

**UNIVERSIDAD RICARDO PALMA
FACULTAD DE INGENIERÍA
PROGRAMA DE TITULACIÓN POR TESIS
ESCUELA PROFESIONAL DE INGENIERÍA
ELECTRÓNICA**



**RECONOCIMIENTO DE GESTOS CORPORALES,
UTILIZANDO PROCESAMIENTO DIGITAL DE
IMÁGENES PARA ACTIVAR SISTEMA DE ALARMA
TESIS**

**PARA OBTENER EL TÍTULO PROFESIONAL DE
INGENIERO ELECTRÓNICO**

PRESENTADO POR:

Bach. MORENO MORENO FLAVIO DAVID

ASESOR: Dr. HUAMANÍ NAVARRETE PEDRO FREDDY

LIMA - PERÚ

AÑO : 2015

AGRADECIMIENTOS

Un agradecimiento muy especial a mi esposa e hija por su apoyo moral e incondicional día a día, a la espera de concluir con éxito el proyecto emprendido.

Un inmenso agradecimiento a mis padres, que con su apoyo y esfuerzo permitieron que concluya mis estudios, en tiempos difíciles.

Un sincero reconocimiento al esfuerzo realizado por el grupo humano de la Universidad Ricardo Palma integrante del Primer Programa Tites de Ingeniería Electrónica, en especial al Dr. Pedro Huamaní N. por su apoyo didáctico.

Flavio D. Moreno M.

ÍNDICE TEMÁTICO

INTRODUCCIÓN	01
Capítulo I: PLANTEAMIENTO DEL PROBLEMA	02
1.1 Problematización	02
1.2 Objetivos	03
1.3 Importancia y justificación.....	03
1.4 Estado del arte	04
Capítulo II: FUNDAMENTOS TEÓRICOS	07
2.1 Lenguaje corporal.....	07
2.1.1 Concepto de lenguaje corporal.....	07
2.1.2 Los gestos.....	07
2.1.3 Tipos de gestos.....	08
2.1.4 Reconocimiento de gestos.....	08
2.2 Procesamiento digital de imágenes.....	09
2.2.1 Concepto de imagen digital.....	09
2.2.2 Concepto de procesamiento digital de imágenes.....	10
2.2.3 Escala de grises	11
2.2.4 Componentes RGB.....	12
2.2.5 Binarización.....	13
2.2.6 Planos de bits.....	13
2.2.7 Transformaciones morfológicas.....	15
2.3 Hardware.....	19
2.3.1 Video cámara de vigilancia.....	19
2.3.2 Tecnología CCD y CMOS.....	19
2.3.3 DVR (Digital Video Record).....	23

2.3.4 PC.....	23
2.3.5 Puerto de comunicación serial, paralelo y USB.....	24
2.3.6 Interface Arduino.....	24
2.3.7 Circuito de control de relé con transistor bipolar.....	26
2.3.8 Gabinete de alarma	28
2.4 Software.....	28
2.4.1 Entorno MatLab.....	28
2.4.2 Interface gráfica de usuario (GUI).....	29
2.4.3 Toolbox de “Image processing”	29
2.4.4 Comandos de MatLab para arduino	31
2.4.5 Comandos de programación para arduino UNO R3.....	31
Capítulo III: TÉCNICAS DE PROCESAMIENTO Y ALGORITMOS DE	32
RECONOCIMIENTO	
3.1 Diagrama de bloques general	33
3.2 Gestos corporales específicos.....	35
3.3 Captura de imágenes	36
3.4 Técnicas de procesamiento y algoritmo matemático de reconocimiento...	39
3.4.1 Técnicas de procesamiento espacial aplicadas.....	40
3.4.2 Algoritmo matemático de reconocimiento.....	44
3.4.3 Diagrama de flujo	47
3.4.4 Configuración de puerto USB.....	49
Capítulo IV: DISEÑO Y PROGRAMACIÓN DE INTERFACE EXTERNA	53
4.1 Interface 01: Comunicación de ordenador con plataforma Arduino.....	54
4.2 Interface 02: Generación de 02 señales eléctricas pulsantes.....	54
4.3 Interface 03: Circuito electrónico para activar relé	56
4.4 Interface 04: Conexión eléctrica de relé con alarma.....	57

Capítulo V: PRUEBAS DE CAMPO	59
5.1 Atributos del proyecto implementado	59
5.2 Reconocimiento de gestos.....	61
5.2.1 Prueba de tiempo de respuesta	61
5.2.2 Prueba de ausencia de personal	61
5.2.3 Prueba de desactivación de alarma	62
5.2.4 Prueba de activación de alarma	62
5.2.5 Interferencias en el hardware.....	62
5.3 Prueba de activación de alarma en un edificio multifamiliar.....	63
CONCLUSIONES.....	65
RECOMENDACIONES.....	67
REFERENCIAS BIBLIOGRÁFICAS.....	68
ANEXOS	
Anexo 01 Encuesta: Ocurrencias que vulneran seguridad del edificio.....	73
Anexo 02 Datasheet de cámara de vigilancia.....	74
Anexo 03 Programa computacional desarrollado en MatLab y Arduino.....	75
Programa 'Alarmagestosarduino' en MatLab.....	75
Programa 'Apagaralarma'.....	77
Programa 'Activaralarma'.....	77
Anexo 04 Programa Arduino para generar dos tipos de señal eléctrica.....	78
Anexo 05 Costo del proyecto de tesis.....	79
Anexo 06 Encuesta: Implementación tecnológica en edificios de San Isidro....	80

ÍNDICE DE TABLAS

Tabla 01 Comandos MatLab utilizados frecuentemente.....	30
Tabla 02 Comandos MatLab para Arduino.....	31
Tabla 03 Comandos de programación para Arduino.....	31
Tabla 04 Tabla de comparación entre pin 12 y pin 13.....	55
Tabla 05 Costo de implementación en un edificio multifamiliar.....	61

ÍNDICE DE FIGURAS

Figura 01	Escala de grises	11
Figura 02	Cubo RGB	12
Figura 03	Componente R.....	13
Figura 04	Componente G.....	13
Figura 05	Componente B.....	13
Figura 06	Planos de bits	14
Figura 07	Aplicación de un elemento estructural (EE)	16
Figura 08	(a) Tecnología CCD.....	20
Figura 08	(b) Tecnología CMOS.....	22
Figura 09	Símbolo del USB.....	24
Figura 10	Placa arduino UNO R3	26
Figura 11	Circuito de control del relé con transistor bipolar.....	27
Figura 12	Diagrama de bloques	34
Figura 13	Counter vacío	36
Figura 14	Señal de apagar parado y mano a la altura del rostro.....	36
Figura 15	Señal de apagar parado y mano encima del rostro.....	36
Figura 16	Señal de apagar sentado y mano encima del rostro.....	36
Figura 17	Señal de activar sentado y manos a la altura del rostro.....	36
Figura 18	Señal de activar parado y manos encima del rostro.....	36
Figura 19	Señal de activar parado y manos a la altura del rostro.....	36
Figura 20	Imagen capturada a color e imagen procesada final.....	37
Figura 21	Cámara CCD.....	38
Figura 22	Posición de la línea 140.....	38
Figura 23	Extracción del plano rojo.....	41
Figura 24	Obtención del plano de bits más significativo.....	41
Figura 25	Obtención de imagen binarizada.....	42
Figura 26	Recorte de imagen	43
Figura 27	Transformación morfológica.....	44

