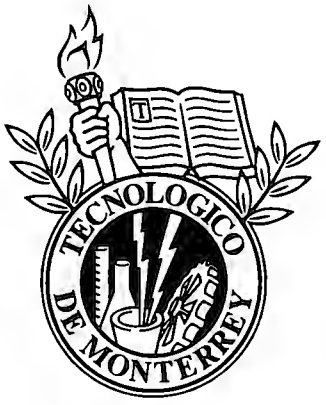
INSTITUTO TECNOLÓGICO Y DE ESTUDIOS SUPERIORES DE MONTERREY

CAMPUS CIUDAD DE MEXICO

ESCUELA DE GRADUADOS EN INGENIERÍA Y ARQUITECTURA

MAESTRÍA EN COMERCIO ELECTRÓNICO



TECNOLÓGICO
DE MONTERREY.
BIBLIOTECA
Campus Ciudad de México

SISTEMA AUTOMÁTICO DE MONITOREO VISUAL

TESIS PRESENTADA POR:

HÉCTOR HUMBERTO DÍAZ LÓPEZ

[redacted]

AGOSTO DEL 2004

Resumen

Nombre del proyecto

Sistema automático de monitoreo visual

Autor

Héctor Humberto Díaz López

Fecha de realización del proyecto

30 de agosto del 2004

Campus

Ciudad de México

El objetivo del proyecto consiste en generar un algoritmo que permita analizar imágenes para detectar cambios de posiciones en objetos o bien diversos niveles de movimiento.

Desarrollar una aplicación capaz de implementar el algoritmo generado en donde se permita la configuración de niveles de alerta para realizar notificaciones por medio de correo electrónico.

Almacenar la información generada por la aplicación conservando la calidad de los documentos y que los videos generados sean menores en tamaño que la suma de las imágenes que los componen.

La metodología consistió en desarrollar una aplicación para captura de imágenes (APCANI) la cual se encarga de analizar secuencias de imágenes

para convertirlas en video con un formato de compresión que permite reducir el tamaño de los archivos.

Se implementó una base de datos para el almacenamiento de las referencias sobre cada video capturado, también se desarrolló una configuración de alertas para realizar notificaciones vía correo electrónico.

Finalmente, dentro de los resultados obtenidos, se logró discriminar el movimiento en secuencias de imágenes con lapsos de tiempo muy corto.

Se descubrió que los niveles de alerta funcionan adecuadamente cuando los tiempos de grabación continuos son pequeños debido al algoritmo desarrollado se basa en promedios.

En cuanto a recomendaciones para trabajo futuro, se encontró la posibilidad de modificar la implementación del almacenamiento de los archivos, sustituyendo la base de datos con un archivo XML que guarde todas las referencias.

Se recomienda realizar métodos de compresión adicionales que permitan reducir aún más el tamaño de los archivos conservando la calidad de las imágenes.

El manejo de múltiples cámara por instancia de APCANI ayudaría a tener más información concentrada en un mismo lugar.

Palabras clave

Sistema de monitoreo, circuito cerrado de televisión, cámara, video, base de datos, error cuadrático medio, compresión de imágenes.

Índice

Agradecimientos ii
Resumen iii
Índice vi
Índice de tablas, diagramas y gráficas viii

Introducción 9
Investigación previa 13
1. Sistemas de monitoreo 13
2. Sistemas de circuito cerrado 14
2.1 Cámaras 14
2.2 Pantallas 15
2.3 Dispositivos de almacenamiento 15
3. Productos existentes en el mercado 16
3.1 DigiOpG2 16
3.2 Axis Network 20
3.3 TrueLook 23
Análisis de fundamentos 27
1. Imagen 27
2. Procesamiento digital de imágenes 28
3. Compresión de imágenes 29
3. 1 Algoritmos de compresión de imágenes sin pérdida de información 31
3.2 Algoritmos de compresión de imágenes con pérdida de información 31
4. Medios de transmisión 32
4.1 Medios Magnéticos 32
4.2 Par Trenzado 32
4.3 Cable coaxial 33
4.4 Fibra óptica 33
5. Almacenamiento de datos 33
Base de datos 34
DBMS (Database Management System) 34
Definición del problema 37
Descripción de la aplicación 40
Captura y configuración 41
Análisis 41
Compresión y generación de video 42
Almacenamiento 43
Servidor de Web 44
Implementación de APCANI 45
1. APCANI (Aplicación de captura y análisis de imágenes) 45
1.1 CBitmap2Avi 49
1.2 CBitmap2Avi::Convert 49

1.3 CBitmap2Avi::CompressBitmap 50
1.4 CImageAnalyzer 51
1.5 Diagrama de clases 52
2. Base de datos 55
3. Página de Internet 60
Calibración 64
1. eRMSAverage 66
2. eRMSMaxValue 67
3. MaxErrorPercentage 67
4. Escenario diurno con objeto fijo 68
4.1 Descripción 68
4.2 Análisis 68
5. Escenario diurno con objeto pasando frente a la cámara 71
5.1 Descripción 71
5.2 Análisis 71
6. Escenario diurno con cámara en movimiento 73
6.1 Descripción 73
6.2 Análisis 73
7. Escenario nocturno con objeto fijo 74
7.1 Descripción 74
7.2 Análisis 75
8. Escenario nocturno con objeto pasando frente a la cámara 76
8.1 Descripción 76
8.2 Análisis 77
9. Escenario nocturno con cámara frente a lámpara y apagón repentino 78
9.1 Descripción 78
9.2 Análisis 78
Resultados 80
Conclusiones 81
Recomendaciones 83
1. Trabajo Futuro 83
1.1 Manejo de múltiples cámaras 83
1.2 Métodos de compresión adicionales 83
1.3 Manejo remoto 83
1.4 XML 84
1.5 Selección de área 86
Bibliografía 87

Índice de Tablas

Tabla 1 Tipos de cámaras 17
Tabla 2 Productos de Axis Network 22
Tabla 3 Precio solución AxisNetwork 23
Tabla 4 Precio TrueLook por hospedaje 25
Tabla 5 Precio TrueLook con licenciamiento 26
Tabla 6 Listado de Clases APCANI 48
Tabla 7 Listado de tablas de la base de datos. 56
Tabla 8 Listado de procedimientos almacenados 59
Tabla 9 Descripción esquema página de Internet 61

Índice de Diagramas

Diagrama 1 Descripción del flujo de procesamiento 29
Diagrama 2 Descripción de aplicación 40
Diagrama 3 Clase CAppDatabase 53
Diagrama 4 Clase CBitmap2Avi 53
Diagrama 5 Clase CCapThread 54
Diagrama 6 Clase CImageAnalyzer 54
Diagrama 7 Tablas de base de datos APCANI 59

Índice de Gráficas

Gráfica 1 Análisis de escenarios. 66
Gráfica 2 Escenario diurno con objeto fijo. 69
Gráfica 3 Escenario diurno con objeto fijo en 10 segs 70
Gráfica 4 Escenario diurno con objeto pasando frente a la cámara en 5 segs 71
Gráfica 5 Escenario diurno con objeto pasando frente a la cámara 10 segs 72
Gráfica 6 Escenario con cámara en movimiento. 73
Gráfica 7 Escenario nocturno con objeto fijo 5 segs 75
Gráfica 8 Escenario nocturno con objeto fijo 10 segs 75
Gráfica 9 Escenario nocturno con objeto pasando frente a la cámara. 77
Gráfica 10 Escenario nocturno objeto pasando frente a la cámara 10segs 77
Gráfica 11 Escenario nocturno frente a lámpara y apagón. 79

Introducción

La necesidad del ser humano por sentir que tiene el control de todo lo que sucede a su alrededor cada vez se va incrementando más. Sin embargo, el hombre está limitado por naturaleza en dos cosas. Solamente cuenta con dos ojos que le permiten un ángulo visual de 180 grados, eliminándole la capacidad de observar en un mismo tiempo la otra periferia restante. También, su memoria es muy limitada, y cuando hablamos de artículos visuales, es muy complicado recordar todos los detalles. Por ejemplo, cuando vamos a un museo y observamos una pintura, después de algunas horas o bien al día siguiente, sería muy complicado para una persona recordar todos los detalles de la misma, colores, contrastes, posiciones de los objetos, etc.

Ante situaciones como la descrita en el párrafo anterior, el hombre se fue dando cuenta de la necesidad de poder guardar de alguna manera las diversas situaciones, eventos que estuvieran pasando en ese momento para posteriormente ser recordados por futuras generaciones o bien, utilizarlos para actividades científicas.

La capacidad de poder video-grabar eventos de la vida cotidiana ha traído grandes avances científicos ya que se han podido realizar diversos análisis sobre experimentos que a simple vista serían muy complicados de descubrir. Los videos pueden ayudar a observar el crecimiento de una planta, o bien,

grabar el comportamiento del agua en la formación del hielo a ciertas temperaturas para poder generar materiales anticongelantes, etc.

Ante el avance que se ha desarrollado en las grabaciones, ahora se ha caído en un exceso de monitoreo, hoy en día, nos encontramos con cámaras de video colocadas en casi todos los lugares que frecuentamos, desde centros financieros, bancarios, oficinas o incluso escuelas. Los usos de este tipo de videos han sido muy variados, aunque en casi todos los casos se tiene un común denominador: seguridad. Sin embargo, también se tiene la parte de entretenimiento por ejemplo el programa de televisión de "Big Brother" donde el monitoreo es continuo durante las 24 horas hacia una casa con sus habitantes y con más de 10 cámaras grabando al mismo tiempo. Simplemente el almacenaje de toda esa información en manera digital debe andar alrededor de varios discos con capacidad de Gigabytes o bien al menos una cinta de video por cámara por día.

Hasta este momento se ha mostrado la parte de monitoreo, o video grabación a niveles comerciales, o al menos de negocios, sin embargo en los hogares las necesidades de estar monitoreando diversos bienes ha aumentado considerablemente, incluso las empresas de seguridad han tenido que desarrollar equipos para usos domésticos, lo cual incluso los ha forzado a disminuir los precios de sus diferentes equipos. Cuando se habla de un cliente de tipo doméstico, entonces comienzan algunas restricciones, por un lado la

experiencia o conocimiento de hardware y software es muy limitada, los presupuestos son casi nulos, además en ocasiones las necesidades son distintas. Cuando se habla de almacenamiento de video, no se cuenta con grandes volúmenes para almacenar todo lo que se grabe, además el usuario debe poder acceder la información o videos desde cualquier lugar de manera remota de una manera simplificada y sin la necesidad de realizar configuraciones extensivas ni tampoco conexiones de equipo avanzado.

El presente documento se desarrolló en la Ciudad de México, precisamente un lugar muy especial en donde existe una inseguridad a todos los niveles lo cual ha ocasionado que sus habitantes tengan la necesidad de estar protegiendo todos sus bienes, desde objetos personales hasta inmuebles, autos entre otros. Por la situación anterior, se ha vuelto común la existencia de alarmas, circuitos cerrados en las casas, incluso hasta el uso de seguridad privada por aquellos que tienen los medios económicos para hacerlo.

En este documento se pretende mostrar una alternativa para el monitoreo o vigilancia de objetos que pueda ser utilizada por cualquier persona o empresa sin necesidad de realizar configuraciones muy complicadas. En la primera parte del documento, se presentarán algunas alternativas de empresas y productos actuales, posteriormente se describirán algunos conceptos utilizados a lo largo del documento. Debido a que el presente documento no es una investigación, no se profundiza tanto en una base teórica.

La segunda parte del documento se enfoca principalmente en la aplicación desarrollada explicando en detalle sus componentes para finalizar con los resultados obtenidos del desarrollo.

Investigación previa

A continuación se describirán algunos conceptos desde el punto de vista comercial sobre algunos sistemas y su funcionamiento así como también se describirán ciertos productos del mercado cuya funcionalidad es el monitoreo para diversos usos.

1. Sistemas de monitoreo

Es el conjunto de herramientas necesarias para poder determinar los cambios de estado entre objetos o situaciones en un lapso de tiempo determinado. El monitoreo se puede utilizar para diversas actividades, desde plantas de producción en donde se puede determinar el comportamiento en alguna línea de producción, o bien en plantas farmacéuticas para determinar el comportamiento de ciertos fármacos.

Para el presente documento, se determina la utilización del término sistema de monitoreo, a la acción de estar grabando algún objeto para la detección de algún cambio en el mismo ya sea por cambios propios o aquellos generados por agentes externos.

2. Sistemas de circuito cerrado

Los CCTV (Circuito Cerrado de Televisión) es un tipo de aplicación generalmente utilizado con fines de seguridad, en su sitio de Internet [Panasonic, 2003], clasifican los componentes de un CCTV de la siguiente manera:

2.1 Cámaras

La tecnología de las cámaras actualmente ha desarrollado mejoras para cada uno de sus componentes, por lo cual se podría comenzar clasificándola en: blanco y negro, color, industrial a color de micro-cabeza (*color industrial micro head*), domo de color (*color unitized dome*), color a prueba de vandalismo (*color vandal proof*), Para redes, entre otras. [Panasonic Camaras, 2003]

Existen varias características a considerar en la adquisición de una cámara para su utilización en un CCTV, algunas de ellas son la resolución, tamaño, luminosidad de lente, sensibilidad de los sensores, tipo de lente, tipo de telefoto para poder acercar los objetos (zoom) y por último, la capacidad para grabar de noche por medio de lentes infrarrojos.

Desde el punto de vista del autor de este documento, lo más importante es determinar si la cámara tiene una salida análoga o digital ya que para las cámaras digitales permiten exportar los resultados a otras aplicaciones mientras

que las cámaras análogas se encuentran completamente ligadas a dispositivos adicionales que realicen las conversiones necesarias para poder ser exportadas.

2.2 Pantallas

Este tipo de dispositivos permiten al usuario ver la información que fue capturada por los cámaras, generalmente se pueden clasificar en blanco y negro o en color, aunque el precio es un factor determinante en la selección del tipo de pantalla que será utilizada por un CCTV. En general para este tipo de sistemas, las pantallas son blanco y negro, y solamente se cambia el tamaño del dispositivo, ya que a mayor tamaño, se mejora la capacidad de percepción de algunos detalles visuales.

2.3 Dispositivos de almacenamiento

Existen diversos dispositivos de almacenamiento, partiendo desde lo más básico, siendo esto una video casetera VHS hasta dispositivos de almacenamiento digital. Dado el tipo de sistema que la presente tesis desarrollará, tomaremos a los dispositivos digitales para su breve descripción. Lo más importante en este tipo de componentes es el tamaño y la velocidad del disco duro así como algún método de reproducción y grabación adicional ya sean unidades de CD o DVD así como interfaces de conexión para redes IP y algún tipo de unidad de expansión adicional para escalar más discos duros y

aumentar el espacio de almacenamiento. Existen algunos dispositivos que permiten realizar la transformación de las imágenes analógicas a digitales, logrando ofrecer a las empresas tener dispositivos antiguos y modernos en un mismo concentrador de información. Generalmente los dispositivos de almacenamiento manejan para sus entradas y salidas el estándar BNC /Coaxial (Conector cilíndrico) o bien S-Video (Super Video).

3. Productos existentes en el mercado

Se investigaron diferentes soluciones existentes en el mercado. Principalmente se obtuvo información sobre sistemas de monitoreo con fines de vigilancia, dentro de los que se puede destacar, DigiOpG2, Axis Network y True Look entre otros.

