego seleccione la región de interés, coloque una descripción del objeto, gradúe el grado de sensibilidad del objeto y de la toda la imagen. Al colocar todos los requerimientos de calibración seleccione la opción “Ingresar”. Nota: Este paso se debe realizar por objeto, excepto la calibración de sensibilidad de toda la imagen que se realiza una sola vez.

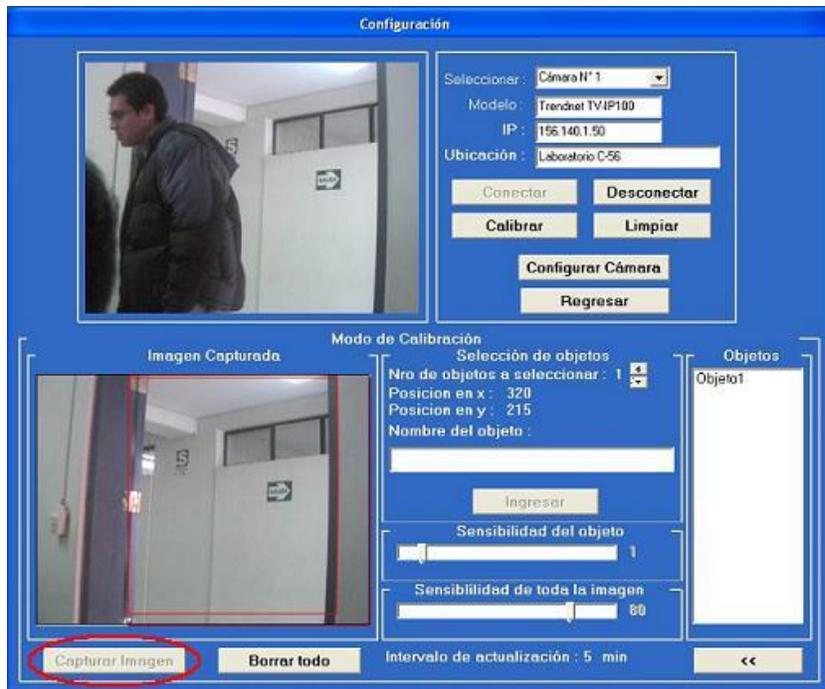


Figura. Captura de imagen en el modo de calibración.

6. Al finalizar la calibración para la detección de evento, seleccione la opción “Regresar” para retornar a la ventana de principal, y el sistema ya se encuentra listo para la detección automática de eventos.

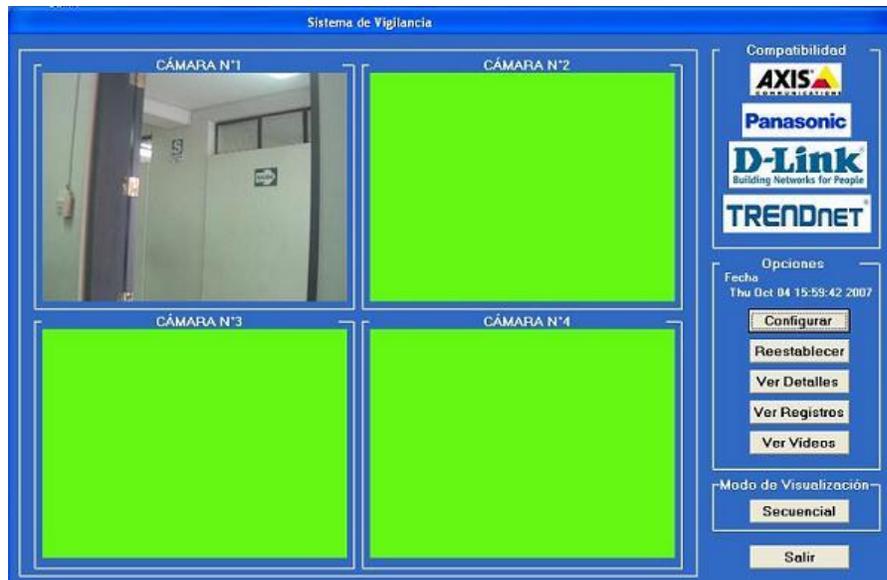


Figura. Sistema conectado en modo detección de eventos.