Figura 28 Imagen inversa.....	44
Figura 29 Aplicación de algoritmos.....	46
Figura 30 Diagrama de flujo	48
Figura 31 Señales eléctricas generadas en el pin 08 del Arduino 02.....	56
Figura 32 Circuito interface para activar relé	57
Figura 33 Conexión eléctrica entre relé y gabinete de alarma.....	58
Figura 34 Interface implementada para pruebas de campo.....	64
Figura 35 Interface con cable apantallado para pruebas de campo.....	64

RESUMEN

La investigación realizada a los sistemas de seguridad electrónica de edificios, plantea como objetivo principal el reconocimiento de tres gestos de un lenguaje corporal del personal de vigilancia, y la consecuente activación de alarma en forma automática. Inicialmente se realizó una encuesta dirigida a las administraciones y personal de edificios, para saber cuales eran las ocurrencias que vulneraban la seguridad de un edificio multifamiliar, luego se observaron y analizaron las imágenes capturadas por una cámara de vigilancia ubicada en la recepción, identificando las ocurrencias más vulnerables y gestos asociados a dichos eventos; se seleccionaron tres gestos que en forma inconsciente realizaba el personal de vigilancia ante dichas situaciones. A determinados cuadros que comprenden estas imágenes se le aplicaron técnicas de procesamiento espacial, con ayuda de una iluminación artificial que era más intensa en la parte posterior del sujeto de análisis, consiguiéndose la definición de una silueta binarizada en el entorno Matlab, técnicas como selección del plano rojo, plano de bits más significativo, invertir imagen y transformaciones morfológicas tipo cerradura, definieron una silueta que ayudó a desarrollar un algoritmo matemático para generar una señal eléctrica en el puerto serial USB del ordenador, donde se conectó físicamente una plataforma de hardware Arduino que activa la alarma. La elección de esta plataforma se debió a que Matlab cuenta con un grupo de instrucciones para Arduino, con el objetivo de lograr una comunicación sincronizada entre ordenador e interface. Las técnicas utilizadas reconocieron 62,5% de los eventos descritos en las encuestas realizadas y que no son mencionadas en temas de investigación similar. Para lograr el objetivo fue necesario analizar un cuadro por segundo.

PALABRAS CLAVES

RGB: Señal de video compuesta por tres colores separados: rojo, verde, azul.

MatLab: Software matemático con lenguaje de programación propio.

IPT: Conjunto de instrucciones para procesar imágenes, provee de algoritmos.

Imagen binarizada: Imagen digital que solo contiene píxeles en blanco y negro.

Arduino: Plataforma de hardware y software libre con microcontrolador.

ABSTRACT

The research poses as their main objective the three gestures recognition of a body language of surveillance personnel and the consequent activation of alarm automatically. It was initially carried out a survey of the administration and the offices of the buildings to know which were the occurrences that violate the security of a multi-family building, then were observed and analyzed images captured by a surveillance camera located in the reception, identifying the most vulnerable occurrences and gestures associated with these events; were selected three gestures that unconsciously performs surveillance personnel before such situations; to certain pictures that comprise these images were applied spatial processing techniques, with the help of an artificial lighting that was more intense in the back of the subject of analysis, getting the definition of a silhouette binarized in the Matlab environment, techniques such as plane selection red, more significant bit plane, to invest an image and convolution close type, defined a silhouette that allowed to develop a mathematical algorithm that generated an electrical signal in USB serial port of the computer, where it is physically connected a hardware platform Arduino that active the alarm. This platform choice is due to the fact that Matlab has a group of instructions for Arduino, achieving an orderly communication between computer and interface.

The techniques used recognized 62.5 % of the events described in the surveys carried out and which aren't mentioned in similar research topics. To achieve the objective was necessary to analyze a picture per second.

KEYWORDS

RGB: Color system constructs all the colors from the combination of the Red, Green and Blue colors.

MatLab: Multi-paradigm numerical computing environment and fourth-generation programming language.

IPT: Image processing toolbox provides with an extensive set of algorithms.

Image binarized: Image that has only two possible values, black and white.

Arduino: Open - source electronic prototyping platform.

INTRODUCCIÓN

Los sistemas de seguridad electrónica en edificios multifamiliares, cuyo objetivo es activar una alarma cuando perciben alguna ocurrencia que vulnera la seguridad integral, a sido objeto de estudio constante; desarrollándose diversos diseños, configuraciones, soluciones e implementaciones de distintos costos para el resguardo del perímetro del edificio, áreas comunes e interior de departamentos sin tomar en cuenta eventos suscitados en la recepción; como resultado de ello los sistemas de seguridad electrónica envían automáticamente una señal eléctrica para activar una alarma local ó dar una alerta a una estación remota vía internet.

Parte de este sistema, lo conforma el circuito cerrado de televisión con o sin sistema de grabación y el personal de vigilancia ubicado en la recepción de los edificios; estos últimos actúan como un complemento del sistema de seguridad electrónica, ya que si bien perciben ocurrencias no pueden activar una alarma en forma automática, situación que se revierte en esta investigación.

En la investigación, la ausencia ó mal estado de salud del vigilante, robos ó asaltos en recepción, son algunas de las ocurrencias que se reconocen con técnicas de procesamiento espacial de imágenes de gestos corporales del personal de vigilancia, captados por una cámara de video del circuito cerrado de televisión, activando la alarma sin exponer la integridad del personal.

Por tal razón se hace un detalle de los fundamentos teóricos que sustentan la investigación de reconocimiento de gestos, y la descripción de las técnicas de procesamiento utilizadas para obtener una imagen binarizada con atributos que permitieron elaborar un algoritmo matemático apropiado, generándose una información y tomando una decisión en el entorno de Matlab, con el objeto de entregarla a una interface externa, dicha interface cuenta con un protocolo de comunicación, para que el ordenador transmita y reciba información en forma ordenada por el puerto USB, por lo que se usa la plataforma de hardware y software libre de Arduino, la misma que llega a activar la alarma.

CAPÍTULO I: PLANTEAMIENTO DEL PROBLEMA

1.1 Problematicación

El problema general consiste en reconocer imágenes de gestos de un lenguaje corporal realizado por el personal de vigilancia de un edificio, para generar una señal eléctrica capaz de activar una alarma.

Una gran ayuda es la cámara de seguridad del circuito cerrado de televisión ubicada en la recepción de los edificios multifamiliares, porque a partir de ella se capturan imágenes con determinada resolución y velocidad de transmisión, que al ser operacionalizadas con técnicas matemáticas de procesamiento espacial se reconocen como una alerta, por lo que se describen algunas de las técnicas mencionadas que permiten reconocer digitalmente estos gestos corporales.

La captura y procesamiento espacial de imágenes, permite desarrollar algoritmos matemáticos que tomen decisiones según reglas definidas establecidas en un programa, consistente en enviar o no una señal eléctrica, al sistema de alarma a través de una interface externa conectada en el terminal USB del ordenador.

Con una adecuada elección de la cámara de seguridad, las técnicas de procesamiento y el algoritmo; el ordenador genera señales eléctricas en uno de sus terminales seriales USB, donde se conecta con una interface externa cuyo diseño y programación es compatible con el sistema de alarma y el ordenador, para poder activar la alarma cuando se requiera su uso.