3.1 DigiOpG2

DigiOpG2, desarrollado por la empresa "General Solutions", [GeneralSolutions, 2004] especializada en video de alta definición y tecnologías de audio. Dentro de sus objetivos se puede destacar la reducción en los tamaños de los videos, y su fácil integración para aplicaciones de seguridad. DigiOpG2, cuenta con diferentes productos divididos en 3 áreas principalmente, grabadoras digitales de video, grabadores digitales de audio y cámaras. Dentro de las características de sus productos, se destacan:

- Acceso remoto a sus dispositivos
- Tamaño de archivos reducidos
- Calidad mejorada en video
- Aumento en los tiempos convencionales de grabación
- Reducción en los requerimientos de ancho de banda
- Integración con sistemas existentes
- Respaldos simples y revisiones simples

DigiOpG2 integra desde el "hardware" necesario para la captura de video, tanto el desarrollo de un software completamente integrado con el equipo que la empresa vende. Dentro de los tipos de cámaras que DigiOpG2 produce se tienen las siguientes presentadas en la tabla 1:

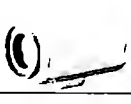
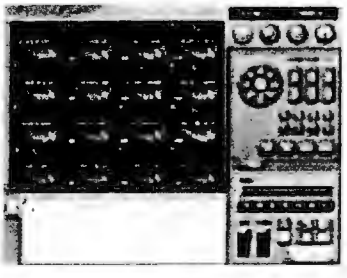
	Alta resolución
	Infrarrojas
	Mini-Domo
	Cámaras de especialidades
	Resolución estándar
	Cámaras de zoom

Tabla 1 Tipos de cámaras de DigiOpG2

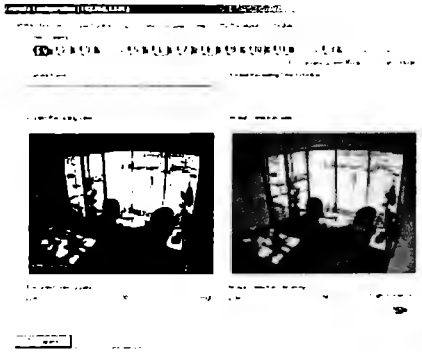
Como se puede observar en la tabla 1, existe una gran variedad de cámaras así como usos diversos, para esto, DigiOpG2 ofrece un sistema de software, que integra las diferentes opciones en un mismo punto de concentración. La solución se encuentra dividida en diversos módulos los cuales son:



1. "Remote & Web Agent": este agente opera en plataformas Windows 2000 y permite obtener video en demanda, video en tiempo real, copiado de archivos, permite el manejo de varios dispositivos, permitiendo un monitoreo remoto y la recepción de notificaciones remotas asociadas a eventos captados mediante sensores de movimiento.

2. "Wireless & Transaction Agent": ofrece tareas de monitoreo remoto, pero pueden ser usados en dispositivos portátiles "PDA´s (Dispositivos de bolsillo "Pocket Device Asistant") principalmente inalámbricos. En cuanto a la parte del agente de transacciones, se debe a su integración con diversos "POS (Point of Sales)" para la administración de datos.

3. "DigiOpG2 Web Agent": utiliza principalmente Internet Explorer y permite a los usuarios tener acceso en tiempo real a la cámara o sobre información que haya sido almacenada.



4. "DigiOpG2" Enterprise Agent": este agente es la integración de los módulos previamente descritos ya que permite tareas de administración remota, así como la consulta por "Web" de actividades en tiempo real o de almacenamiento.

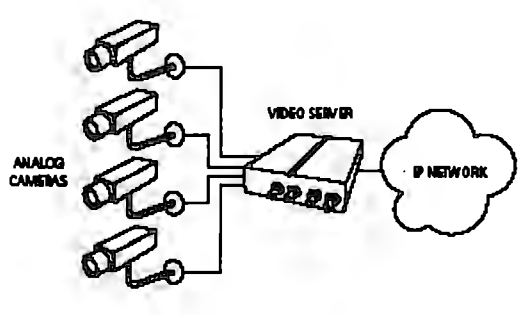
Se investigó el precio de una solución de este producto, y se encontró que únicamente la grabadora digital de video con 16 entradas de video, un disco duro de 120GB, capacidad para manejar hasta 120fps (frames per second) y capacidad hasta por más de dos discos duros y conexión de ethernet. En cuanto al software se incluye el "Remote Agent" y "Web Agent", cuyo costo aproximado es de $2, 400 USD sin contar las cámaras que se deseen adquirir. La instalación final de toda la solución puede elevar el costo dependiendo la cantidad de cámaras requeridas hasta $5,000.00 USD o más.

3.2 Axis Network

Empresa con experiencia en el área de video desde 1984 [Axis, 2004], y en el área de video sobre redes desde 1996. Actualmente cuenta con más de 200,000 productos profesionales instalados.

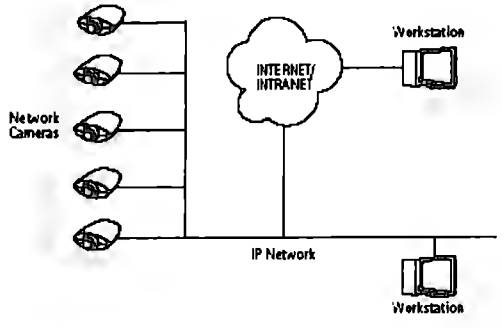
Axis network [Axis, 2004] divide los servicios de red en tres categorías siendo estos:

- Vigilancia "Security Surveillance": dentro de las principales actividades de esta categoría está la protección de personal y bienes.
- Monitoreo remoto "Remote monitoring": consiste principalmente en la video grabación a distancia, por ejemplo, el monitoreo de tráfico en las grandes ciudades, eventos naturales, etc.
- Broadcasting: se refiere a la transmisión de imágenes a través de la red en tiempo real.



Axis network ofrece diversas soluciones desde analógicas generalmente utilizadas para sistemas de CCTV las cuales conectan a un "video server" como se muestra en la figura. Con esta solución, se puede tener acceso a la información de las cámaras desde la red ya que el servidor de video se encarga de digitalizar la información. La

ventaja de esta solución radica en poder utilizar la infraestructura ya existente sin tener que realizar inversiones iniciales.



También se ofrece otra solución en donde ya se tienen la infraestructura digital. Es decir las cámaras ya se encuentran conectadas directamente a la red como se muestra en la figura y pueden ser manipuladas desde cualquier estación de trabajo.

Dentro de las características que Axis Network resalta sobre sus productos están la de tener costos de instalación bajos, así como costos bajos de mantenimiento, tiempos muy bajos para modificaciones, etc.

Al igual que DigiOpG2, Axis tiene una gran gama de productos de los cuales se pueden resumir en la tabla 2 presentada a continuación.

	Cámaras para red con características para interiores y exteriores. Permiten resoluciones hasta de 30fps, detección de movimiento y resoluciones hasta 704 * 480

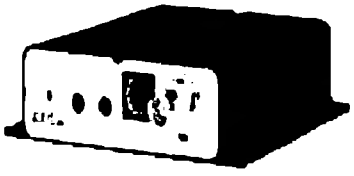
	Servidores de video MPEG-2 con resoluciones de 720 * 576, compresión en tiempo real por medio de hardware en formato MPEG-2. Sistema independiente con un servidor de Web integrado.
	Servidor serial con 2 conectores RS 232 y conexión a red Ethernet con un servidor de Web integrado.
	Access Point Wireless 802.11b para velocidades de 11Mbps con alcance hasta de 300 metros para exteriores y 50 a 100 metros para interiores.
	Software para exploración de cámaras para dispositivos Pocket PC.

Tabla 2 Productos de Axis Network

Como se puede observar Axis Network tiene una gran variedad de productos y especialización en el área de grabación por medio de la red. El mercado objetivo se puede observar que está completamente dirigido hacia áreas comerciales desde empresas gubernamentales para la protección contra el crimen, vigilancia de entradas y salidas a edificios, o bien utilizarlo en

aeropuertos para el monitoreo en la llegada y salida de aviones y por último el uso más común en actividades bancarias frente a cajas o cajeros automáticos.

En cuanto a costos, en su sitio Axis network, [Axis, 2004] provee de un escenario de configuración típico para la implementación de un sistema de seguridad por IP con 16 canales (cámaras) para interiores. El detalle de las características necesarias es el siguiente:

Nota: Los costos se encuentran expresados en dólares americanos

Cámara	$4,800 ($300 por cámara)
Infraestructura (Hardware + instalación)	$600 (Cableado categoría 5) $700 (Switch de 16 puertos)
Hardware de grabación	$2,000 (PC Server)
Software de grabación	$2, 500 (NVR software)
TOTAL	$10, 600
Costo por cámara por canal	$662

Tabla 3 Precio solución AxisNetwork

3.3 TrueLook

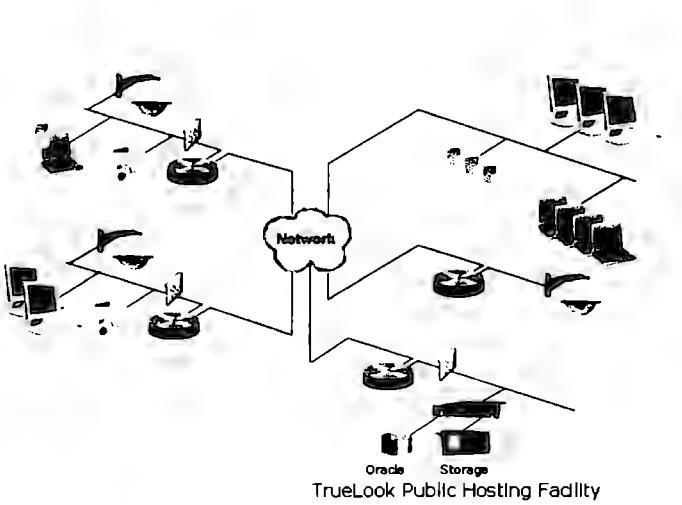
Este producto se encuentra producido por una empresa llamada SilkRoad [Silkroad, 2004] quien se encarga de producir software para diversos propósitos, dentro de ellos se encuentra "TrueLook", [TrueLook, 2004] donde se le define como una plataforma de administración de cámaras en donde cada usuario por

medio del Web puede tener el control de la cámara para realizar cualquier tipo de movimiento zoom, etc.

"TrueLook", se caracteriza por ser simplemente un medio de contacto a un punto remoto, sus aplicaciones consisten en poder tener acceso en tiempo real a lugares lejanos simplificando y ahorrando gastos por tiempos de viaje o viáticos.

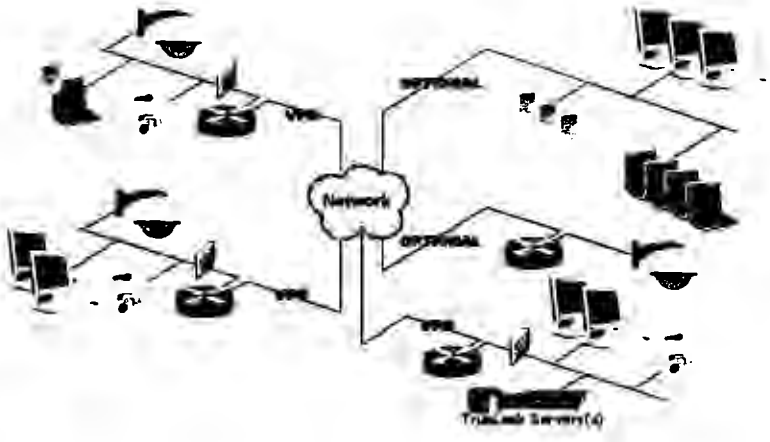
"TrueLook" maneja dos esquemas para su comercialización, por un lado un servicio de hospedaje "Hosted Service" en donde Silkroad se encarga de poner toda la infraestructura necesario ofreciendo ventajas para rápida escalabilidad, servidores estables para acceso por red, donde solamente se requiere a cada cliente tener una cámara con acceso por IP.

Como se puede observar en la gráfica de la derecha, toda la información se concentra en las instalaciones de "TrueLook" quien cuenta con la infraestructura necesaria desde almacenamiento temporal, así como dispositivos de video



TrueLook Public Hosting Facility

El otro esquema utilizado por "TrueLook" consiste en proveer de la infraestructura necesaria por la aplicación para fácil integración en cualquier institución, desde el servidor, hasta cámaras. De esta manera, la administración puede realizarse internamente dentro de la empresa sin tener que comprometer ningún tipo de seguridad.



Headquarters

La figura del lado izquierdo nos muestra claramente que la infraestructura ahora radica en la matriz de la empresa.

En cuanto a los precios manejados por "TrueLook" en sus dos esquemas, se detallan en la tabla 4 y estos son:

Servicio de hospedaje	
Costo por cámara con soporte y actualizaciones	$320.00 USD por mes

Tabla 4 Precio TrueLook por hospedaje

1. Servicio de Licenciamiento	
1 cámara	$7,200 USD por año
Licenciamiento primer año	$1,440
TOTAL	$8, 640.00 USD anual

Tabla 5 Precio TrueLook con licenciamiento

Análisis de fundamentos

1. Imagen

En su libro de Tratamiento Digital de Imágenes, [González, 1987] menciona 2 tipos de imágenes, la monocroma la cual se le refiere como función bidimensional de intensidad de luz f(x,y), donde x e y representan las coordenadas espaciales y el valor de f en un punto cualquiera. De la misma manera, la definición que presenta para imagen digital es que es una imagen I(i, j) que se ha discretizado tanto en las coordenadas espaciales como en el brillo. Puede considerarse como una matriz cuyos índices de fila y columna cuantifican un punto de la imagen y el valor del correspondiente elemento de la matriz indica el nivel de gris en ese punto. Los elementos de una distribución digital de este tipo se denominan píxeles, abreviatura de su denominación inglesa *picture elements.* La definición mencionada anteriormente es muy importante ya que estamos hablando del mínimo elemento que compondrá el sistema, ya que la información que se esté obteniendo del exterior, será dividida en conjunto o serie de imágenes para después poder ser procesada. De esta manera, todas aquellas imágenes discretas, son procesadas por las computadoras, mientras que las imágenes continuas se utilizan principalmente en los sistemas análogos.

2. Procesamiento digital de imágenes

Al mencionar procesamiento de imágenes, es importante destacar que principalmente se está haciendo referencia al cambio del contenido de una imagen con un propósito en particular. Existen diversos propósitos que pueden generar la necesidad de procesar una imagen, entre los que se encuentra la de mejorarla, aumentar o quitar el brillos, aclarar contornos, mejorar contornos, eliminar el color rojo de los ojos que comúnmente aparece en las fotografías, entre otras cosas.

Cuando se habla de procesamiento digital, se debe entender que existe un proceso el cual generalmente sigue los siguientes pasos [González, 1987]: adquisición de imágenes consistente en obtener la imagen a procesar, preprocesamiento en donde se busca principalmente mejorar la imagen para ayudar a los procesos que siguen, segmentación que es partir la imagen en objetos o partes que la formen, descripción o selección de rasgos de donde se extrae información cuantitativa de interés, interpretación donde se le asigna un significado al conjunto de objetos que hayan sido encontrados y finalmente se obtienen los resultados. Gráficamente el procedimiento fluye de la siguiente manera:

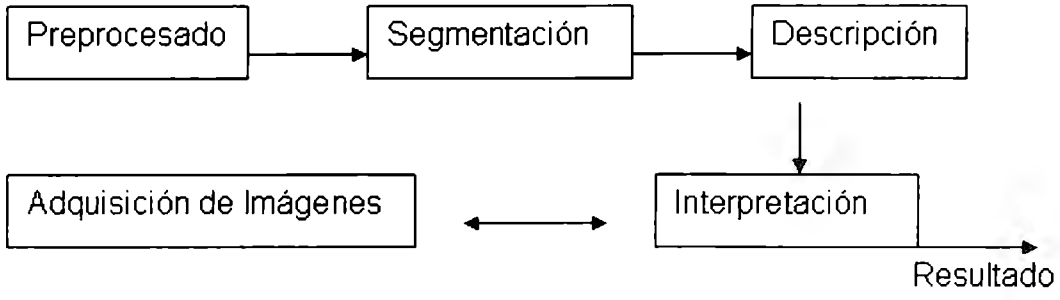


Diagrama 1 Descripción del flujo de procesamiento digital de imágenes

El proceso descrito en la parte superior, será muy similar a los procedimientos que el sistema de monitoreo estará llevando a cabo para la discriminación de imágenes. Partiendo de la adquisición de las imágenes, en donde las cámaras que se encuentren en el sistema enviarán la información al sistema de pre-procesamiento. Posteriormente existe un proceso de segmentación, separación de la información por medio de la interpretación de los datos para la generación de resultados.