7. La detección automática de eventos del sistema puede registrar 2 tipos de eventos.
 - a) La detección de objetos extraños en las zonas seleccionadas en el “Modo de Calibración”.

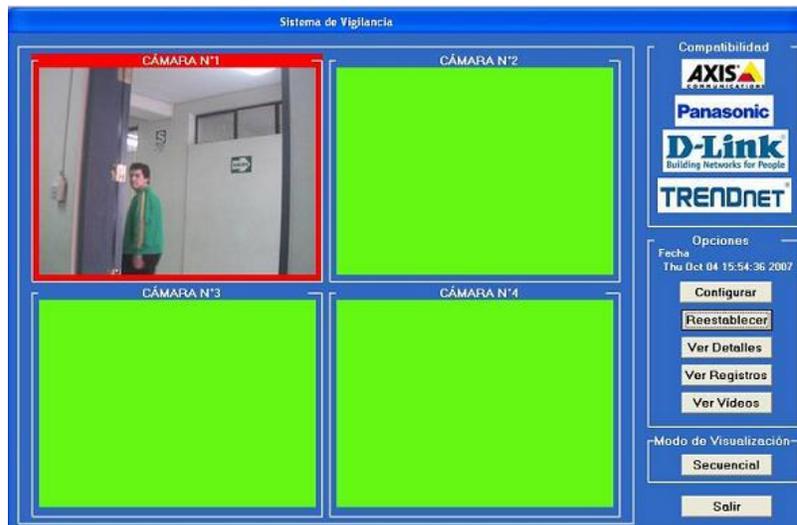


Figura. Detección de movimiento.

- b) El sistema puede detectar el cambio de iluminación del ambiente el cual está monitoreando.

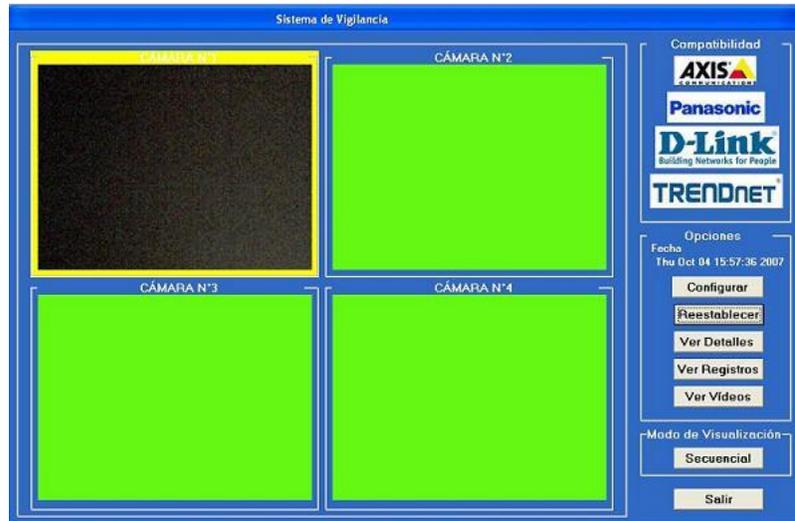


Figura. Detección de cambio de iluminación.

8. Para retornar el sistema a su estado inicial o de detección, se debe seleccionar la opción “Reestablecer”. Esta opción detiene las alarmas y reinicia el sistema para la siguiente detección.



Figura. Sistema reestablecido con los parámetros iniciales de detección.

9. La opción “Ver Detalles”, en el lado derecho se muestran las cámaras que actualmente están conectadas. Para observar la distribución de las cámaras en el lugar, se necesita grabar el plano del lugar en el sistema.