Para implementar este proyecto de tesis fue necesario adquirir físicamente los componentes mencionados, los cuales se recibieron en calidad de

préstamo de la empresa “digital net” dedicada al rubro de seguridad electrónica.

1.2 Objetivos

1.2.1. Objetivo general:

Utilizar técnicas matemáticas de procesamiento espacial de imágenes, en el reconocimiento de gestos corporales específicos, para activar el sistema de alarma en presencia de luz artificial.

1.2.2 Objetivos específicos:

1. Seleccionar la técnica matemática de procesamiento espacial, para el reconocimiento de tres gestos corporales, captados por una cámara de video.
2. Implementar algorítmicamente las técnicas de procesamiento espacial en el entorno del software MatLab para el reconocimiento de tres gestos corporales.
3. Diseñar una Interfaz que reciba una señal eléctrica de una PC, para activar el sistema de alarma.

1.3 Importancia y justificación

Este nuevo sistema permite aumentar la eficacia y el grado de seguridad de los edificios multifamiliares, evitando la exposición innecesaria de la integridad del personal de vigilancia y de los residentes, dejando la labor de respuesta inmediata a las Empresas e Instituciones a fines que brindan

los servicios de seguridad a los alrededores de los edificios multifamiliares. La permanencia y/o ausencia del personal de vigilancia, puede ser supervisada por personal de administración y residentes del predio en tiempo real, significa pues un ahorro de tiempo y en algunos casos de dinero, pues no es necesario contratar servicios de proveedores para revisar videos; pues el conocimiento advertido de la existencia de esta implementación y la activación de la alarma hace de la labor del vigilante en recepción un trabajo más eficiente.

Este nuevo sistema puede aplicarse a otras situaciones similares con una ligera variación del algoritmo computacional, donde se necesita la presencia de un personal permanente como: control de personal al servicio de cuidado del adulto mayor o niños, personal de recepción de diferentes empresas; también en empresas donde el personal no puede alertar una ocurrencia por encontrarse amenazados como: bancos, casinos, notarías, farmacias, bodegas, residencias, conductores de automóvil.

Asimismo el ordenador que envía una señal eléctrica no solo podrá activar un sistema de alarma, si no también puede bloquear un ingreso ó salida, bloquear un auto, activar una nube que bloquee la visión, entre otras actividades u eventos.

1.4 Estado del arte

Siendo el objeto de estudio la seguridad de edificios multifamiliares, debe alertarse toda ocurrencia que genere una posible intrusión, observando la investigación realizada en la Tesis “Diseño de un sistema automatizado de seguridad contra intrusiones en un edificio para departamentos” [1], se observa que se desarrolla un sistema de seguridad electrónica para edificios multifamiliares, para evitar intrusiones, llegando a detectar todos los eventos posibles que atenten contra la integridad del residente que se plantearon, sin mencionar eventos como los descritos en la recepción del

edificio [2]; por esta razón este tipo de solución no es tan eficiente ya que solo percibe un grupo de eventos que están limitados al interior de los departamentos, por lo que se necesita un sistema complementario que detecte ocurrencias que se susciten en las áreas comunes de los edificios. Si bien la tecnología inalámbrica abarata costos de cableado, no suele ser muy efectiva ante la presencia de estructuras metálicas del propio edificio, ya que en el trayecto la transmisión de la señal inalámbrica suele atenuarse, generando falsas alarmas, con este sistema se llega a detectar aproximadamente 50% de eventos que vulneran la seguridad de un edificio.

En cuanto al desarrollo del algoritmo, se observa un gran aporte, en estudios realizados en otras áreas de seguridad como son los supermercados, como la Tesis "Detección y seguimiento de manos en videos digitales utilizando computador" [3], esta investigación aporta una idea de las técnicas de procesamiento espacial de imágenes necesarias, siendo muy efectivo con una adecuada resolución de las cámaras, para llegar a dar una alerta en tiempo real, con un computador. Se desarrolló un algoritmo mas simple para que un ordenador a diferencia de un computador, perciba los gestos corporales en tiempo real y genere una alarma y pueda estar presente en un edificio multifamiliar por un tema económico, por eso la investigación consideró el hardware de una PC logrando una solución más económica. La velocidad de procesamiento que se logra en esta investigación es de 1,87 cps.

Estudios en áreas como: reconocimiento de gestos, tecnología de lenguaje, procesamiento digital de imágenes, aportan ideas interesantes pero dirigida a eventos específicos como: Aplicaciones de Dataglove, que necesita un guante de manos de color negro para que su sistema sea eficaz, o la varita mágica de LG Electronics, Mouse Electrónico, entre otras aplicaciones que no se ajustan a la investigación, porque no es práctico que el personal de vigilancia transporte estos componentes a donde se traslade, pero aportan ideas para el desarrollo

del algoritmo.

Otros aspectos no contemplados por muchos de estos aplicativos, es que requieren de iluminación constante, y no dan solución al fenómeno de oclusión, es decir cuando manos y/o rostro se superponen ante el equipo que los observa.

Una tecnología presente y que puede ayudar es WiSee, que es una interfaz de interacción que potencia las transmisiones inalámbricas en curso en el ambiente (por ejemplo Wifi), para posibilitar la percepción y reconocimiento en todo un domicilio de los gestos humanos, el costo de implementación supera los US\$ 2,500 dólares americanos [4].

IVS (Intelligent Video Surveillance) son sistemas integrados capaces de procesar Metadatos a partir de videos orientado a la toma de decisiones, con capacidad de servicio y gestión estadística e interactúan con DVR y cámaras; al igual que los IVS tienen un costo de implementación de US\$2,000 a US\$4,000 dólares americanos [5].

CAPÍTULO II: FUNDAMENTOS TEÓRICOS

2.1 Lenguaje corporal

2.1.1 Concepto de lenguaje corporal

El Lenguaje corporal ó quinésica, estudia el significado expresivo o comunicativo de los movimientos corporales y de los gestos aprendidos no orales, de percepción visual, auditiva o táctil, usado para formas de comunicación en los que intervienen movimientos corporales y gestos, en vez de los sonidos ó lenguaje verbal u otras formas de comunicación, conscientes o inconscientes, voluntario e involuntario [6].

A veces usamos un texto en lugar de una palabra o de un enunciado, o dibujamos algo con las manos para complementar lo que decimos oralmente. Esto pertenece a la categoría de los paralenguajes, que describen todas las formas de comunicación humana no verbal. Incluye los movimientos más sutiles e inconscientes, incluyendo el guiño y los movimientos leves de las cejas. Además, el lenguaje corporal puede incluir la utilización de expresiones faciales y la postura del cuerpo [7].

2.1.2 Los gestos

Un gesto es una forma de comunicación no verbal ejecutada con alguna parte del cuerpo y producida por el movimiento de las articulaciones, músculos de brazos, manos o cabeza [8].

El lenguaje de los gestos permite expresar una variedad de sensaciones y pensamientos, desde desprecio y hostilidad hasta aprobación y afecto. Prácticamente todas las personas utilizan gestos y el lenguaje corporal en

adición de palabras cuando hablan. Ciertos tipos de gestos pueden ser considerados culturalmente aceptables o no, dependiendo del lugar y contexto en que se realicen.