3. Compresión de imágenes

La compresión de imágenes se encuentra vinculada con las imágenes digitales, por lo tanto, se está refiriendo a la intensidad existente en los diferentes píxeles. Si se quisiera almacenar o procesar toda la información de una imagen sin un proceso previo de procesamiento, el espacio de almacenamiento necesario sería insuficiente y en algunos casos no podría ser manejado por aplicaciones convencionales.

En el libro de Tratamiento de Imágenes, [González 1987] se hace énfasis en la distinción de datos e información; se aclara que los datos son los medios por los cuales se puede llegar a la información, por esta situación, existen diversas maneras de mostrar la misma información con diferentes datos. Ante lo anterior, se puede notar que la redundancia de los datos en una imagen es fundamental para la efectiva compresión. Por lo anterior, se puede notar que la compresión tiene como objetivo la reducción de datos necesarios para representar una imagen. Por medio de métodos de compresión es posible eliminar todos aquellos datos redundantes de una imagen disminuyendo notablemente su tamaño.

Existen diversos tipos de redundancia en las imágenes, dentro de las cuales se pueden destacar: entre píxeles, de codificación, psicovisual, entre otros. Cada tipo de redundancia corresponde diferentes modelos matemáticos, y a su vez se utilizan para distintos algoritmos.

Un sistema de compresión consta de un codificador y un decodificador [González 1987]. Una imagen f(x,y) de entrada alimenta al codificador para que transforme la imagen y la pase hacia el canal generando f'(x,y) la cual alimenta a un decodificador dando como resultado a f''(x,y). Si al final del proceso, f''(x,y) resulta ser igual que f(x,y) entonces se habla de un canal libre de error ya que no hubo pérdida de información, mientras que en caso contrario cuando las imágenes son distintas, entonces existió una pérdida de información y la imagen puede tener algún tipo de distorsión.

3. 1 Algoritmos de compresión de imágenes sin pérdida de información

Existen diversos algoritmos de compresión sin pérdida de información, aunque el más común es el LZW (Lempel Ziv Welch), [Campos, 1999] algoritmo utilizado por el formato .GIF(Graphics Interchange Format). Este algoritmo funciona de la siguiente manera: se recorre la imagen píxel por píxel comparándolo con el color del píxel de la izquierda; en caso de que los colores sean iguales, entonces se continúa con el siguiente píxel hasta encontrar uno de color distinto; será en ese caso cuando se almacene una secuencia de "n" píxeles con el color común.

3.2 Algoritmos de compresión de imágenes con pérdida de información

Como se mencionó anteriormente, este tipo de algoritmos modifican las imágenes ya que eliminan la redundancia de los datos. El formato más común que utiliza este tipo de algoritmo es el de JPEG (Joint Photographic Experts Group) [JPEG, 2002].

JPEG fue diseñado para explotar las limitaciones imperceptibles del ojo humano como los cambios en los colores los cuales no son detectados tan fácilmente como los cambios en la brillantez.

JPEG tiene una propiedad que permite manipular el grado de pérdida de información solamente ajustando los parámetros de compresión y también la

posibilidad de ajustar la velocidad de decodificación de la imagen contra la calidad de la imagen.

4. Medios de transmisión

Para la transmisión de información por medios electrónicos, existen diversos medios, [Tanenbaum 1996] describe los siguientes:

4.1 Medios Magnéticos

Se refiere principalmente a todos aquellos medios convencionales como son cintas magnéticas, discos de 31/2 o bien CD's aunque cuando se habla de videos o multimedia, estos medios no son muy viables debido a su limitada capacidad.

4.2 Par Trenzado

Generalmente utilizado para aplicaciones en línea. Este medio es el más antiguo y ha sido utilizado por las líneas telefónicas. Tienen la capacidad de transmitir a lo largo de varios kilómetros sin la necesidad de amplificar la señal. La ventaja de este tipo de medio es que permite la transferencia de medios digitales y analógicos.

4.3 Cable coaxial

Generalmente conocido como "coax" y tiene una capa de aislamiento lo cual resulta una ventaja ante el par trenzado porque permite ser utilizado en lugares con mucha interferencia. La capacidad de transferencia en un cable de 1km varía entre 1 o 2Gbps. Este tipo de medio de transmisión también fue utilizado para telefonía aunque ha sido desplazado por la fibra óptica.

4.4 Fibra óptica

Este medio utiliza principalmente la velocidad de la luz. Permite rangos de transmisión de 1 Gbps y esto se debe a los problemas para convertir señales eléctricas a ópticas. Las transmisiones ópticas cuentan con tres componentes, fuente de luz, medio de transmisión y detector. La transmisión se relaciona completamente con la presencia de luz, un bit y la ausencia, 0 bits. El detector sirve para mandar pulsos eléctricos cuando existe la ausencia de luz y de esta manera realizar un control de la transferencia de datos.

5. Almacenamiento de datos

Para el almacenamiento de datos existen diversas soluciones, desde el almacenamiento en archivos planos, el uso de bases de datos o bien, mezclar los archivos planos con una estructura determinada por ejemplo XML.

Base de datos

Es una colección de datos organizados que sirven eficientemente a diferentes aplicaciones al mismo tiempo para centralizarlos y minimizar la redundancia entre los datos. Almacena los datos en archivos separados de acuerdo a cada aplicación, pero los datos físicamente para los usuarios se encuentran almacenados en un solo lugar. Una base de datos sirve para múltiple aplicaciones.

DBMS (Database Management System)

Sistema administrador de bases de datos (DBMS – Database management system) es simplemente el software que permite a una empresa centralizar sus datos, administrarlos eficientemente y proveer acceso a los datos almacenados por las diferentes aplicaciones. El DBMS actúa como una interfaz entre la aplicación y los archivos físicos de datos. Cuando una aplicación llama a un dato específico, la DBMS lo encuentra en la base de datos y lo presenta a la aplicación que lo requiere. Usando archivos tradicionales de datos, el programados puede especificar el tamaño y el formato de cada uno de los elementos del dato usado en la aplicación y después le dice a la computadora donde están ubicados.

El DBMS ayuda a que el programador o usuario final se despreocupen de la tarea de entender dónde y cómo están los datos almacenados por separado en la vista lógica y física de los datos.

La vista lógica presenta los datos como son percibidos por los usuarios finales o especialistas de negocios.

La vista física muestra como los datos están actualmente organizados y estructurados en el medio físico de almacenamiento. Hay una sola vista física de los datos, pero puede haber diferentes vistas lógicas. El software para la administración de bases de datos hace que este disponible la base de datos física para distintas vistas lógicas presentadas por varias aplicaciones.

Un DBMS tiene tres componentes:

1. Un lenguaje de definición de datos: es el lenguaje formal de programación usado para especificar el contenido y la estructura de la base de datos. Define cada elemento del dato como aparece en la base de datos antes de que el elemento del dato sea trasladado a las formas requeridas por la aplicación.

2. Un lenguaje de manipulación de datos: es usado en conjunto con algunos lenguajes de programación de aplicaciones convencionales para manipular los datos que están en la base de datos. Este lenguaje contiene comandos que permiten a los usuarios y especialistas de programación extraer datos de la base de datos para satisfacer sus

requerimientos de información y desarrollo de las aplicaciones. El lenguaje más utilizado para esto es el lenguaje estructurado de consulta (SQL – Structured Query Languaje). SQL puede ser utilizado por usuarios y especialistas en sistemas de información como un lenguaje de consulta interactivo para acceder a los datos de una base de datos, y los comandos SQL pueden ser incrustados en las aplicaciones escritas en lenguajes de programación convencionales.

3. Un diccionario de datos: es un archivo automático o manual que almacena las definiciones de los elementos y las características del dato como uso, representación física, quien debe dar mantenimiento al dato, autorizaciones y seguridad. Muchos diccionarios de datos pueden generar listas y reportes de datos por uso, agrupaciones, ubicación del programa, entre otros.

Definición del problema

Hasta este momento se han presentado tres empresas con productos muy similares, incluso utilizando casi la misma tecnología. Cada empresa ofrece diversas funcionalidades que se vuelven muy atractivas a nivel comercial o bien para grandes empresas. Las características comunes encontradas es la utilización de redes IP para la transmisión de la información, cada solución tiene algún tipo de manejo de compresión de la información; además cada empresa tiene sus propios desarrollos para soportar sus soluciones.

Aunque el software que se ha desarrollado cumple perfectamente las necesidades de las grandes empresas, quedan muy alejadas de las necesidades del individuo promedio, este nuevo mercado que está surgiendo por diversas situaciones quien desea poder adquirir un servicio de vigilancia para utilizarlo para actividades personales.

En el nuevo mercado de uso casi en su totalidad doméstico, podemos ver que las soluciones actuales están muy por encima de los requerimientos, se presentan servidores de video con capacidades para 16 cámaras o más cuando un individuo promedio tendrá a lo mucho 2 cámaras. En cuanto a los requerimientos de las cámaras, realmente se tiene uno solo, que capturen la información, la realización de poder ser utilizada en la oscuridad, es algo muy

especializado, o bien poder hacer acercamientos y alejamientos muy poderosos definitivamente está por encima de un uso convencional.

Ahora bien, además de tener equipos que serían subutilizados, los costos son verdaderamente altos, ya sea por licenciamiento o comprar toda la infraestructura eleva por mucho el presupuesto de cualquier usuario doméstico.

Después de haber visto las diferentes soluciones contrastadas contra los requerimientos de un individuo doméstico sin fines comerciales o soportando necesidades de empresas, se determinó la necesidad de generar una aplicación que pudiera utilizar dispositivos que fácilmente pudieran encontrarse en el mercado. Se determinó que la infraestructura más grande necesaria sería una PC Pentium II o superior al menos con 300Mhz y 64MB de memoria en RAM; una o varias "Web-cams". A medida que los requerimientos se fueran incrementando, entonces la aplicación deberá permitir crecer sin embargo esta escalabilidad deberá ser tan transparente y simplificada que no implique algún tipo de asistencia especializada y que pueda efectuarse por cualquier individuo.

Debido a las necesidades de monitoreo, la aplicación deberá almacenar la información capturada por las cámaras para que ésta pueda ser revisada de manera remota en una página de Internet. Dadas las restricciones que un usuario doméstico tiene sobre almacenamiento, cada secuencia de video deberá pasar por algún proceso de compresión para disminuir los requerimientos tanto

de ancho de banda al transmitir como también el espacio necesario para almacenar las secuencias.

Descripción de la aplicación

Una vez que las necesidades han sido detectadas, se decidió realizar una aplicación cuyo esquema se ve en la siguiente figura.

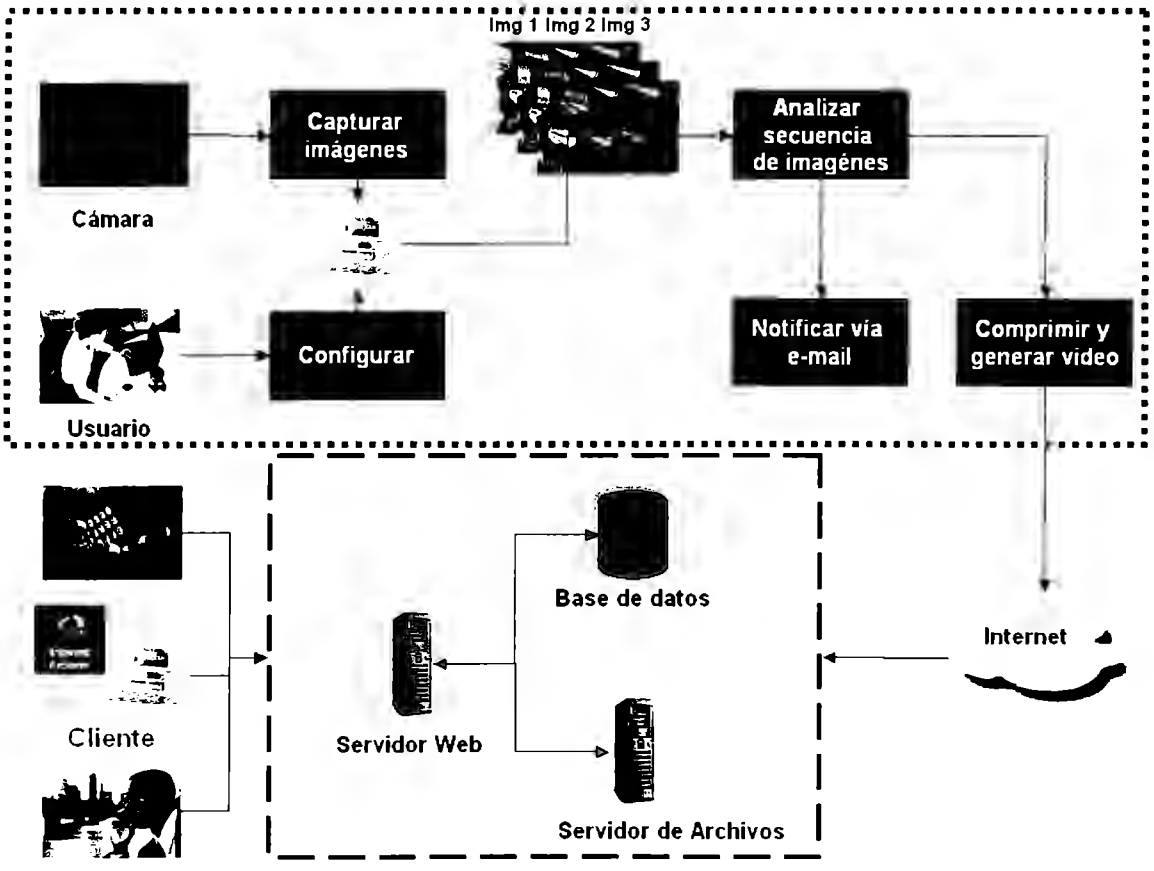


Diagrama 2 Descripción de aplicación

Se puede observar que la aplicación se encuentra dividida en dos niveles, por un lado se tiene la parte del cliente en donde principalmente las actividades realizadas son de captura y análisis. El otro nivel, se puede considerar la parte

del servidor de archivos, base de datos y Web para poder mostrar los resultados en una página de Internet.

Captura y configuración

En el diagrama 2 donde se describe la aplicación, se pueden observar dos puntos de inicio, por un lado la configuración de los parámetros que serán utilizados posteriormente por diversos módulos. El otro punto de inicio, es la cámara que se encontrará enviando pantallazos o fotografías por intervalos de tiempo determinados.

Análisis

Cada imagen capturada por la cámara será enviada al módulo de análisis en donde se revisarán individualmente para poder obtener información cuantitativa que servirá para posteriormente poder determinar si existieron cambios importantes en la secuencia global.

Se implementó un algoritmo basado en los errores promedios que cada imagen presenta comparado contra toda la secuencia que se seleccionó para poder definir de manera cuantitativa el grado de movimiento que se haya generado en toda la secuencia. Cuando se ha determinado el porcentaje total de error promedio de la secuencia, entonces la aplicación decide con base a previa configuración en la interfaz gráfica si vale la pena convertir la secuencia de

imágenes a un video, o bien, en caso de que no existan cambios relevantes solamente almacenar una imagen conteniendo así la información significativa para el rango de tiempo con el que se encuentre trabajando. El algoritmo desarrollado se encuentra basado en los cálculos del error cuadrático medio sobre todas las imágenes que forman parte de la secuencia.

Una vez realizado el análisis, entonces se determina si es necesario realizar una notificación por medio de correo electrónico. Las notificaciones por correo se llevan a cabo cuando los cálculos realizados en el análisis, se encuentran dentro del rango de notificación que solicita el usuario al configurar la aplicación. Para el envío del correo, solamente se solicita una cuenta de correo electrónico a donde las alertas serán enviadas. Los correos en su título indican el nivel de alerta que se excedió y en el cuerpo del correo la hora en que la alerta fue disparada.