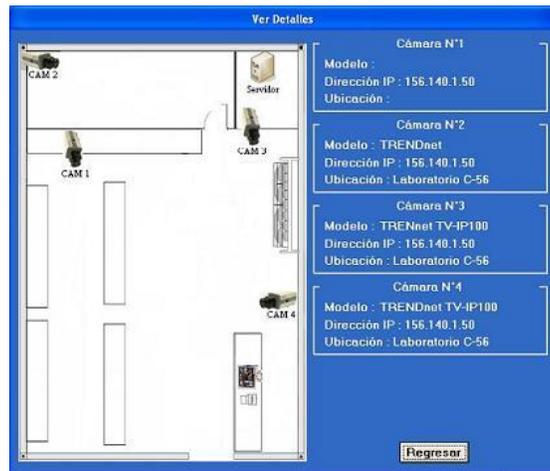


Figura. Localización y descripción de las cámaras IP.

10. Para visualizar los eventos detectados debe seleccionar la opción “Ver Registros”, en este registro se visualizarán todos los eventos detectados por el sistema. Para almacenar los eventos registrados en un archivo Excel se debe seleccionar la opción “Generar archivo de Excel”.

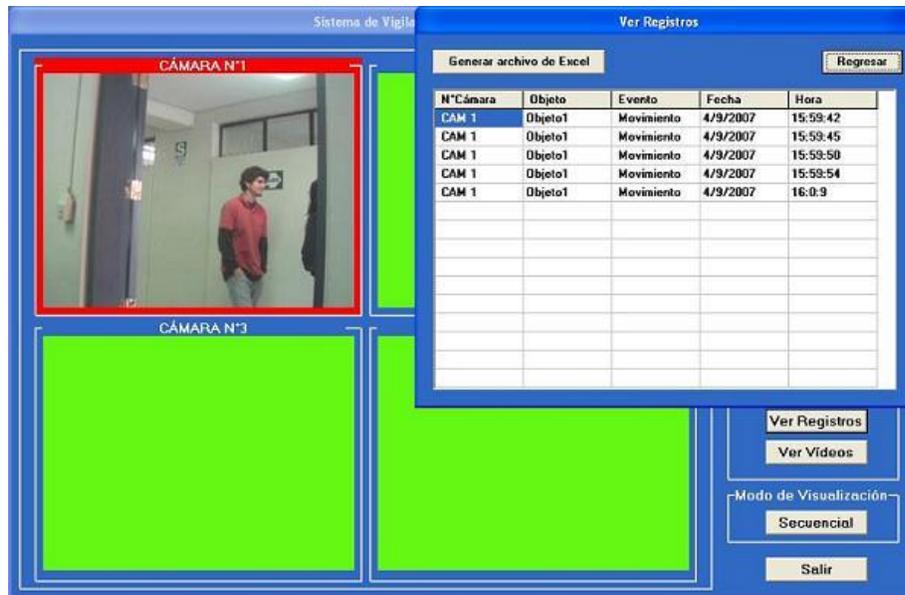


Figura. Ventana donde están registrados los eventos detectados.

11. Para visualizar los videos registrados por el sistema, se debe elegir la opción “Ver Vídeos”. Al ingresar a esta opción se mostrará la ventana en el cual se pueden observar los videos registrados. Para observar los videos se debe seguir con los siguientes pasos:

- a) Seleccionar el número de cámara.
- b) Seleccionar el día.
- c) Seleccionar el número de video y presionar el botón de “Play”



Figura. Ventana para visualización de videos.

Dentro de la ventana de “Ver Vídeos” se tiene la opción “Analizador de Vídeo”. Esta opción permite analizar al objeto extraño dentro de la imagen, calculando su velocidad y dirección. Para realizar el análisis del vídeo se debe colocar una referencia de la distancia del objeto y también se debe colocar referencias de la imagen para poder obtener un resultado más detallado del analizador de vídeo. Si se desea observar una imagen con mayor detalle, el analizador de vídeo permite realizar un “Zoom” digital, seleccionando la opción “Zoom”. Para utilizar el acercamiento sólo se debe seleccionar el punto de interés y para el alejamiento se debe presionar el botón con el signo “-”.



Figura. Ventana para análisis de vídeos.



Figura. Ventana para amplificar la imagen capturada.

12. La última opción que presenta la ventana principal es la de “Modo de Visualización Secuencial”. Este modo permite visualizar las cámaras conectadas de forma independiente.



Figura. Modo de visualización secuencial de una sola cámara.