Según Desmond Morris, los gestos son actos que envían una señal visual mediante el movimiento de alguna parte del cuerpo, las manos, los brazos, la cara o de la cabeza, por lo que forman parte de la comunicación no verbal. Para considerar que esos movimientos representan un verdadero gesto, tienen que ser captados por alguien ajeno a quien lo realiza y transmitir información, porque lo importante no son las señales enviadas, sino las recibidas [9].

2.1.3 Tipos de gestos:

Según Paúl Ekman y Wallace Friesen [10], distinguen cinco tipos de gestos:

1. Gestos emblemáticos ó emblemas: son señales emitidas intencionalmente.
2. Gestos ilustrativos o ilustradores: acompañan a la comunicación verbal para matizar o recalcar lo que se dice.
3. Gestos reguladores de la interacción o reguladores: Con ellos se sincroniza o se regula la comunicación y el canal no desaparece.
4. Gestos que expresan estados emotivos o muestras de afecto: este tipo de gestos reflejan el estado emotivo de la persona.
5. Gestos de adaptación o adaptadores: son aquellos gestos que se utilizan para manejar emociones que no queremos expresar.

2.1.4 Reconocimiento de gestos

El reconocimiento de gestos es una aplicación del área de visión por un computador en la que un conjunto de técnicas de procesamiento de imágenes y análisis de series temporales, son utilizadas para hacer que el ordenador

"entienda" un gesto capturado por una cámara o Webcam [11].

El "Reconocimiento de gestos" es un tema en ciencias de la computación y la tecnología del lenguaje con el objetivo de interpretar gestos humanos a través de algoritmos matemáticos. Los gestos pueden ser cualquier movimiento corporal o estado, pero comúnmente se originan a partir de la cara o la mano.

Las técnicas de reconocimiento de gestos puede ser visto como una manera para que las computadoras empiecen a entender el lenguaje corporal humano, construyendo así una relación más sólida entre máquinas y seres humanos, interactuando naturalmente sin dispositivos mecánicos. Dejando atrás sistemas primitivos como las interfaces de usuario de texto o incluso GUIs (interfaces gráficas de usuario), que limitan la mayoría de las entradas informáticas a el teclado y el mouse. El uso de esta tecnología que puede implementarse crea un nuevo trabajo de investigación en el desarrollo de un nuevo hardware sin necesidad de monitores.

2.2 Procesamiento digital de Imágenes

2.2.1 Concepto de imagen digital

Se caracteriza por ser un conjunto de puntos (píxeles) de intensidades diferentes, con valores finitos. Matemáticamente es una función bidimensional $f(x, y)$ donde x e y son posiciones y $f(x, y)$ es la intensidad o el nivel de gris. Por otro lado un video digital, está compuesto de una secuencia de imágenes digitales a razón de 25 ó 30 cuadros por segundo según el sistema utilizado (PAL ó NTSC respectivamente), bajo un formato de compresión, como H.264, que es el más utilizado en grabadores digitales [12].

2.2.2 Concepto de 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. Este conjunto de técnicas tiene como objetivo fundamental obtener a partir de una imagen origen otra final, mediante filtros u operaciones matemáticas, cuyo resultado sea más adecuado para una aplicación específica, mejorando ciertas características de la misma que posibilite efectuar otras operaciones u algoritmos del procesado sobre ella [13].

Los principales objetivos que se persiguen con la aplicación de filtros son:

1. Suavizar la imagen: reducir la cantidad de variaciones de intensidad entre píxeles vecinos.
2. Eliminar ruido: eliminar aquellos píxeles cuyo nivel de intensidad es muy diferente al de sus vecinos y cuyo origen puede estar tanto en el proceso de adquisición de la imagen como en el de transmisión.
3. Realzar bordes: destacar los bordes que se localizan en una imagen.
4. Detectar bordes: detectar los píxeles donde se produce un cambio brusco en la función intensidad.

Por tanto, se consideran los filtros como operaciones que se aplican a los píxeles de una imagen digital para optimizarla, enfatizar cierta información o conseguir un efecto especial en ella. El proceso de filtrado puede llevarse a cabo sobre los dominios de frecuencia y/o espacio. Las operaciones de filtrado en el dominio del espacio se llevan a cabo directamente sobre los píxeles de la imagen. En este proceso se relacionan, para todos y cada uno de los puntos de la imagen, un conjunto de píxeles próximos al píxel objetivo con la finalidad de obtener una información útil, dependiente del tipo de filtro aplicado, que permita actuar sobre el píxel concreto en que se está llevando a cabo el proceso de filtrado, para de este modo obtener mejoras sobre la imagen y/o datos que podrían ser utilizados en futuras acciones o procesos de trabajo sobre ella.

Los filtros en el dominio del espacio pueden clasificarse en [14]:

Filtros lineales (filtros basados en kernels o máscaras de convolución).

Filtros no lineales.

Usando una PC se puede emular la visión humana en tres niveles:

Bajo: Adquirir imagen, reducir ruido, mejorar contraste y filtros de enfoque.

Medio: Extraer atributos de la imagen como contornos, bordes e identidad individual de los objetos.

Alto: Obtener significado del conjunto de objetos.

2.2.3 Escala de grises

En computación una escala de grises, es una escala acromática empleada en la imagen digital, en la que el valor de cada pixel posee un valor equivalente a una graduación de gris, que generalmente varía con una modulación continua mostrada en la figura 01, donde los valores van desde 0 (color negro) a 255 (color blanco), estos valores están representados por un byte (08 bits) [15].

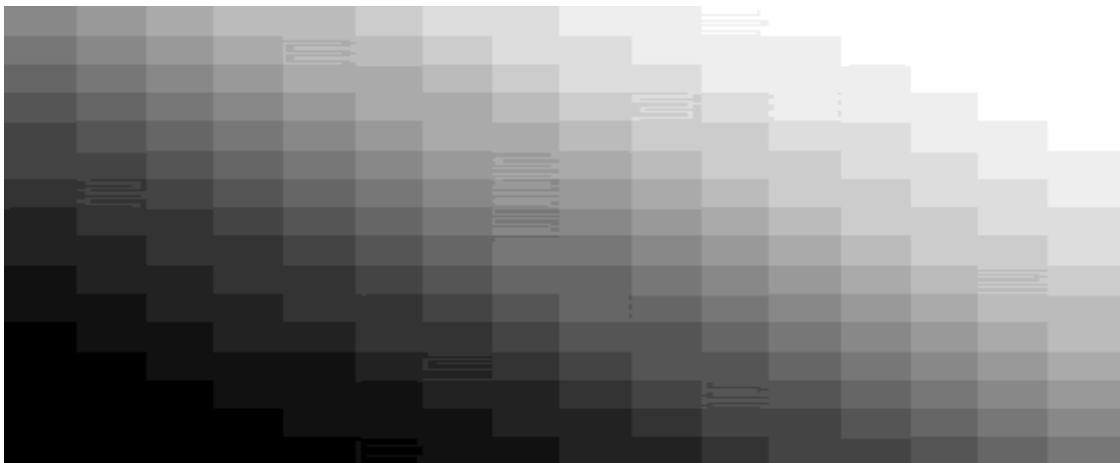


Figura 01: Escala de grises [16].