Compresión y generación de video

Cuando ya se ha notificado al usuario en caso de haber sido necesario, entonces el siguiente paso es la realización del video, es decir, agrupar la secuencia de imágenes recibidas para poderlas convertir en una secuencia, es decir, hacer un video en formato "AVI" (Audio Video Interlaced). Dadas las características convencionales del formato ".avi", se presentó la necesidad de realizar una compresión para poder reducir el tamaño y disminuir los tiempos de transmisión así como el espacio de almacenamiento. La compresión se hace

utilizando el "codec" MSVC de Microsoft, ya que presentó la mejor compresión tamaño – calidad de los videos.

Almacenamiento

El siguiente paso a realizar dentro de la aplicación es el almacenamiento de los videos conforme se van generando. La aplicación almacena los videos en un servidor de archivos al cual en el momento del desarrollo del presente documento, la conexión es por medio del protocolo de Windows, en donde se tiene mapeada en una unidad de red la ruta del servidor destino para la recepción de los archivos.

En la parte de almacenamiento, se guarda una referencia en una base de datos implementada con MS SQL Server en donde se guardan las fechas de generación del video, así como algunos datos relevantes del análisis y el nombre del almacenado en el servidor de archivos.

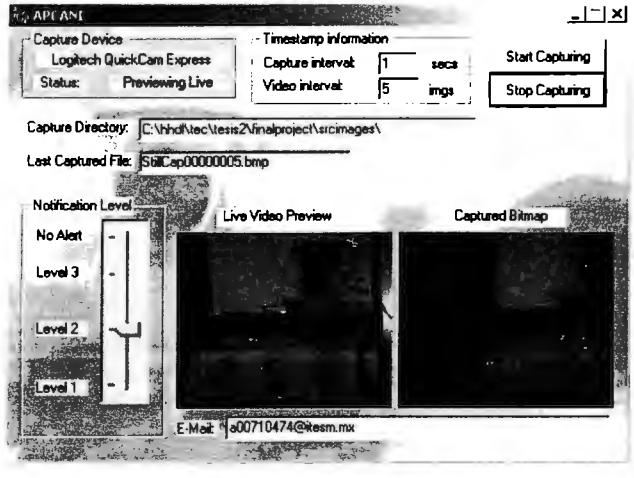
Tanto el servidor de archivos como la base de datos se encuentran conectados con el servidor de Web para que se puedan realizar las búsquedas y mostrar los videos en la página de Internet.

Servidor de Web

La consulta de los videos, se hace por medio de una página de Internet cuya principal función es la realización de consultas por rangos de fechas y tiempos para poder realizar búsquedas puntuales sobre los videos que la aplicación de captura y análisis generado. Los resultados de las búsquedas en la página de consulta, son mostrados en forma de lista desplegando el tiempo capturado en cada video, así como una liga al video para poder comenzar su ejecución automática. La ejecución de cada video se hace utilizando el control "Active Movie" de Microsoft para el despliegue de videos lo cual permite al usuario manipular el desplegado ofreciéndole la capacidad de adelantar o atrasar la secuencia, detenerla o bien poner pausa en algún momento deseado. En la página de Internet, también se pueden realizar algunas opciones de administración (altas, bajas y cambios) de usuarios y cámaras para el caso en que la aplicación funcione como repositorio central de videos, entonces cada usuario podrá ver únicamente las cámaras que tenga asignadas y no los videos generados por otras cámaras.

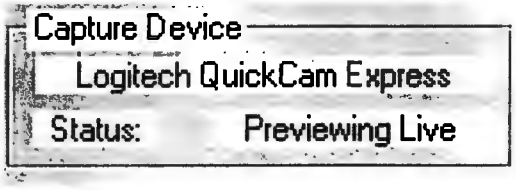
Implementación de APCANI

1. APCANI (Aplicación de captura y análisis de imágenes)



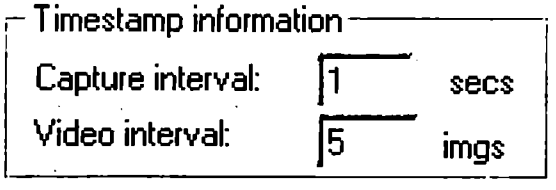
Como su nombre lo indica, las principales funciones de esta aplicación son la captura de imágenes y el análisis de las mismas. Visualmente la aplicación se encuentra dividida horizontalmente en 4 secciones, comenzando de la parte superior hacia la inferior, lo primero que se despliega es la información del dispositivo de video que se está utilizando, así como los tiempos con los que la aplicación realizará las capturas, en la parte de en medio se pueden observar 2 cuadros con imágenes, por un lado es la imagen en tiempo real, y por otro la imagen que en ese momento se está capturando. Finalmente, las 2 secciones inferiores cuentan con información sobre la ruta de almacenamiento y los niveles de configuración para las alertas.

Cada una de las secciones previamente especificadas se ve de la siguiente manera.

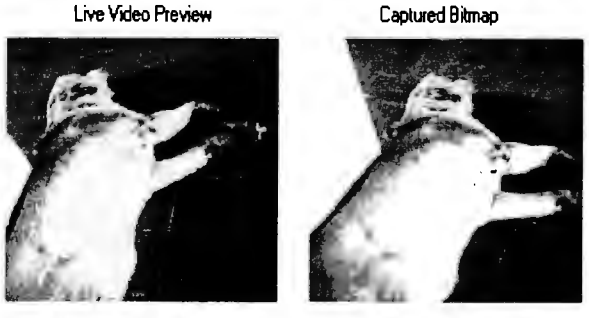


La información que esta imagen se encuentra desplegando es la de "Logitech QuickCam Express", es decir el nombre común del dispositivo que estará realizando la captura de la información. En la parte de "status" indica "Previewing Live Video" ya que la columna del lado izquierdo se utiliza principalmente para mostrar lo que la cámara está capturando.

Esta sección muestra principalmente el intervalo en que se estarán tomando muestras (imágenes) que serán utilizadas para la generación del video en caso de que sea necesario hacer uno. De esta manera, el usuario define si quiere estar obteniendo muestras hasta un segundo como tiempo mínimo de frecuencia. En la gráfica el tiempo de captura es de 1 segundo, mientras que la secuencia de imágenes se realiza cada 5 imágenes capturadas.

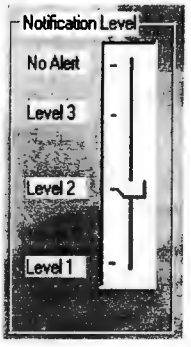


Estos dos cuadros muestran por un lado en tiempo real el objeto al cual la cámara se encuentra enfocada y en la parte derecha, la última imagen que fue capturada para formar parte de una secuencia.

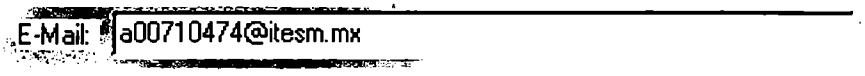


En este campo, el usuario proporciona la ruta completa donde se estarán almacenando las imágenes para que posteriormente sean analizadas y procesadas.

Tecnológico de Monterrey, Campus Ciudad de México
Biblioteca



Finalmente se tiene la configuración del nivel de alerta que será utilizado para las notificaciones. Se pueden configurar 4 niveles, el nivel donde no se requieren alertas, hasta el nivel 3 dependiendo el grado de alerta que desee. Para los avisos de las alertas, es necesario el envío de correos electrónicos, por lo tanto se debe especificar el correo electrónico a donde se enviarán las alertas.



APCANI se programó utilizando Visual C++ 6.0 y Direct X 9.0 debido a su alta integración para el desarrollo de aplicaciones multimedia. En el desarrollo de la aplicación se tomó como base el ejemplo de "StillCap" de Direct X, el cual tiene como principal función la captura de imágenes ("screen shots") y la captura de video. Debido a los requerimientos de esta tesis, se eliminó de la implementación del ejemplo la generación de videos, dejando únicamente la capacidad para realizar capturas de imágenes, acciones similares a una cámara de video digital. Se aprovechó la capacidad del ejemplo para generar imágenes .bmp (Bitmap) las cuales son útiles para el desarrollo completo del APCANI.

APCANI es una aplicación basada en diálogos la cual consiste en las siguientes clases:

CAboutDlg	Encargada principalmente de mostrar el diálogo con la información que aparece cuando un usuario selecciona la opción de "Acerca".
CAppDatabase	Encargada principalmente de la interacción con la base de datos. Utilizada para insertar en la base de datos las entradas necesarias sobre cada video que haya sido desarrollado y al mismo tiempo obtener las cámaras existentes en el sistema para asociarlas a la captura.
CBitmap2Avi	Su principal tarea como su nombre lo indica es realizar las acciones de conversión de imágenes en formato .bmp a el video ".avi"
CCapThread	Provee del ambiente necesario para la realización de "threads" de procesamiento para el análisis, conversión y almacenamiento. Su objetivo es que las acciones descritas anteriormente no interfieran en tiempo de ejecución con la captura
CImageAnalyzer	Realiza todas las acciones necesarias para el análisis de las imágenes, desde la lectura de cada imagen por secuencia, su descomposición y cálculos de estadísticos.
CSampleGrabberCB	Utilizada para la captura en tiempo real e imágenes en formato ".bmp"
CStillCapApp	Clase de la aplicación APCANI
CStillCapDlg	Contiene todos los métodos y variables necesarios para el despliegue, configuración y manejo de la aplicación.

Tabla 6 Listado de Clases APCANI

Para fines del actual documento, se considera pertinente especificar con más detalle las siguientes clases ya que contienen gran parte de la lógica de la aplicación.

1.1 CBitmap2Avi

Esta clase es instanciada una vez que se cumpla el intervalo especificado para captura de secuencia. Esta clase parte de la premisa que las imágenes fueron previamente almacenadas en un directorio y se encuentran listas para comenzar el proceso.

Debido a que el proceso de captura no puede ser interrumpido por ninguna actividad de procesamiento, entonces es necesario utilizar un "thread" (hilo de ejecución) y precisamente se implementó un concepto denominado "working-thread" para que la clase CBitmap2Avi pudiera realizar el análisis sin intervenir a la captura.

La lógica de uso para esta clase radica en 2 métodos principalmente:

1. Convert()
2. CompressBitmap

1.2 CBitmap2Avi::Convert

Lo primero que este método debe realizar es la parte de análisis del error de cada imagen así como los valores promedio de la secuencia

```
CImageAnalyzer *pIAnlyzr = new CImageAnalyzer(szWorkingDirectory, m_lfStartValue,
m_lfEndValue);
        pIAnlyzr->AnalyzeWorkingDirectory();
```

```
if(pIAnlyzr->GetError() == "" && pIAnlyzr->GetImagesNumber() > 0)
{
        pIAnlyzr->CalculateRMSAverageNMax();
        pIAnlyzr->CalculateABSAvgMax();
        pIAnlyzr->CalculatePercentageValue();
        nbImages = pIAnlyzr->GetImagesNumber();
                . . .
```

Una vez que se cuenta con la información cuantitativa de la secuencia es necesario generar el archivo "AVI" que contendrá la secuencia de imágenes

```
create_avi_file( (char *)aviFileName, nbImages, dx, dy );
```

Finalmente comienza una iteración para todas las imágenes que vienen en la secuencia para realizar la compresión de las imágenes y es donde se manda llamar al método de compresión para cada imagen.

```
for(i=m_lfStartValue; i<m_lfEndValue; i++)
{
        ...
        f = open_file(szBuff);
        if( f == NULL )
        {
                . . .
        )
        else
        {
                bResult = CompressBitmap(f, nCounter);
        }
```

1.3 CBitmap2Avi::CompressBitmap

La responsabilidad de este método es comprimir la imagen y agregarla al archivo de video que fue previamente generado. Es importante destacar en este método el uso del objeto de tipo AVICOMPRESSOPTIONS ya que en él se puede determinar el tipo de "codec" a utilizar para la compresión. En el código presentado a continuación se presenta el "MSVC". Al ser utilizado con la función AVIMakeCompressedStream, permitió comprimir los archivos de su tamaño original, aproximadamente 11MB para 5 segundos de grabación a solamente

50kb lo cual fue un avance importante en cuanto a los niveles de almacenamiento, sobre todo porque es necesario almacenar los videos de manera remota mejorando notablemente el desempeño de la aplicación.

```
BOOL CBitmap2Avi::CompressBitmap(FILE * fBMPFile, int nFrame)
{
        . . .
       pbmih = getBitmap( fBMPFile, nFrame, &fsize, &s );
          . . .
               if(nFrame ==0)
               {
                       AVICOMPRESSOPTIONS opts;
                       ZeroMemory(&opts,sizeof(opts));
                       opts.fccHandler = mmioFOURCC('M','S','V','C');
                       opts.fccType = streamtypeVIDEO;
                       opts.dwQuality = 5000;
                       opts.dwFlags = AVICOMPRESSF_KEYFRAMES|AVICOMPRESSF_VALID;

                       HRESULT hr = AVIMakeCompressedStream( . . .);
                       switch(hr)
                       {
                              . . .
                              case (AVIERR_OK):
                                     if( AVIStreamSetFormat(. . . )
                                     {
                                            . . .
                                     }
                                     else
                                     {
                                            if( AVIStreamWrite( . . . ) != 0 )
                                            {
                                            err_exit( "Can not comrpessed AVI file." );
                                            }
                                            else
                                                   bResult = TRUE;
                                     }
                                     break;

                       }
                       . . .
               }
               else
               {
                       if( AVIStreamWrite( . . . ) != 0 )
                       {
                              . . .
                       }

}
```

1.4 CImageAnalyzer

La principal responsabilidad de esta clase es encontrar el eRMS "Root Mean Square Error" para la secuencia, también encontrar el promedio de eRMS, así como el máximo valor encontrado en la secuencia.

Los principales métodos utilizados en esta secuencia son:

```
AnalyzeWorkingDirectory();
CalculateRMSAverageNMax();
CalculateABSAvgMax();
CalculatePercentageValue();
```

El primer método, "AnalyzeWorkingDirectory", es el más importante ya que contiene la implementación de la fórmula de eRMS, error cuadrático medio explicado en el capítulo de fundamentos teóricos. La fórmula es:

$$e_{rms} = \left[\frac{1}{MN} \sum_{x=0}^{M-1} \sum_{y=0}^{N-1} [\hat{f}(x,y) - f(x,y)]^2 \right]^{1/2}$$

Una vez que se tiene eRMS para cada imagen, se obtiene el promedio y el máximo permitido, de esta manera se hace el proceso de discriminación con base en el porcentaje que se haya especificado en los niveles de notificación.

1.5 Diagrama de clases

A continuación se presentan los diagramas de clase de aquellas que se encuentran directamente ligadas con el proceso principal de captura y análisis de APCANI.