2.2.4 Componentes RGB

RGB es un modelo de color basado en la síntesis aditiva, con el que es posible representar un color mediante la mezcla por adición de los tres colores de luz primarios. El modelo de color RGB no define por sí mismo lo que significa exactamente rojo, verde ó azul, por lo que los mismos valores RGB pueden mostrar colores notablemente diferentes en diferentes dispositivos que usen este modelo de color. Aunque utilicen un mismo modelo de color, sus espacios de color varían considerablemente [17].

RGB es el tratamiento de la señal de vídeo que trata por separado las señales de los tres colores (R) rojo, (G) verde y (B) azul. Al usarlo independientemente, proporciona mayor calidad y reproducción más fiel del color.

La figura 02 nos muestra el resultado de ir variando gradualmente el porcentaje de cada color primario, donde cada uno forma parte de un eje de coordenadas, de allí que si no variamos valor alguno es decir el punto (0,0,0) obtenemos el color negro; si en igual proporción variamos R,G y B obtenemos el color blanco, es decir el punto (1,1,1), si se varia solo el rojo, obtendremos rojo puro, es decir el punto (1,0,0) al igual que el azul (0,1,0) y el verde (0,0,1).

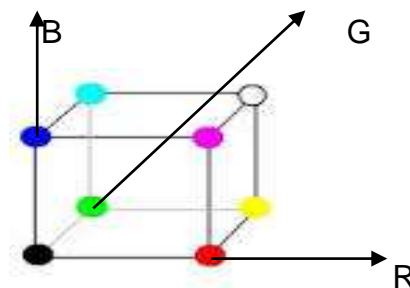


Figura 02: Cubo RGB [18].

Al ver las figuras 3, 4 y 5 se muestran componentes aditivas de planos, en forma individual representada en niveles de grises, un color oscuro indica un nivel de gris bajo, es decir poca presencia del color componente elegido y viceversa.



Figura 03
Componente R



Figura 04
Componente G



Figura 05
Componente B

2.2.5 Binarización

La binarización de imágenes es una técnica del procesamiento de imágenes que consiste en un proceso de reducción de la información de una imagen digital a dos valores: 0 para el negro y 1 para el blanco. Esta técnica consiste en comparar cada pixel de la imagen con un determinado umbral, que es un valor límite que determina si un pixel será de color blanco o negro. Los valores de la imagen que sean mayores que el umbral que se toma como referencia toman un valor de 255 es decir blanco, y el resto de píxeles toman valor 0 es decir negro [19].

2.2.6 Planos de bits:

Esta técnica se basa en el concepto de descomposición de una imagen multinivel (monocromática o en color) en una serie de imágenes binarias, una por cada bit usado en la representación de la intensidad del pixel. Los niveles de gris de una imagen con una escala de grises de m bits se puede representar como un polinomio en base 2, como se muestra en la ecuación (1):

$$a_{m-1} 2^{m-1} + a_{m-2} 2^{m-2} + \dots + a_1 2^1 + a_0 2^0 \dots\dots\dots(1)$$

Donde a_i , $i=0, \dots, m-1$ toma uno de los dos valores 0 o 1. El plano de bits de orden cero se genera a partir de cada uno de los bits a_0 de todos los píxeles, mientras que el plano de bits de orden $m-1$ contiene los bits a_{m-1} . Por ejemplo,

para el caso de una imagen de 8 bits, el plano de bit más significativo de un pixel es representado por un 1 si la correspondiente intensidad del pixel es igual o mayor que 128. Se puede observar que una imagen binaria puede ser representada en un solo plano de bit. Una descomposición de un plano de bit para una imagen de 8 bits se representa a continuación, en la figura 06 para un nivel de gris de 130 [20].

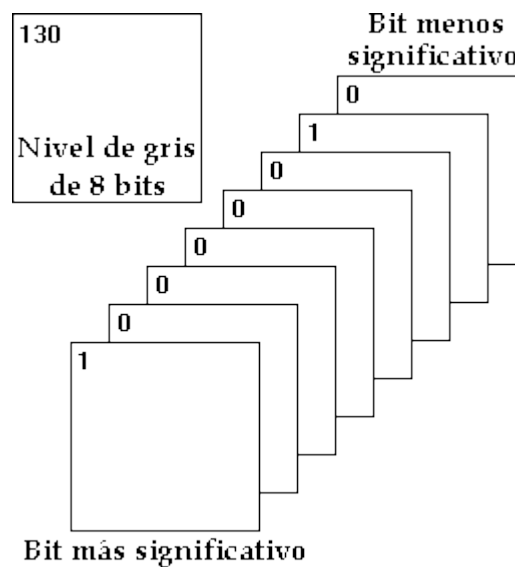


Figura 06: Planos de bits [21].

Una desventaja de la representación del plano de bit anterior es que algunos cambios pequeños en los niveles de gris, como variaciones debido al ruido, pueden causar variaciones en todos los planos de bits. Por ejemplo, la representación binaria para 127 es 01111111, y para 128, este es 10000000. Para reducir el efecto de tales variaciones de nivel de gris en un plano de bits, se pueden representar las intensidades de un pixel mediante un código Gray

de m bits, donde las palabras de código sólo difieren en la posición de un bit. El código Gray de m bits $g_{m-1} \dots g_2 g_1 g_0$ se puede calcular utilizando las series definidas en las ecuaciones (2) y (3), donde g es el nuevo valor resultante de las operaciones resultante de los coeficientes 'a' del polinomio de la ecuación (1).

$$g_i = a_i \oplus a_{i+1} \quad 0 \leq i \leq m-2 \dots\dots\dots(2)$$

$$g_{m-1} = a_{m-1} \dots\dots\dots(3)$$

Donde \oplus se indica la operación OR exclusivo. Además, los planos de bits codificados con el código Gray son menos complejos que los correspondientes planos de bits binarios.

2.2.7 Transformaciones morfológicas

El análisis morfológico en imágenes se aplica basándose en dos campos la geometría y la forma.

Su utilización puede ser para distintos campos:

1. Suavizar los bordes de las figuras. Esto es útil si se quiere mejorar un borde ya que normalmente en las capturas de las imágenes usando las técnicas de segmentación estándar los bordes suelen presentarse con mucho ruido apareciendo pequeñas perturbaciones
2. Separar determinadas regiones. El proceso de segmentación puede unir regiones distintas que con el análisis morfológico son separables.
3. Unir regiones. Al igual que la segmentación puede unir regiones también puede separar partes de una región.
4. Facilita el cómputo de las regiones de una imagen.

Para el análisis se utilizan una serie de transformaciones más o menos complejas dependiendo si se trabaja con imágenes binarias o con imágenes

en las que se estudia es la intensidad, sobre todo en el tratamiento de grises, el símbolo que representa estas transformaciones es: ϕ .

Este tipo de transformaciones se basan en que toda imagen real se puede componer y ser modeladas como conjuntos en cualquier dimensión. Las computadoras usan unos conjuntos cuyos elementos son pares de enteros, así un punto viene dado por 2 coordenadas, entonces una imagen binaria puede tratarse como un conjunto de punto 2D. Estos puntos se guardan en una matriz $A(i,j)$ cuyo valor en i es su componente x en el espacio 2D y la componente j es la componente en y . Los valores '1' significan que ese píxel pertenece a la figura y los valores a '0' que no pertenece [22].