CAppDatabase
m_strConnection:CString = "" m_strUserId: CString = "" m_strPassword: CString = ""

m_bstrSource: _bstr_t = “” m_bstrErrorDescription: _bstr_t = “” m_lErrorCode: long m_strErrorMeaning: CString
public CAppDatabase(): void AddVideo(bstrFileName, CameraId, dtStardDate, dtEndDate, lfMaxErrorPercentage, eRMSMaxValue, lfImagesNumber, eRMSMaxErrorPercentage, nAlertPercentage): BOOL ListCameras(): _RecordsetPtr GetErrorSource(): _bstr_t GetErrorDescription(): _bstr_t GetErrorCode(): _bstr_t GetErrorMeaning: _bstr_t private SetErrorInfo(_com_error): void PrintProviderErro(_ConnectionPtr): void

Diagrama 3 Clase CAppDatabase

CBitmap2Avi
interlace:int = 0 deleteAfter: int = 0 framesperSecond: int = 1 aviFileName: char[] szPrefix: char[] nbImages: int = 0 dx: int = 0 dy: int = 0 firstImg: int = 0 m_nAlertPercentage: int = 0 m_lfStartValue: long = 0 m_lfEndValue: long = 0 pDlg: CStillCapDlg = NULL m_strAviFileName: CString = “” aviFile: PAVIFILE = NULL pstream: PAVISTREAM = NULL psCompressedStream = NULL dtStartDate: CTime dtEndDate: CTime m_strError: CString
err_exit(char *): void create_avi_file(name, nbImages, dx,dy): void delete_file(prefix, num): void delete_file(szfileName): void open_file(prefix, num,totalimages): FILE * open_file(prefix): FILE * getBitmap(f, frame, fsize, s):LBITMAPINFOHEADER Convert():void geWorkingDirectory(szPath, szPrefix): char * CompressBitmap(FILE *, frame): BOOL GetError(): CString

Diagrama 4 Clase CBitmap2Avi

CCapThread
m_hwndNotifyRecalcDone:HWND m_hEventStartRecalc: HANDLE m_hEventRecalcDone: HANDLE m_hEventKillRecalcThread: HANDLE m_hEventRecalcThreadKilled: HANDLE m_nRecalcSpeedSeconds: int m_hwndNotifyProgress: HWND
StartThreadProc(pParam): UINT

Diagrama 5 Clase CCapThread

CImageAnalyzer
m_szWorkingDir: char[] m_Image1Data: char* = NULL m_arrRMS: float[] = NULL m_ABSRMS: float[] = NULL m_lmageIndex: long = 0 m_ldeRMSAverage: float = 0 m_ldeRMSMax: float = 0 m_ldMaxErrorPercentage: float = 0 m_ldABSAvgMax: float = 0 m_lfStartValue: long = 0 m_lfEndValue: long = 0 m_strError: CString = ""
public GeteRMSAverageValue(): float GeteRMSMaxValue(): float GetPercentageValue(): float GetDistance(): float GetImagesNumber(): float GeteRMS4Image(nImage): float AnalyzeWorkingDirectory(): void CImageAnalyzer(): CImageAnalyzer(szPath, start, end): CalculateRMSAverageNMax():void CalculateABSAvgMax(): void CalculatePercentageValue(): void GetError(): CString ShowLastError(bShowAlert): void protected LoadBMP(filename, x, y, d): unsigned char SaveIntensity4File(imagename, x, y, size): BOOL getint(fileptr): unsigned int getshort(fileptr): insigned int SaveTGA(filename, x, y, d): void CalculateRMS(filename, x, y, d): BOOL

Diagrama 6 Clase CImageAnalyzer

2. Base de datos

APCANI requiere para su funcionamiento una base de datos relacional, aunque podría ser utilizado un documento en texto plano o bien un archivo XML que almacene la información. La base de datos ayudó al despliegue de resultados que requerían de la realización de búsquedas por rangos de fechas así como la rapidez con la que los resultados se podrían desplegar a diferencia de los archivos planos o XML

Las tablas y los campos utilizados para el despliegue de los resultados así como también para la administración de la página de Internet son los siguientes.

Nota: Los tipos de datos mostrados en esta tabla corresponden a una base de datos de Microsoft SQL Server 2000

Nombre	**Descripción**		
	Campo	**Tipo**	**Tamaño**
tbCamera	Contiene cada una de las cámaras que estarán dadas de alta en el sistema para posteriormente ser asignadas a un usuario.		
	CameraId	int	4
	Label	varchar	50
	Description	varchar	400
tbProfile	Utilizada para el manejo de los perfiles en la página, aunque actualmente sólo se maneja el de administrador y de consultas		
	ProfileId	int	4
	Name	varchar	100
tbUser	Lista de usuarios que tienen acceso a la página de Internet		
	UserName	varchar	10
	Password	varchar	10
	FirstName	varchar	50
	LastName	varchar	50

	EMail	varchar	100
	ProfileId	int	4
tbUserCamera	Relación entre cámaras y usuarios		
	UserName	Varchar	10
	CameraId	Int	4
	StatusId	Int	4
tbVideo	**La tabla más importante y medular en las consultas del sistema ya que contiene la información sobre el análisis y las referencias de los videos que se generaron**		
	VideoId	Int	4
	GeneratedDate	datetime	8
	StartCapturedDate	datetime	8
	EndCapturedDate	datetime	8
	FullFileName	varchar	400
	CameraId	int	4
	eRMSAverage	float	8
	IsProduction	bit	1
	BackupFile	varchar	400
	eRMSMaxValue	float	8
	ImagesNumber	float	8
	MaxErrorPercentage	float	8
	AlertPercentage	int	4

Tabla 7 Listado de tablas de la base de datos.

A continuación se presenta la lista de procedimientos almacenados utilizados para el desarrollo de la primera versión de APCANI. Estos procedimientos, se encuentran completamente vinculados a la implementación realizada con Microsoft SQL Server. La nomenclatura utilizada es la siguiente:

spI -> Procedimientos encargados de hacer internamente algún tipo de "Insert" en la base de datos

spU -> Procedimientos encargados de actualizaciones, "Update"

spD -> Procedimientos encargados de eliminar de la base de datos "Delete"

spS -> Generalmente son "selects" y se utilizaron para la realización de las consultas de los videos.

Nombre	Descripción		
	Parámetros	**Tipo**	**Tamaño**
spD_Camera			
	@CameraId	Int	4
	@Result	bit	1
spD_UnAssignCamera	Quitar la asignación de una cámara al usuario especificado		
	@UserName	varchar	10
	@CameraId	int	4
	@Result	bit	1
spD_User			
	@UserName	varchar	10
	@Result	bit	1
spD_Video			
	@VideoId	int	4
	@Result	bit	1
	@ResMessage	varchar	400
spI_AssignUserCamera	Asignar cámara a un usuario para poder consultar los videos de la misma.		
	@UserName	varchar	10
	@CameraId	int	4
	@StatusId	int	4
	@Result	bit	1
spI_Camera			
	@Label	varchar	50
	@Description	varchar	400
	@Result	bit	1
	@ResMessage	varchar	400
spI_User			
	@UserName	varchar	10
	@Password	varchar	10
	@FirstName	varchar	50
	@LastName	varchar	50
	@EMail	varchar	100
	@Result	bit	1
	@ResMessage	varchar	100
spI_Video			
	@FileName	varchar	400
	@CameraId	int	4
	@StartCapturedDate	datetime	8
	@EndCapturedDate	datetime	8
	@ErrorPercentage	float	8
	@eRMSMax	float	8
	@ImagesCounter	float	8
	@MaxErrorPercentage	float	8
	@AlertPercentage	int	4
	@Result	bit	1
	@ResMessage	varchar	400
spS_CanLogin			
	@UserName	varchar	10
	@Password	varchar	10
	@FirstName	varchar	50

	@LastName	varchar	50
	@EMail	varchar	100
	@ProfileId	int	4
	@Result	bit	1
spS_GetCameraInfo			
	@CameraId	int	4
	@Label	varchar	50
	@Description	varchar	400
spS_GetUserInfo			
	@UserName	varchar	10
	@FirstName	varchar	50
	@LastName	varchar	50
	@EMail	varchar	100
	@ProfileId	int	4
	@Password	varchar	10
sps_GetVideos4Camera	Lista todos los videos para la cámara especificada		
	@CameraId	int	4
	@PageSize	int	4
	@SelPage	int	4
	@Asc	bit	1
spS_ListCameras	Lista todas las cámaras existents en el sistema		
spS_ListCameras4User	Lista las cámaras del usuario especificado		
	@UserName	Varchar	10
spS_ListUsers			
spS_SearchVideos			
	@CameraId	int	4
	@StartDate	datetime	8
	@EndDate	datetime	8
spS_SearchVideosGenerated			
	@CameraId	int	4
	@GenDate	datetime	8
	@EndDate	datetime	8
	@Asc	bit	1
	@PageSize	int	4
	@SelPage	int	4
spS_SearchVideosGenerated4Percentage	**Procedimiento más utilizado ya que lista los videos para la fecha y nivel de notificación especificado.**		
	@CameraId	int	4
	@GenDate	datetime	8
	@EndDate	datetime	8
	@Asc	bit	1
	@PageSize	int	4
	@SelPage	int	4
	@AlertPercentage	int	4
spU_Camera			
	@CameraId	int	4
	@Label	varchar	50
	@Description	varchar	400
	@Result	bit	1
spU_ChangePassword			
	@UserName	varchar	10
	@CurrPassword	varchar	10
	@NewPassword	varchar	10
	@Result	bit	1

	@ResMessage	varchar	400
spU_User			
	@UserName	varchar	10
	@Password	varchar	10
	@FirstName	varchar	50
	@LastName	varchar	50
	@EMail	varchar	100
	@Result	bit	1

Tabla 8 Listado de procedimientos almacenados

La relación entre las tablas se puede ver en la siguiente gráfica.

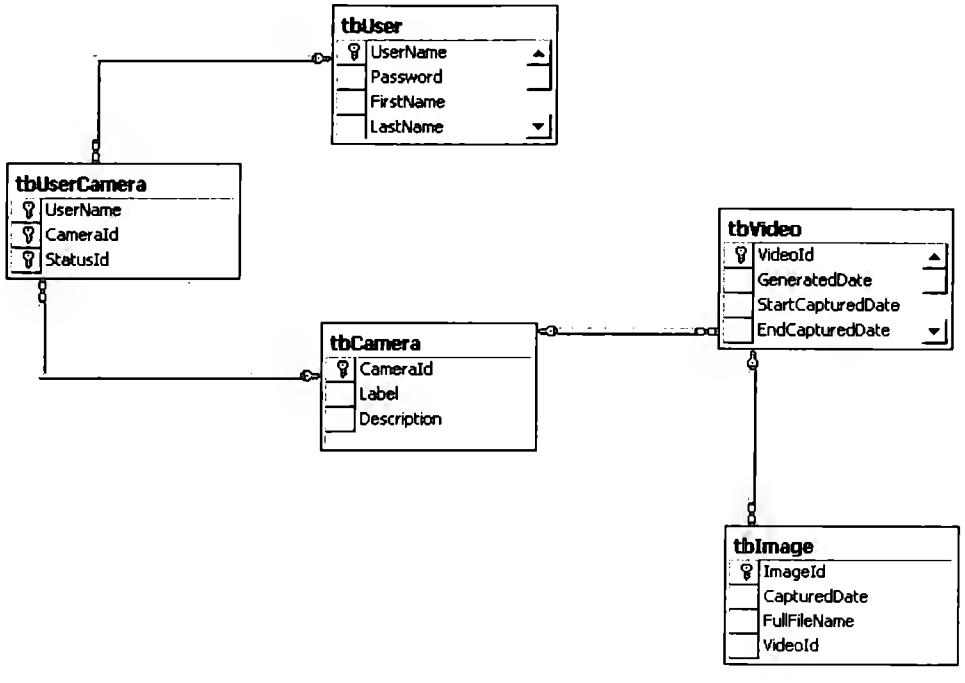


Diagrama 7 Tablas de base de datos APCANI

Para el desarrollo del sistema se utilizó una base de datos, sin embargo también se contempló la utilización de archivos XML para el almacenamiento ya que de esta manera se podría eliminar el uso de un manejador de bases de datos.

3. Página de Internet

El desarrollo de la página se hizo con páginas ASPs, esto por la facilidad de su implementación y acceso ya que es muy común encontrar computadoras con Windows. Además, tanto la versión de Profesional, Server y Advanced Server de este sistema operativo, cuentan con un servidor de páginas Web conocido como "Internet Information Server". También, existe una versión simplificada de este servidor de páginas llamado "Personal Web Server".

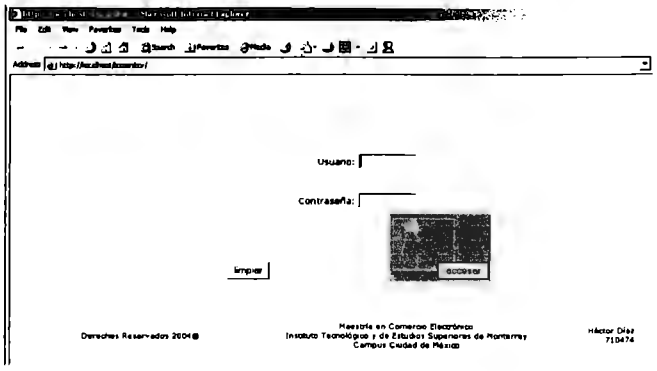
En general, se cuenta únicamente con 2 tipos de páginas, por un lado se tienen las de administración de recursos como serían, cámaras o usuarios, y por otro lado las páginas de consultas.

Las páginas que se generaron se describen a continuación:

Esquema	Directorio	Descripción
admin cameras.asp default.asp procCameras.asp procUsers.asp users.asp	admin	Contiene todos los archivos para la administración (ABC) de la página, tanto para los usuarios como para las cámaras. Este tipo de páginas solamente pueden ser vistas por el perfil "Administrador"
app default.asp	app	Página por omisión de inicio una vez que el usuario ha ingresado a la página.
images includes calendar camera_lib.asp camusers_lib.asp cmonitor.css consts.asp footer.html header.asp mail_lib.asp query_leftmenu.asp users_lib.asp videos_lib.asp	includes	Directorio más importante ya que contiene la lógica de negocio de la página. Aquí se encuentran algunas librerías / funciones que son utilizadas por otras páginas para su funcionamiento. Generalmente cada uno de los archivos "_lib.asp" internamente hace referencia a la base de datos por medio de procedimientos almacenados. Estos archivos son la única interacción explícita con la base de datos que existe en la página; ningún archivo fuera de este directorio hace referencia a la base de datos.
query all.asp default.asp listalerts.asp lists.asp showvideo.asp templates default.asp	query	Aquí podemos encontrar toda la lista de páginas que permiten la realización de la búsqueda de videos con los diferentes criterios especificados.

Tabla 9 Descripción esquema página de Internet

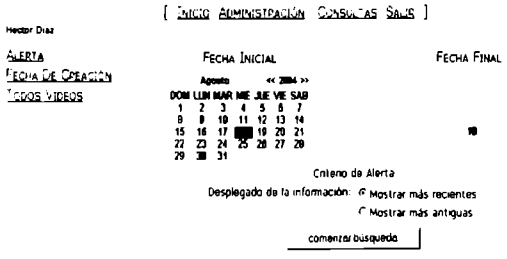
El proceso de interacción con la página de Internet se puede resumir en 4 pasos los cuales se describen a continuación.



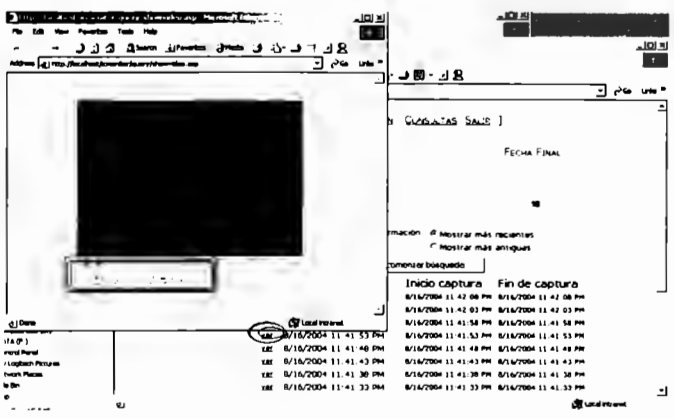
1. Cada usuario deberá autenticarse en una página de inicio, el objetivo de esto es que los usuarios podrán buscar videos únicamente de las cámaras que tienen registradas.

2. Una vez que el usuario se autenticó, entra a la página de inicio donde podrá seleccionar de un menú la opción deseada. Al momento de la realización de este documento solamente se puede ingresar a una sección pública y en caso de que el usuario tenga perfil de administrador, poder entrar a la sección de "Administración".

3. Cuando el usuario de sea realizar algún tipo de consulta, la página que aparece permite la selección de un rango de fecha por medio de unos calendarios. Además de la fecha, se puede seleccionar el nivel de alerta de información que se desea obtener. Las búsquedas pueden ser ordenadas por tiempo, mostrando las más recientes o las más antiguas al principio.



4. Por último, los resultados aparecen en forma de lista con una liga en cada video para que pueda ser seleccionado y se muestre el mismo en una página mostrándolo únicamente.



Calibración

Una vez desarrollada la aplicación, fue necesario realizar ciertas corridas con diferentes escenarios para poder definir de manera cuantitativa los rangos de movimientos permisibles para la aplicación.