Una transformación morfológica viene dada por la relación del conjunto de valores '1' de la matriz y un conjunto de valores llamados elemento estructural (EE). Una transformación morfológica básicamente es desplazar el elemento estructural (EE) por toda la imagen X , haciendo coincidir el centro del elemento estructural con cada uno de los píxeles de X ; siendo X el conjunto de puntos de la figura o mejor dicho los valores '1' de la matriz, mostrando dicho proceso en la figura 07.

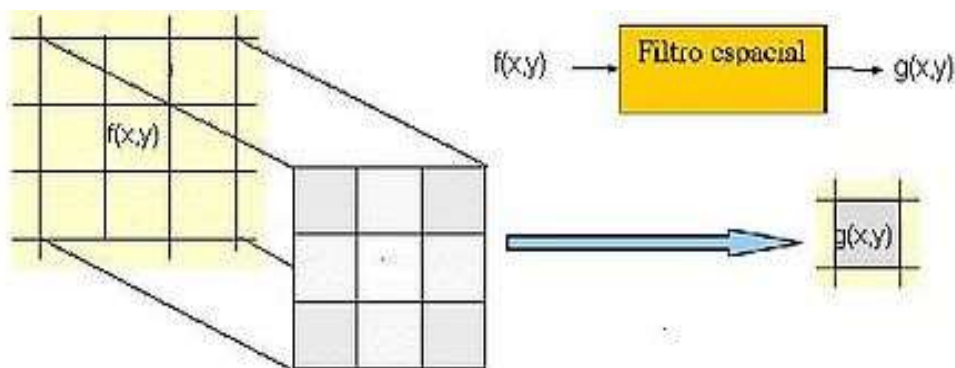


Figura 07: Aplicación de un elemento estructural (EE) [23].

El pixel final de la imagen se calcula colocando el centro del elemento estructural sobre el píxel y aplicando una relación, ya sea unión, intersección, contenido, entre otros.

Sea M la matriz que contiene una imagen binaria, donde “ * ” es el origen de coordenadas, es decir el punto de inicio donde se instala el elemento estructural, como se muestra en la matriz siguiente.

$$\begin{array}{r}
 *0 \ 1 \ 1 \ 1 \ 0 \\
 0 \ 1 \ 1 \ 1 \ 0 \\
 0 \ 0 \ 1 \ 1 \ 0 \\
 0 \ 0 \ 1 \ 1 \ 0 \\
 \mathbf{M} = 0 \ 0 \ 0 \ 0 \ 0
 \end{array}$$

Por ejemplo, si consideramos el arreglo X con los pares (i , j) cuyos valores sean igual a ‘1’ en la matriz M, obtenemos el conjunto de la ecuación (4) :

$$X = \{(0,1),(0,2),(0,3),(1,1),(1,2),(1,3),(2,2),(2,3),(3,2),(3,3)\}.....(4)$$

La transformación se aplicaría a X con un elemento estructural a elegir como algunos de los mostrados:

$$\begin{array}{ccc}
 1 & 1 & 1 \\
 1 & *1 & 1 \\
 1 & 1 & 1
 \end{array}$$

Figura Cuadrado

$$\begin{array}{ccc}
 & & 1 \\
 & 1 & *1 \ 1 \\
 & 1 & 1
 \end{array}$$

Figura Rombo

$$\begin{array}{ccc}
 & & & & 1 \\
 & & & 1 & *0 \ 1 \\
 & & & 1 & 1
 \end{array}$$

Figura Horizontal

Se puede elegir una figura arbitraria (‘arbitrary’), el valor marcado con “ * ” se colocará encima del pixel que tratamos y será quien otorgue el valor final después de las operaciones lógicas descritas anteriormente, para evitar que los valores de cada matriz en análisis se vea alterada, previamente se incrementará una fila y una columna a la matriz original (M) ya sea de ceros,

unos o repitiendo el valor de las primeras columnas para las columnas y la primera fila en las filas, proceso que debe invertirse al concluir el proceso de deslizamiento del elemento estructural.

Existen cuatro tipos de transformaciones morfológicas:

1. Dilatación (Referencia en Matlab: 'imdilate')

La dilatación puede interpretarse como una transformación en la que los píxeles del fondo que están tocando al objeto cambian, mientras el resto queda igual. Se designa por el operador \oplus , se puede representar matemáticamente por la fórmula (5) donde B es el elemento estructural y X la matriz procesada, b son los elementos de B y x de la matriz X.

$$X \oplus B = \{d \in E^2 : d = x + b \text{ para cada } x \in X \text{ y } b \in B\} \dots \dots \dots (5)$$

Se va recorriendo la imagen y donde nos encontramos un 1 situamos el origen del elemento estructural. Se realiza la unión de la imagen con el elemento estructural en los puntos que coincidan. Si todos los 1 de la imagen coinciden con 1 del elemento estructural entonces marcamos el píxel del origen del elemento con un 1.

2. Erosión (Referencia en Matlab: 'imerode')

La erosión combina dos conjuntos utilizando la sustracción de vectores. Su representación es mediante el operador \otimes . Gráficamente la erosión sería ir recorriendo toda la imagen y donde exista un uno lo compara con el elemento estructural, y si todos los '1' del elemento coinciden con '1' de la imagen se pone '1' en el origen del elemento estructural. Se puede representar matemáticamente por la fórmula (6):

$$X \otimes B = \{d \in E^2 : d + b \in X \text{ para cada } b \in B\} \dots \dots \dots (6)$$

La erosión elimina objetos de poca importancia o muy simples además de simplificar la estructura de los objetos. Objetos complicados mediante erosión pueden quedar descompuestos en objetos más simples.

3. Apertura (Referencia en Matlab 'imopen')

Siendo la apertura combinación de erosión y dilatación, y B un elemento estructural; se puede representar matemáticamente con la siguiente fórmula:

$$X \circ B = (X \otimes B) \oplus B \dots\dots\dots(7)$$

4. Cierre (Referencia en Matlab 'imclose')

Siendo el cierre combinación de erosión y dilatación, y B un elemento estructural; se puede representar matemáticamente con la siguiente fórmula:

$$X \bullet B = (X \oplus B) \otimes B \dots\dots\dots(8)$$

2.3 Hardware

2.3.1 Video cámara de vigilancia:

Es un dispositivo para capturar imágenes y/o sonidos, que luego de procesarla, envía la información a un dispositivo de registro para su emisión en un monitor. Poseen características que determinan su elección, como el tipo de lente variable o fija (medida en mm), leds infrarrojos, tecnología (CCD o CMOS), color, número de líneas que determinan su resolución, entre otras expresadas en su catálogo.

2.3.2 Tecnología CCD y CMOS

Hoy día existen dos tipos de tecnologías utilizadas para la fabricación de

sensores de cámaras digitales, ya sean compactas o réflex. Se trata de los CCD (Charge Coupled Device) o CMOS (Complementary Metal Oxide Semiconductor). Ambos tipos de sensores están formados en su esencia por semiconductores de metal-óxido (MOS) y están distribuidos en forma de matriz. Su función es la de acumular una carga eléctrica en cada una de las celdas de esta matriz. Estas celdas son los llamados píxeles. La carga eléctrica almacenada en cada píxel, dependerá en todo momento de la cantidad de luz que incida sobre el mismo. Cuanta más luz incida sobre el píxel, mayor será la carga que este adquiera.