Durante el proceso de análisis de cada secuencia de imágenes, APCANI realiza ciertas operaciones sobre cada imagen, aunque el proceso de análisis se mencionó con la descripción de APCANI, se considera importante retomarlo para poder entender el origen de las gráficas que se mostrarán en las siguientes páginas.

El criterio de análisis utilizado para el presente documento se basa en los denominados "criterios de fidelidad" [González, 1987] en donde se busca expresar la pérdida de información como función de las imágenes de entrada. Para este análisis se utilizó el error cuadrático medio entre una imagen de entrada y una de salida.

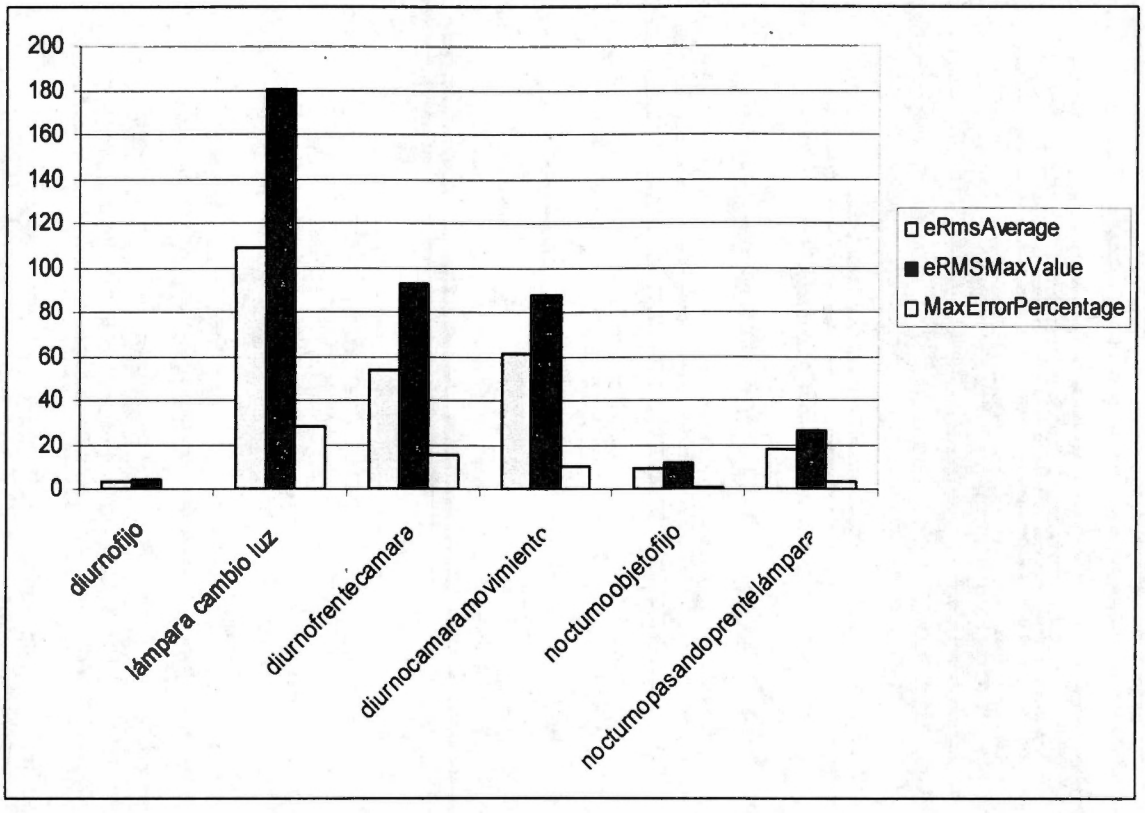
Para calcular el error cuadrático medio del total de las imágenes, fue necesario utilizar la fórmula del eRMS entre f(x,y) (imagen original) y f'(x,y) (imagen entrante de la secuencia). Prácticamente hablando la lógica que se siguió, fue encontrar la suma de todas las imágenes comparadas contra la imagen anterior para poder ir viendo los cambios o evolución que vaya surgiendo en la secuencia.

En la captura de secuencias, se calculan tres valores los cuales son:

- eRMSAverage: Consiste en sacar el promedio de eRMS de todas las imágenes que formaron parte de la secuencia.
- eRMSMax: Es el eRMS máximo de la secuencia de imágenes en valor absoluto para solamente manejar los rangos más altos positivos.
- MaxErrorPercentage: Se refiere al porcentaje de cambio en la intensidad de las imágenes comparado contra el 100% de la intensidad máxima generada por una imagen con mucha luz.

Para el análisis se realizaron diversos escenarios de los cuales sus resultados se mostrarán posteriormente, sin embargo, en este momento se describirá el comportamiento en conjunto de todos ellos. La siguiente gráfica engloba todos los escenarios y muestra los resultados obtenidos para el eRMSAverage, eRMSMax y MaxErroPercentage.

Los valores que esta gráfica muestra son valores promedio de diferentes muestras para cada valor.



Gráfica 1 Análisis de escenarios.

A continuación se analizará cada uno de los valores:

1. eRMSAverage

El error cuadrático medio promedio de cada imagen nos da una muestra sobre la cantidad de cambio existente en las secuencias de imágenes. Los valores para objetos fijos tanto diurnos como nocturnos son muy pequeños. Lo anterior se debe a que la imagen siempre es la misma. Es en el experimento de cambio de luz donde se nota que el valor promedio aumenta considerablemente; incluso se da el valor más alto de todos los experimentos, y este pico se debe a que la imagen es completamente polarizada, por un lado tiene exceso de luz y por otro

repentinamente carece de la misma. En donde hubo una gran diferencia de valores fue en los experimentos en donde se pasaba un objeto frente a la cámara ya que los valores del experimento diurno son mayores que el nocturno. Lo anterior se debe a que en el experimento nocturno, la cámara no podía enfocar mucha información, es decir, grababa casi pura oscuridad, entonces los movimientos frente a ella resultaron casi imperceptibles por lo que no había cambios en su cálculo de error.

Por último los valores que resultaron muy similares fueron los del experimento de movimiento tanto frente a la cámara como el de mover la cámara de un lugar a otro. Esto quiere decir que prácticamente los cambios repentinos de movimiento que puede capturar APCANI en su nivel promedio máximo se pueden encontrar con un eRMSAverage alrededor del 60.

2. eRMSMaxValue

El comportamiento del valor máximo del eRMS va muy de la mano con el valor eRMSAverage, sin embargo aquí podemos ver los valores en intensidad de rango entre 0 y 255. Donde 0 implica cambios de imagen casi nulos y el 255 implica cambios muy drásticos como el valor del experimento de prender y apagar la luz que tuvo un valor máximo de 180.

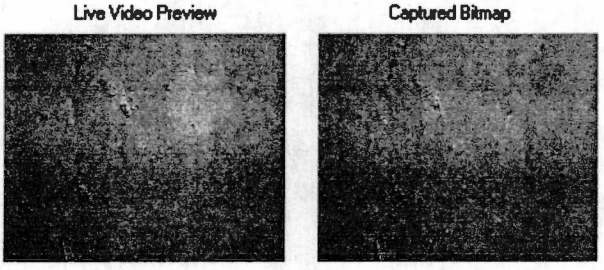
3. MaxErrorPercentage

En cuanto al valor promedio máximo obtenido contra los cambios bruscos, en la gráfica se puede observar que el 40% casi es el valor más alto que una

secuencia de imágenes puede obtener; mientras que las secuencias de los demás experimentos se encuentran en un rango menor al 20%.

4. Escenario diurno con objeto fijo

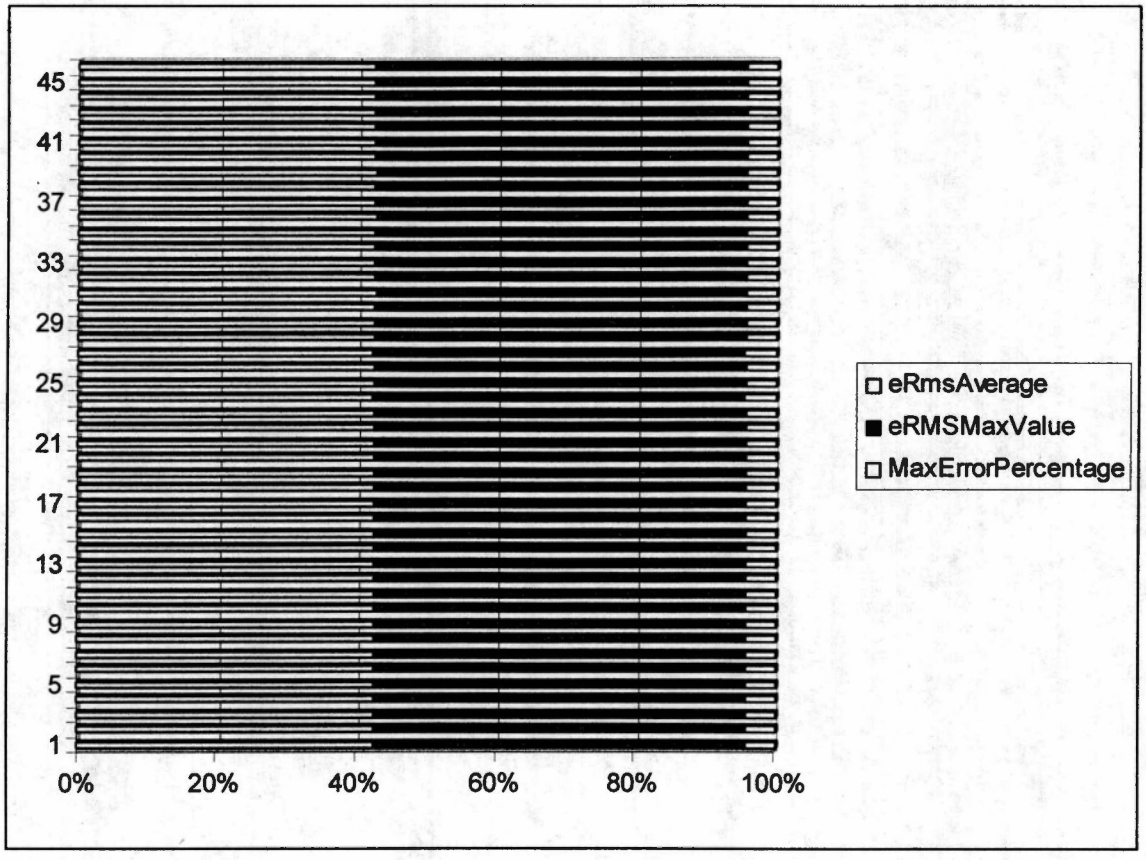
4.1 Descripción



Se colocó la cámara a plena luz del día grabando un objeto sin movimiento (una pared) como se muestra en la imagen.

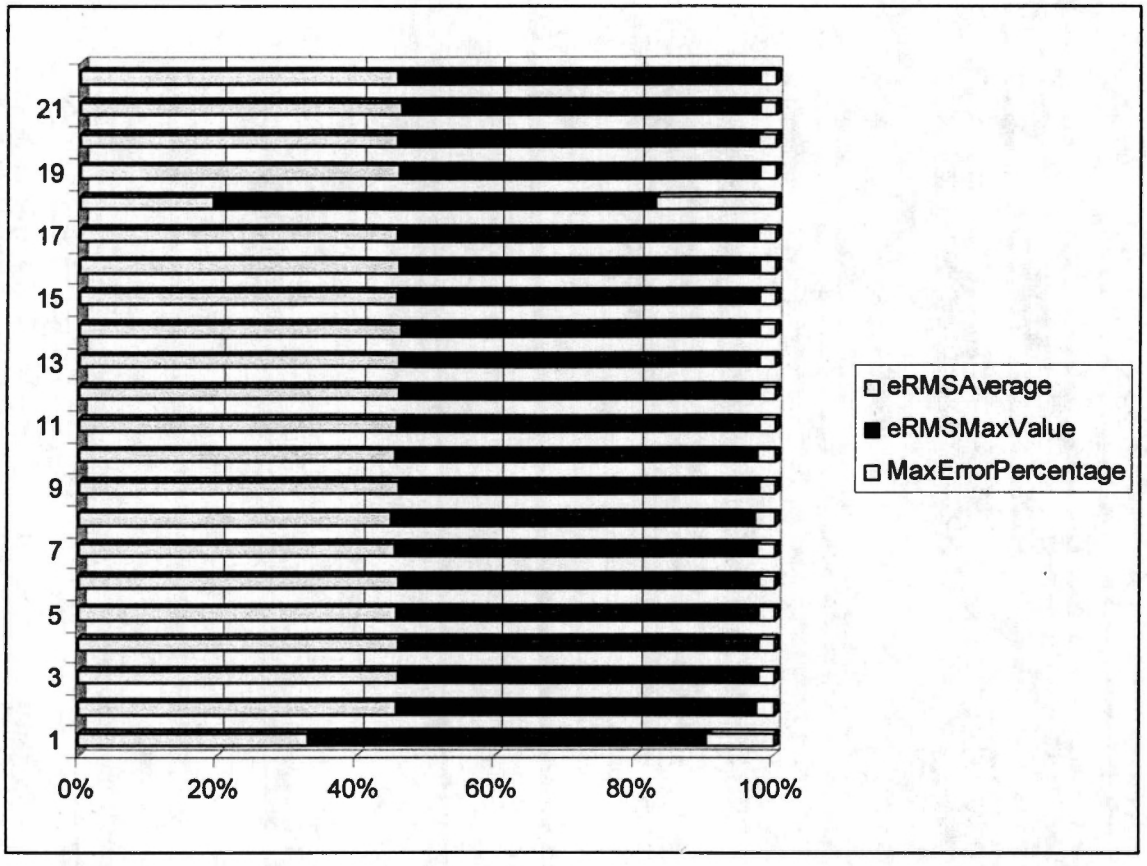
4.2 Análisis

El lapso de tiempo entre captura de imágenes fue de 1 segundo y las secuencias fueron generadas cada 5 imágenes.



Gráfica 2 Escenario diurno con objeto fijo.

Mismo experimento, sin embargo ahora se aumentó el intervalo para la generación de las secuencias a 10 segundos.

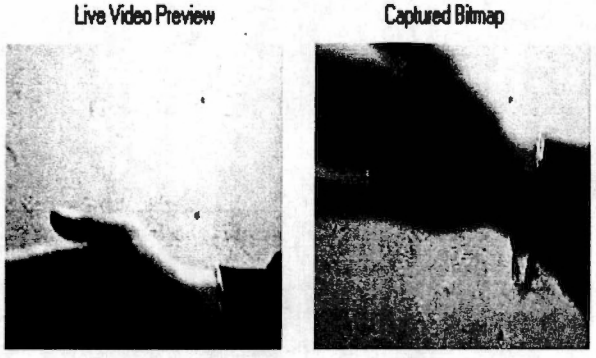


Gráfica 3 Escenario diurno con objeto fijo en 10 segs.

Al ver las gráficas anteriores, se puede observar que no hay relación entre los intervalos de captura cuando los objetos se encuentran fijos, el comportamiento de los 3 indicadores es el mismo además de encontrar un comportamiento completamente uniforme para cada secuencia analizada.