Aunque en su esencia, los CCD y CMOS funcionan de una manera muy similar, hay algunas diferencias entre ambas tecnologías.

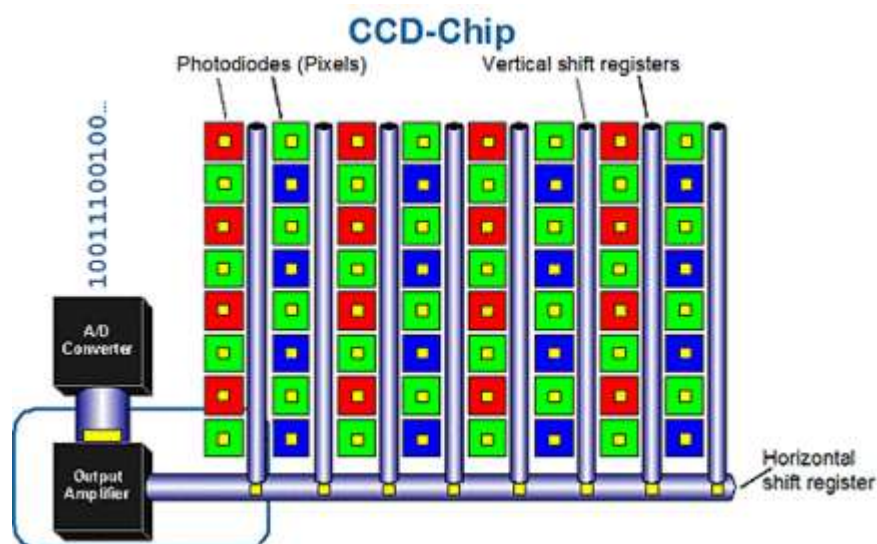


Figura 08 (a): Tecnología CCD [24].

En el caso del CCD, este convierte las cargas de las celdas de la matriz en voltajes y entrega una señal analógica en la salida, que será posteriormente digitalizada por la cámara. En los sensores CCD, se hace una lectura de cada uno de los valores correspondientes a cada una de las celdas. Entonces, es

esta información la que un convertidor analógico-digital traduce en forma de datos. En la figura 08 (a), se muestra la estructura interna del sensor la cual es muy simple, pero se observa como inconveniente la necesidad de un chip adicional que se encargue del tratamiento de la información proporcionada por el sensor, lo que se traduce en un gasto mayor y equipos más grandes.

En el aspecto del rango dinámico, es el sensor CCD el ganador absoluto, pues supera al CMOS en un rango de dos. El rango dinámico es el coeficiente entre la saturación de los píxeles y el umbral por debajo del cual no captan señal. En este caso el CCD, al ser menos sensible, los extremos de luz los tolera mucho mejor.

En cuanto al ruido, también son superiores a los CMOS. Esto es debido a que el procesado de la señal se lleva a cabo en un chip externo, el cual puede optimizarse mejor para realizar esta función. En cambio, en el CMOS, al realizarse todo el proceso de la señal dentro del mismo sensor, los resultados serán deficientes, pues hay menos espacio para colocar los fotodiodos encargados de recoger la luz.

La respuesta uniforme es el resultado que se espera de un píxel sometido al mismo nivel de excitación que los demás, y que éste no presente cambios apreciables en la señal obtenida. En este aspecto, el que un sensor CMOS esté constituido por píxeles individuales, le hace más propenso a sufrir fallos.

En el CCD, al ser toda la matriz de píxeles uniforme, tiene un mejor comportamiento. A pesar de todo, la adición de circuitos con realimentación nos permite subsanar este problema en los CMOS, la tecnología CCD está ligeramente por encima.

En la figura 08 (b) puede apreciarse las características de la tecnología CMOS definidas con anterioridad.

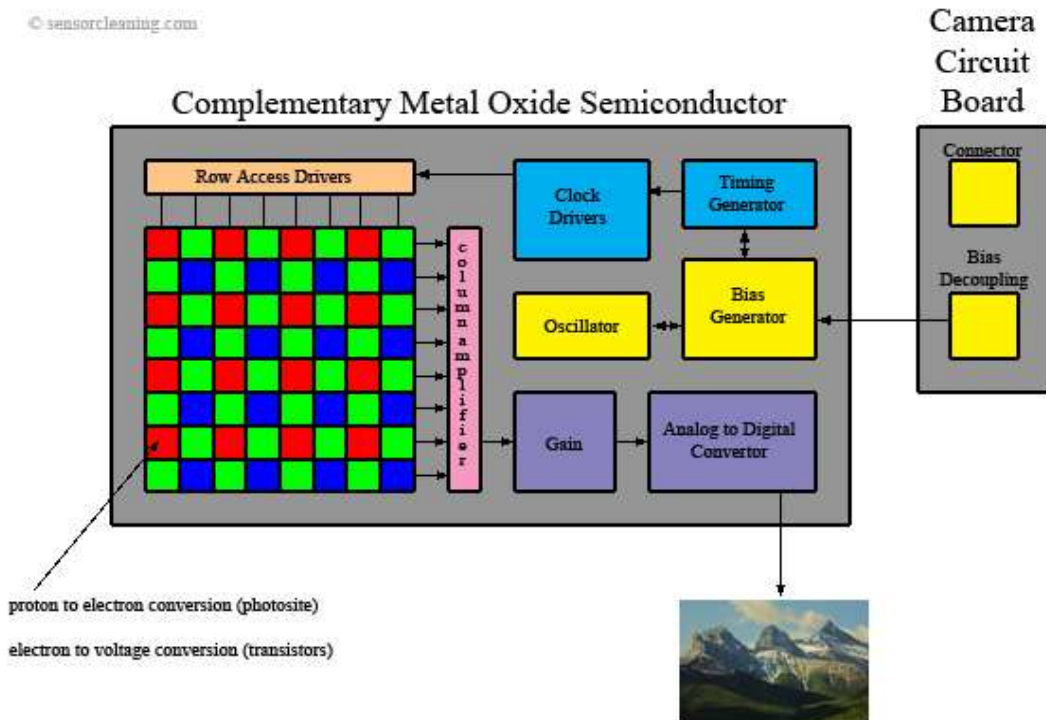


Figura 08 (b): Tecnología CMOS [24].

En el caso del CMOS, aquí cada celda es independiente. La diferencia principal es que aquí la digitalización de los píxeles se realiza internamente en unos transistores que lleva cada celda, por lo que todo el trabajo se lleva a cabo dentro del sensor y no se hace necesario un chip externo encargado de esta función. Con esto se consigue reducir costes y equipos más pequeños.

Además de ofrecer más calidad, los CMOS son más baratos de fabricar precisamente por lo que se comentó anteriormente. Otra de las grandes ventajas es que los sensores CMOS son más sensibles a la luz, por lo que en condiciones pobres de iluminación se comportan mucho mejor. Esto se debe principalmente a que los amplificadores de señal se encuentran en la propia celda, por lo que hay un menor consumo a igualdad de alimentación. Todo lo contrario que ocurría en los CCD.