5. Escenario diurno con objeto pasando frente a la cámara

5.1 Descripción

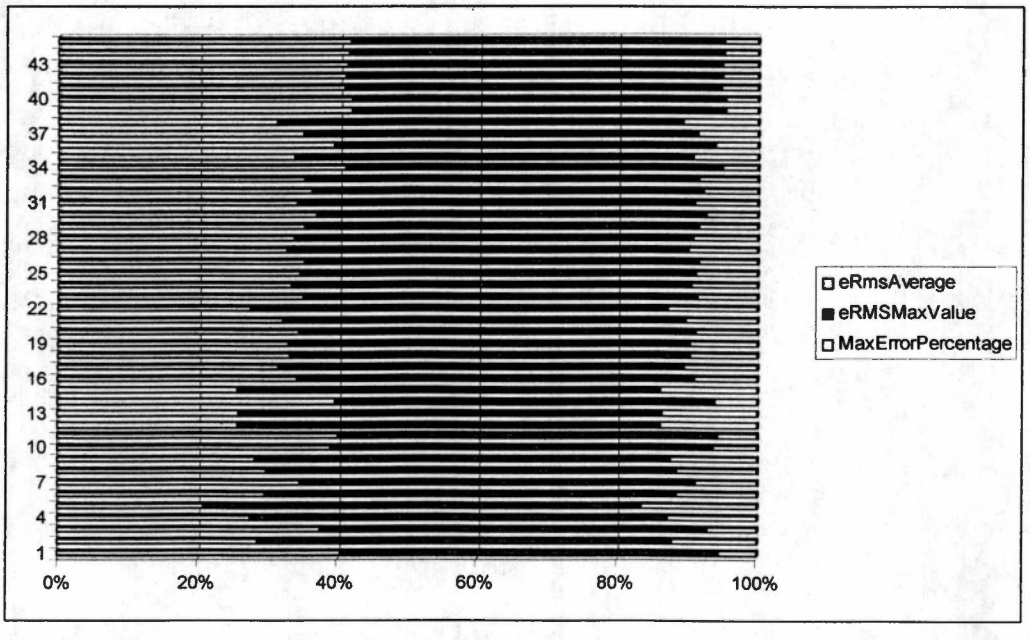


Grabando frente a una pared y un objeto se pasó frente a la cámara. El objeto en algunas ocasiones tapó por todo el ángulo de visión de la cámara.

Lo anterior ocasiona que existan movimientos polarizadós en la intensidad de todos los píxeles.

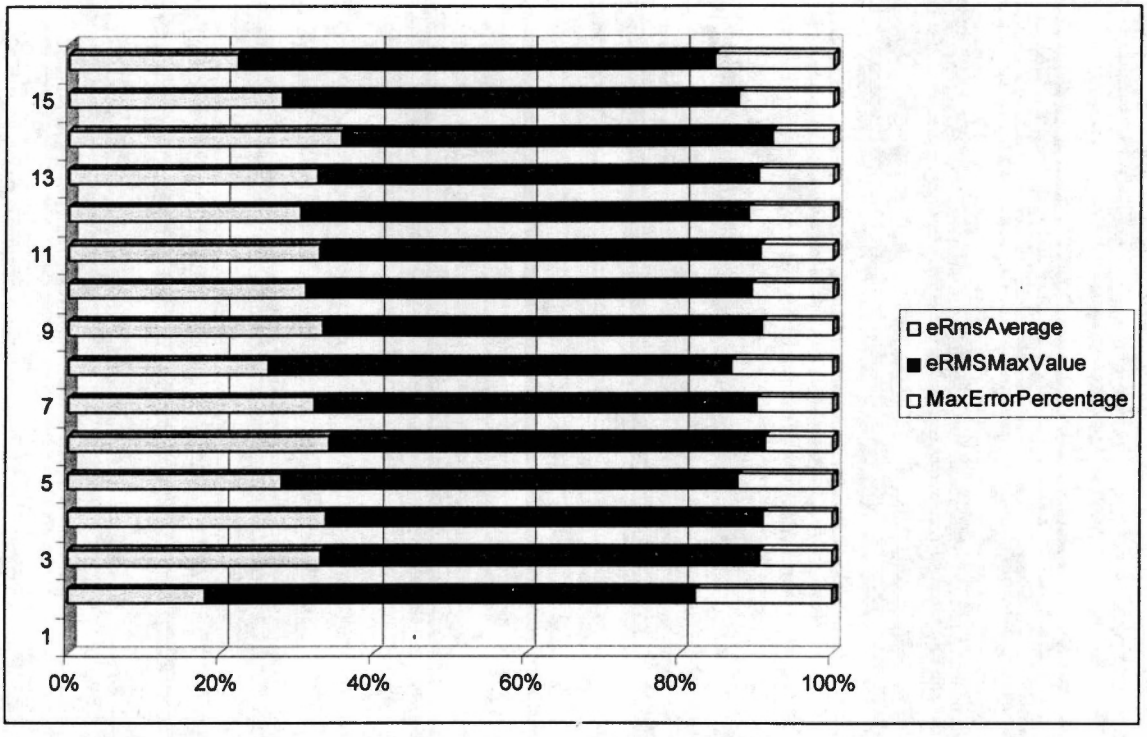
5.2 Análisis

5 segundos



Gráfica 4 Escenario diurno con objeto pasando frente a la cámara en 5 segs.

10 segundos

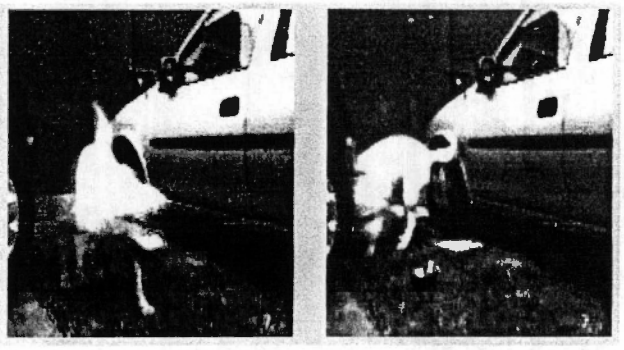


Gráfica 5 Escenario diurno con objeto pasando frente a la cámara 10 segs.

Los resultados de este escenario nos muestran las diferentes variaciones en los promedios del error cuadrático, esto se debe a los diferentes cambios en la información de cada imagen aunque el comportamiento difícilmente sobre pasa el 40% en comparación con el error máximo

6. Escenario diurno con cámara en movimiento

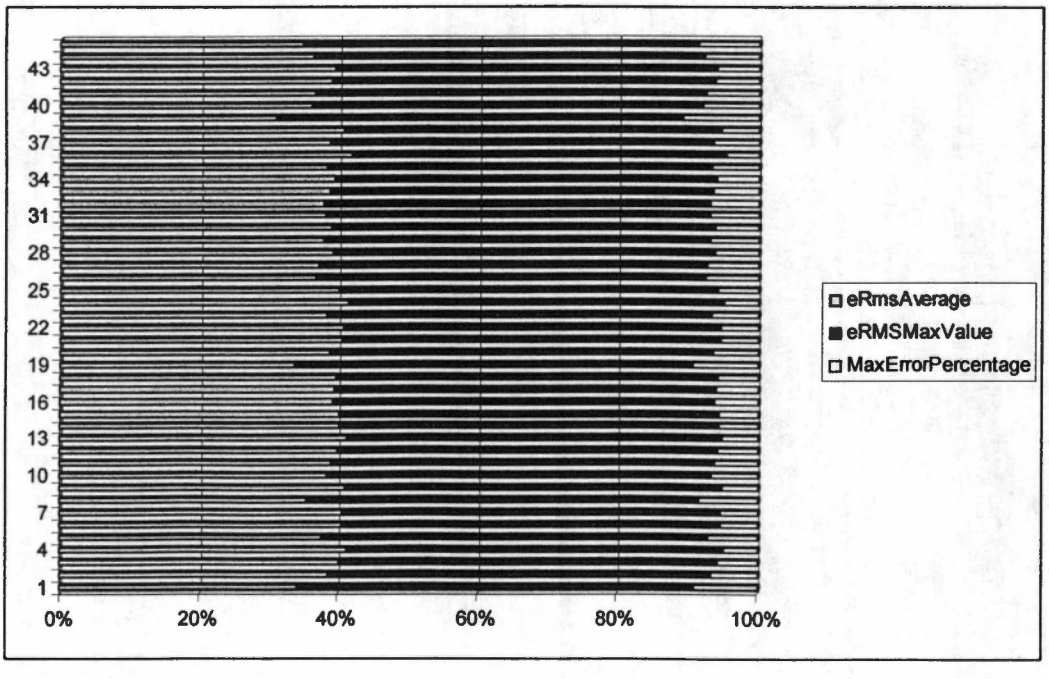
6.1 Descripción



En este experimento la cámara se fue moviendo en círculos para que cada imagen capturada sea completamente distinta a la anterior sin ningún rasgo parecido Además se tenían diversos movimientos frente a la cámara en cada secuencia.

6.2 Análisis

El intervalo de captura fue de un segundo y el de secuencia de 10 segundos.



Gráfica 6 Escenario con cámara en movimiento.

A comparación del escenario anterior, el promedio de eRMS aumenta considerablemente, esto se debe a que tanto la cámara como los objetos grabados se encontraban en movimiento lo cual ocasiona que los cambios en la información de las imágenes fueran considerables. Este escenario ayudó mucho en la calibración ya que permitió ver el comportamiento de uno de lo escenarios más cambiantes que podrían encontrarse.

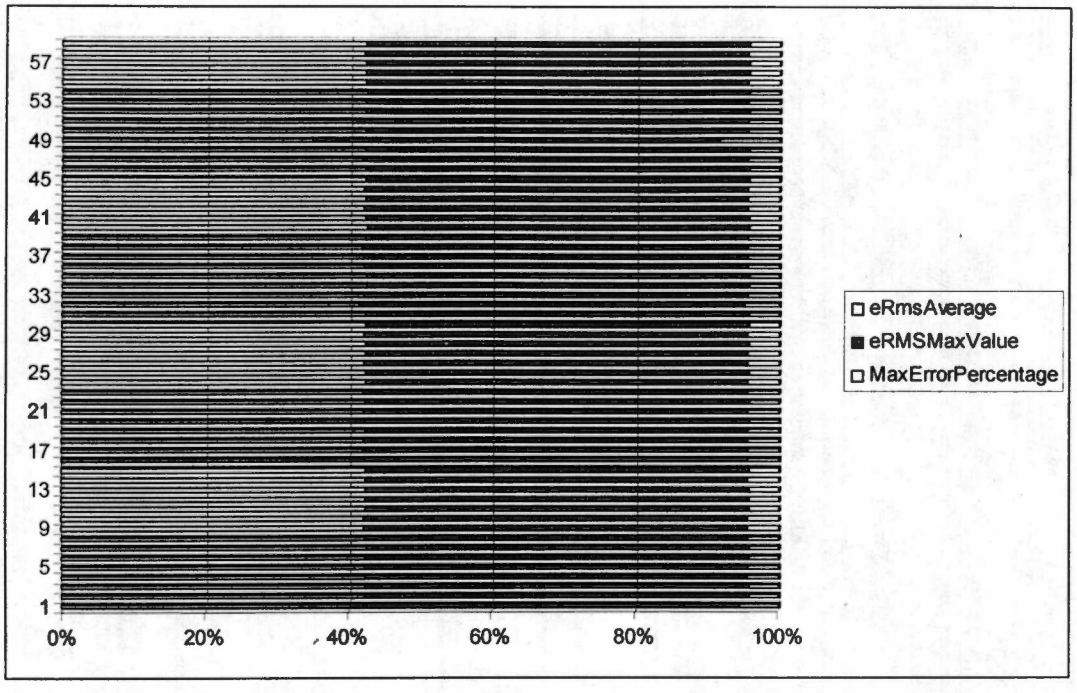
7. Escenario nocturno con objeto fijo

7.1 Descripción

Se situó la cámara en un lugar iluminado por luz artificial, grabando una pared, capturando imágenes cada segundo y con las secuencias de generación de video de 5 y 10 segundos.

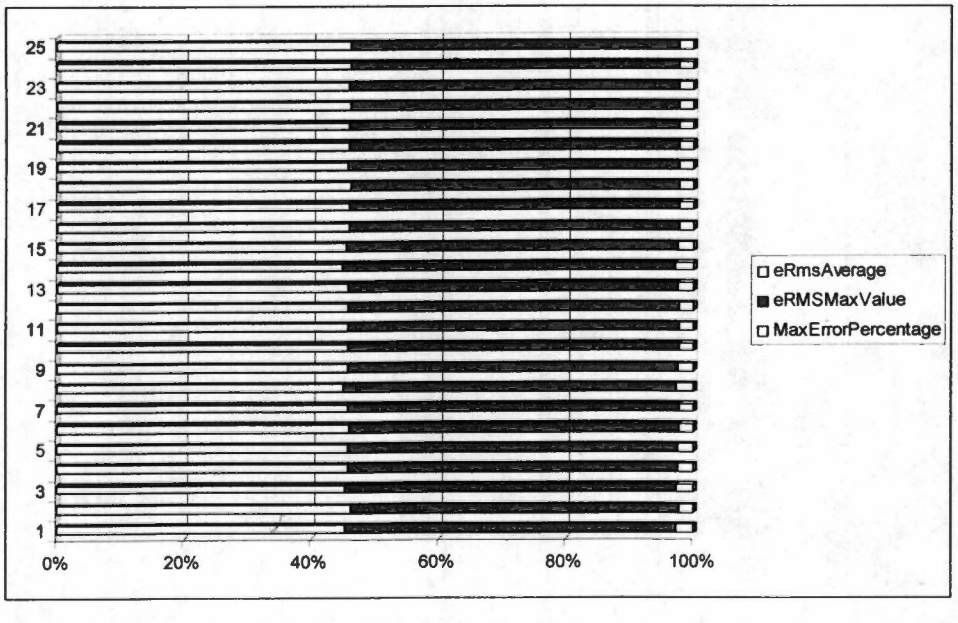
7.2 Análisis

5 segundos



Gráfica 7 Escenario nocturno con objeto fijo 5 segs.

10 segundos

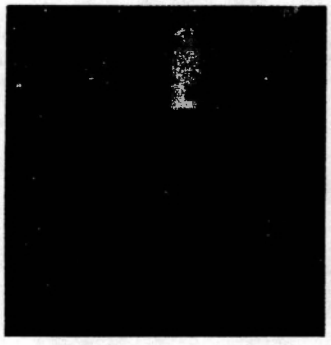


Gráfica 8 Escenario nocturno con objeto fijo 10 segs.

Este escenario se desarrolló para poder contrastarlo con la misma situación pero con la diferencia de la luz, sin embargo el comportamiento es muy similar tanto ya que no hay movimiento y el sistema no detecta cambios.

8. Escenario nocturno con objeto pasando frente a la cámara

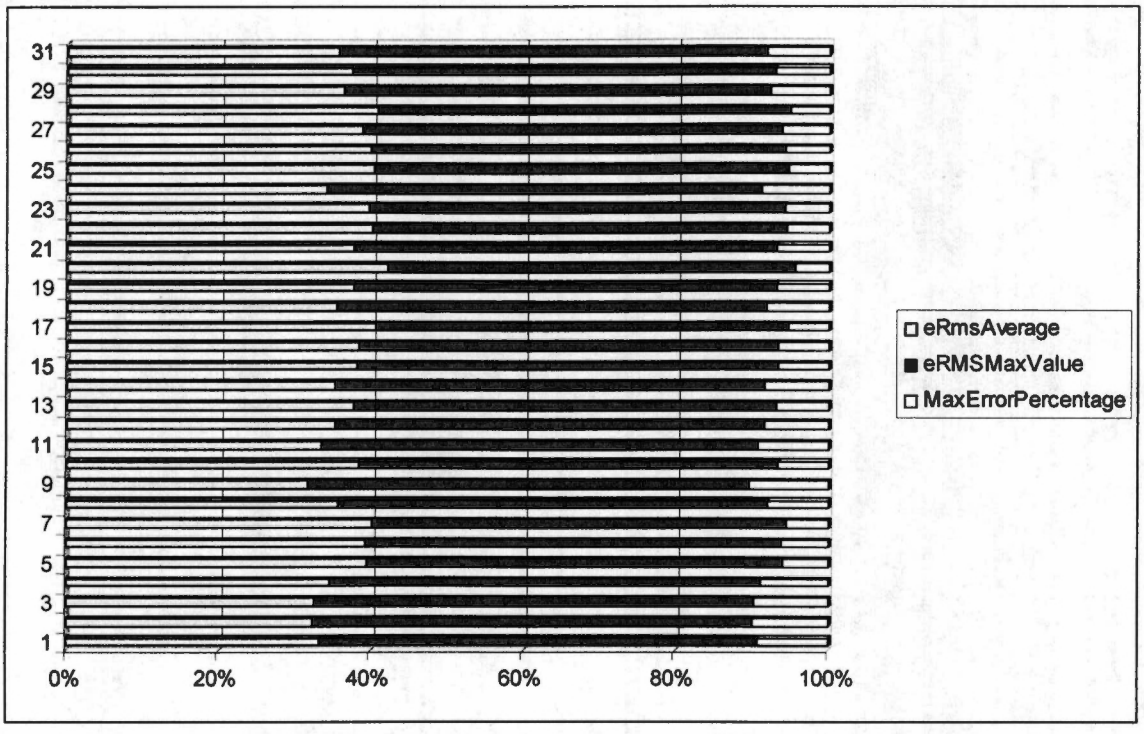
8.1 Descripción



Al igual que en el experimento anterior en cuanto al ambiente, con la diferencia que ahora un objeto se hacía pasar frente a la cámara.

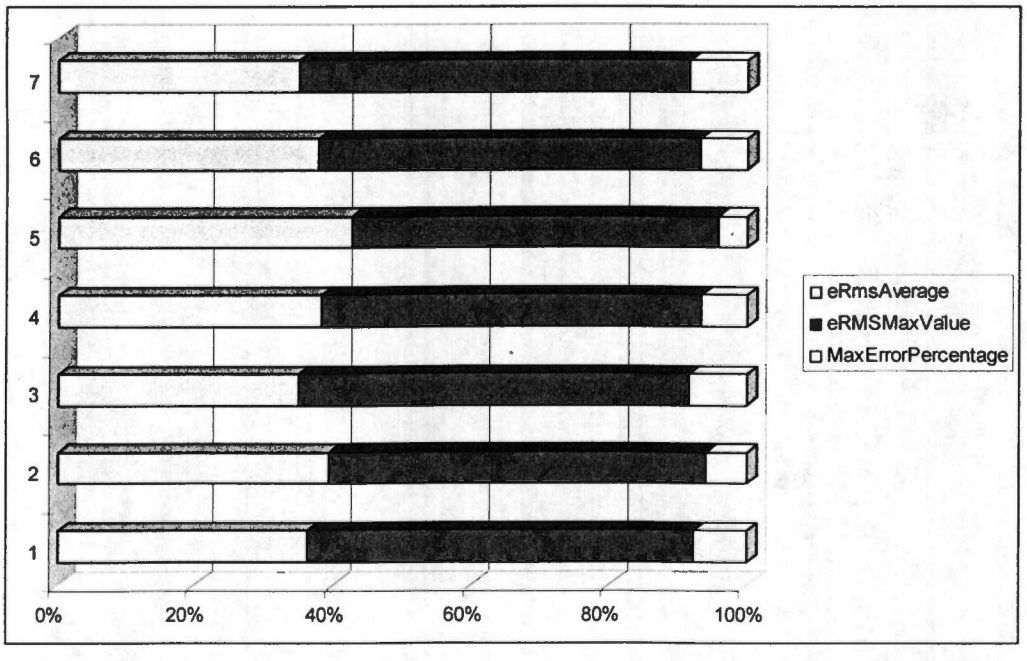
8.2 Análisis

5 segundos



Gráfica 9 Escenario nocturno con objeto pasando frente a la cámara.

10 segundos

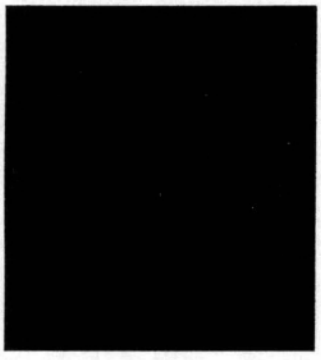


Gráfica 10 Escenario nocturno con objeto pasando frente a la cámara 10segs..

Dado que el experimento involucraba obscuridad, es claro que la cámara no tenía mucha información, por lo cual existió una diferencia entre el escenario diurno con el nocturno en cuanto a los valores de eRMS. Debido a la oscuridad, en este escenario la cámara no tenía mucha información por lo tanto los cambios de movimiento no eran detectados con tanta sensibilidad como en el escenario diurno.

9. Escenario nocturno con cámara frente a lámpara y apagón repentino

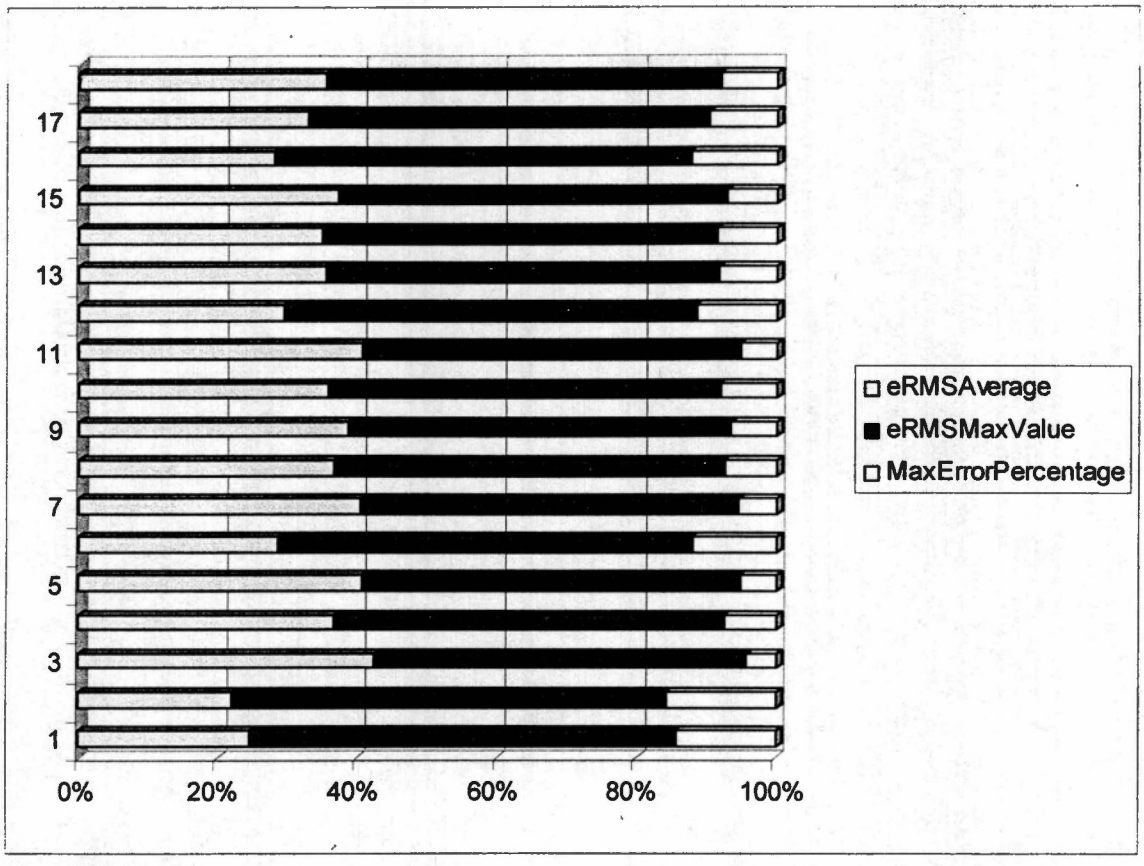
9.1 Descripción



En este escenario, se colocó a la cámara frente a una lámpara de tal manera que solamente pudiera percibir una luz blanca intensa, posteriormente se apagó repentinamente la luz con la intención de que en cada secuencia existieran altos y bajos en la intensidad de la luz.

9.2 Análisis

Para este experimento solamente se tomó el intervalo de generación de video de 5 segundos.



Gráfica 11 Escenario nocturno frente a lámpara y apagón.

Este escenario fue el más radical encontrado ya que se buscaba jugar con los cambios de intensidad bruscos, para poder analizar el comportamiento en un escenario extremo. Los valores obtenidos ayudaron en la calibración ya que de esta manera se logró definir el los valores más altos posibles que se pueden obtener.

Resultados

Después de haber descrito y analizado todos los escenarios anteriores, se descubrió que APCANI encuentra diversos obstáculos con cambios muy suaves dentro de secuencias muy larga, y esto se debe a que el algoritmo de análisis está directamente relacionado con los promedios de los errores, lo cual imposibilita poder lanzar alertas de niveles muy altos aunque la imagen sea completamente distinta.

Con el uso de APCANI, se pudo determinar que su mejor funcionamiento se dio al tener intervalos de captura muy pequeños ya que de esta manera el análisis para disparar las alertas funcionaban mejor. Esto se encuentra íntimamente ligado con el párrafo anterior debido al manejo del algoritmo.

APCANI logró exitosamente la discriminación de secuencias sin cambio, en donde solamente generó archivos de un segundo con la fotografía de la imagen, lo cual es una gran ventaja ya que el usuario puede monitorear un objeto fijo sin tener que ver secuencias que no presentan información valiosa.

Conclusiones

Con el presente documento se logró finalmente encontrar algunos de los requerimientos que el mercado doméstico tiene con respecto a actividades de monitoreo, principalmente de seguridad por el entorno en que actualmente se vive.

Dados los requerimientos encontrados, a lo largo del documento se presentó una parte teórica básica que ayudaría al entendimiento posteriormente del desarrollo de la aplicación que se programó. Con los conceptos de imágenes y compresión proporcionados, se pudo explicar el algoritmo que se utilizó para el análisis de las imágenes previas a la generación de secuencias.

El desarrollo de la aplicación "APCANI" cumplió casi todos los requerimientos definidos en los objetivos iniciales, desde poder tener archivos de tamaño reducido logrado con la compresión, sin perder la calidad de los videos, así como también programar la discriminación de videos para generar notificaciones vía correo electrónico cuando la configuración lo solicitara.

Después de la realización de la calibración, en donde se generaron diversas pruebas, aunque se logró tener notificaciones correspondientes a los niveles especificados, se encontró que el análisis de una secuencia requiere de más

estudio, incluso el uso de diversos algoritmos para que en conjunto se pueda lograr niveles de efectividad más altos.

Aunque APCANI se encuentra enfocado a usuarios sin conocimientos detallados sobre características de video, imágenes, etc.; resulta complicado encontrar una relación en cuanto a los niveles de alerta y su asociación con la cantidad de movimiento que un usuario pretende definir. Probablemente el uso continuo de APCANI pueda ayudar a los usuarios a familiarizarse con los niveles de alerta para lograr configuraciones que cumplan sus requerimientos.

Con el desarrollo de APCANI, el autor se pudo dar cuenta del posible aislamiento que cada módulo tiene con respecto a los demás, lo cual abre la oportunidad de pensar en lograr un producto englobado en una caja que podría ser comercializado con costos muy por debajo de los existentes en el mercado actualmente.

Recomendaciones

1. Trabajo Futuro

1.1 Manejo de múltiples cámaras

APCANI al momento de la escritura del presente documento solamente soporta una cámara por instancia, sin embargo, se recomienda hacer la implementación para poder colocar simultáneamente más cámaras lo cual permitiría en algún momento cubrir un ángulo de 360 grados sobre algún objetivo.

1.2 Métodos de compresión adicionales

APCANI se implementó utilizando el "codec" de "MSVC" aunque se sugiere hacer un módulo de configuración donde se presente la opción de configurar "codecs" adicionales como por ejemplo: mpeg o divx. Además en la configuración se podría determinar el grado de calidad deseado para cada video.

1.3 Manejo remoto

Tomando en cuenta varias de las soluciones existentes en el mercado, una mejora para APCANI sería la posibilidad de tener un manejo de la cámara,

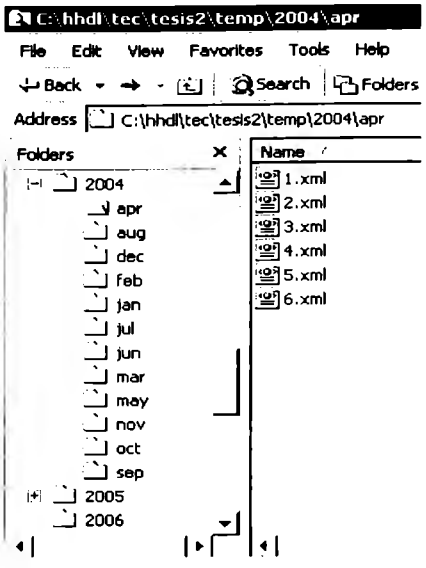
desde poderla mover para buscar algún objetivo en particular, hasta realizar acercamientos o alejamientos. Al contar con los puntos anteriores, también se debería implementar la opción de prender la cámara desde la página de Internet.

1.4 XML

APCANI fue implementado utilizando una base de datos, sin embargo se encontró que solamente existían una tabla con la mayor parte de la información, lo cual despertó la inquietud de realizar una implementación utilizando XML y de esta manera tener únicamente un servidor de archivos.

La implementación XML sugerida se resume de la siguiente manera.

1. Generar en el servidor de almacenamiento un esquema de directorios cuya raíz sea el año teniendo como hijos un directorio por cada mes del año.
2. Hacer un archivo XML para cada día del mes, en donde se irían almacenando todas las entradas (videos) por cada día. Es mejor tener 30 archivos en un directorio por si se llega a borrar alguno, no se pierda la información de todo un mes de almacenamiento.



3. Cada archivo XML tendría la siguiente estructura.

```
<videos>
      <year></year>
      <month></month>
      <day></day>
      <video>
            <id></id>
            <generated_date></generated_date>
            <start_date></start_date>
            <end_date></end_date>
            <full_file_name></full_file_name>
            <camera_id></camera_id>
            <images_number></images_number>
            <eRMSMax></eRMSMax>
      </video>
</videos>
```

Aunque no se pudo probar esta solución, se cree que los tiempos de respuesta para las búsquedas serán más lentos que los proporcionados por un manejador de base de datos ya que en estos últimos la información se encuentra indexada facilitando su búsqueda.

1.5 Selección de área

Debido al manejo del algoritmo, algunas secuencias presentan información no valiosa que genera alertas en momentos que probablemente el usuario no considere importantes, ante esta situación, surge la necesidad de poder definir un área de interés dentro del ángulo de captura de la cámara. Lo anterior permitiría enfocar la atención en un área específica la cual dispararía las alertas en caso de ser necesario.

Bibliografía

[Axis, 2004] Axis Network, (2004)
"http://www.axis.com".
19 de agosto del 2004

[Campos, 1999] Campos, Arturo, LZW, gif decoding. (documento web). (1999).
"http://www.arturocampos.com/ac_lzw_gif.html#Lzw%20decoding".
17 de julio del 2004

[Deitel, 1994] Deitel, H.M y Deitel, P.J. (1994).
[Como programar en C/C++]
Prentice Hall Hispanoamericana. Prentice Hall

[GeneralSolutions, 2004], General Solutions. (2004).
"http://www.generalsolutions.com".
4 de julio del 2004.

[González, 1987] González, Reafael y Woods, Richard. (1996).
[Tratamiento digital de imágenes]
Addison Wesley Publishing.London, England.

[JPEG, 2002], JPEG2000, The JPEG2000 Source. (documento web). Julio 2004.
"http://www.jpeg2000info.com/what/index.html".
10 de julio del 2004.

[Panasonic, 2003] Panasonic ideas for life. Noviembre 2003.
"http://www.panasonic.com/cctv/products/digitalrecording.asp".
1 de noviembre del 2003.

[Panasonic Camaras, 2003] Panasonic cameras. Noviembre 2003.
"http://www.panasonic.com/cctv/products/cameras.asp".
1 de noviembre del 2003

[SilkRoad, 2004] SilkroadTech. (2004)
"http://www.silkroadtech.com".
13 de junio del 2004.

[Tanenbaum, 1996] Tanenbaum, Andrew. (1996).
[Redes computacionales]
Computer Networks. Prentice Hall. Upper Saddle River, New Jersey

[TrueLook, 2004], TrueLook, (2004)
"http//www.truelook.com"
10 de julio del 2004

[UML, 1999] Fowler, Martin y Scott, Kendall. (1999).
UML Distilled Second Edition.. Addison Wesley Longman Inc.