En cuanto a la velocidad, el CMOS es claramente superior al CCD debido a

que todo el procesado se realiza dentro del propio sensor, ofreciendo mayor velocidad. Es esta una de las principales razones por las que la empresa Casio empezó a imponer los sensores CMOS en sus cámaras y por la cual éstas permiten grabar vídeos a velocidades de hasta 1000 fps.

Otro aspecto en el que los sensores CMOS son superiores a los CCD es en el blooming. Este fenómeno se produce cuando un pixel se satura por la luz que incide sobre él y a continuación empieza a saturar a los que están a su alrededor. Aunque este defecto puede subsanarse gracias a algunos trucos en la construcción, en el caso de los CMOS nos olvidamos del problema.

2.3.3 DVR (Digital Video Record)

Un DVR, es un equipo diseñado para trabajar con cámaras de seguridad, su función es registrar lo que la cámara captura, en un disco duro propio del DVR en formato digital, los tipos de compresión de los equipos DVR son diversos, pero hoy en día el más utilizado es H.264. El DVR puede ser configurado para que grabe por sensor de movimiento, grabación por semanas, por días, grabación 24 horas, entre otras funciones. Los archivos generados por el DVR tienen una extensión .DAV (audio y video digital) que permite registrar amplios tiempos y cámaras de vigilancia.

2.3.4 PC

Una computadora es una máquina electrónica usada para procesar todo tipo de información, como guardar datos, imágenes, escribir cartas, comunicarse, emitir videos, dibujar, hacer informes, crear programas. Consta de 02 partes: hardware y software, además de puertos de comunicación al exterior donde se conectan periféricos como el monitor de video, impresoras, entre otros.

2.3.5 Puertos de comunicación serial, paralelo y USB

Un puerto serial, es una interface de comunicaciones de datos digitales, frecuentemente utilizado por computadoras (PC) y periféricos, donde la información es transmitida bit a bit, enviando un solo bit a la vez; en contraste con el puerto paralelo que envía varios bits simultáneamente. Su símbolo es representado en la figura 09.USB Significa ("Universal Serial Bus") ó su traducción al español es línea serial universal de transporte de datos. Es básicamente un conector rectangular de 4 terminales que permite la transmisión de datos entre una gran gama de dispositivos externos (periféricos) con la PC, por ello es considerado puerto; mientras que la definición de la Real Academia Española de la lengua es: "toma de conexión universal de uso frecuente en las computadoras" [26].

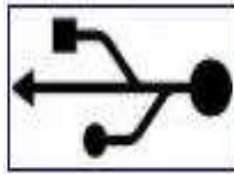


Figura 09: Símbolo del USB [27].

2.3.6 Interface Arduino

Arduino es una plataforma de hardware libre, basada en una placa con un microcontrolador y un entorno de desarrollo, diseñada para facilitar el uso de la electrónica en proyectos multidisciplinarios.

El hardware consiste en una placa con un microcontrolador Atmel AVR y puertos de entrada/salida. Los microcontroladores más usados son el Atmega168, Atmega328, Atmega1280, y Atmega8 por su sencillez y bajo coste que permiten el desarrollo de múltiples diseños. Por otro lado el software consiste en un entorno de desarrollo que implementa el lenguaje de programación Processing/Wiring y el cargador de arranque que es ejecutado

en la placa. Se programa en el ordenador para que la placa controle los componentes electrónicos.

Desde octubre de 2012, Arduino se utiliza también con microcontroladores CortexM3 de ARM de 32 bits, que coexistirán con los más limitados, pero también económicos AVR de 8 bits. ARM y AVR no son plataformas compatibles a nivel binario, pero se pueden programar con el mismo IDE de Arduino y hacerse programas que compilen sin cambios en las dos plataformas. Eso sí, los microcontroladores CortexM3 usan 3.3V, a diferencia de la mayoría de las placas con AVR, que generalmente usan 5V. Sin embargo, ya anteriormente se lanzaron placas Arduino con Atmel AVR a 3.3V como el Arduino Fio y existen compatibles de Arduino Nano y Pro como Meduino en que se puede conmutar el voltaje.

Arduino puede tomar información del entorno a través de sus entradas analógicas y digitales, puede controlar luces, motores y otros actuadores. El microcontrolador en la placa Arduino se programa mediante el lenguaje de programación Arduino (basado en Wiring) y el entorno de desarrollo Arduino (basado en Processing). Los proyectos hechos con Arduino pueden ejecutarse sin necesidad de conectar a un ordenador.

También cuenta con su propio software que se puede descargar de su página oficial que ya incluye los drivers de todas las tarjetas disponibles lo que hace más fácil la carga de códigos desde el computador.

Arduino se puede utilizar para desarrollar objetos interactivos autónomos o puede ser conectado a softwares tal como Adobe Flash, Processing, Max/MSP, Pure Data. Una tendencia tecnológica es utilizar Arduino como tarjeta de adquisición de datos desarrollando interfaces en software como JAVA, Visual Basic y LabVIEW. Las placas se pueden montar a mano o adquirirse.

En la figura 10 puede apreciarse en la parte superior los terminales pin de ingreso y/o salida de señales de forma continua (5 voltios), mientras que en la parte inferior derecha (A0 al A5) son terminales de ingreso de señales analógicas; y el componente más representativo es el microcontrolador anteriormente descrito.



Figura 10: Placa arduino UNO R3 [29].

2.3.7 Circuito de control de relé con transistor bipolar

Los circuitos electrónicos de control, realizados con componentes de tecnología TTL, CMOS, LVTTTL, LVCMOS, soportan cargas en sus terminales que no superan los 40 mA, caso contrario sufren severos daños, por esta razón es necesario desarrollar circuitos complementarios como el mostrado en la figura 11, que controla la activación de un relé. Se describe un breve análisis de la función de cada componente el cálculo de resistencia base según tensiones de trabajo y potencia de relé. Un modelo sencillo para que un circuito electrónico de control pueda activar un relé es con transistor NPN, observando el circuito de la figura 11, vemos que cuando el transistor conduce, el terminal del relé se va a tierra. El valor del voltaje de control suele ser en un nivel bajo

'0' voltios y en un nivel alto '5' voltios.

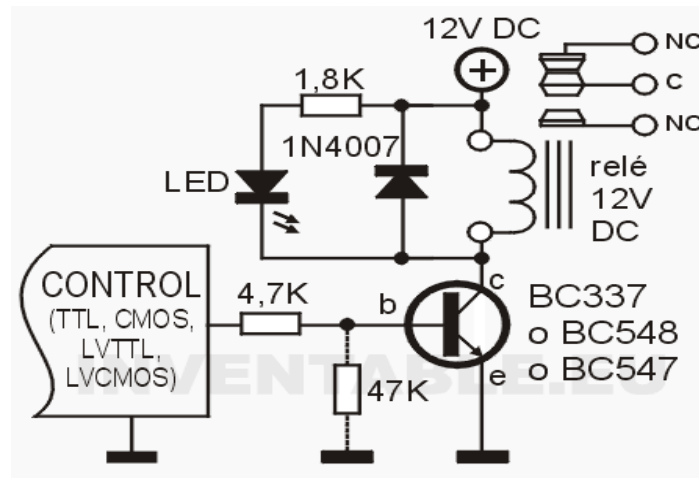


Figura 11: Circuito de control del relé con transistor bipolar [30].

En el primer caso el transistor no se encuentra polarizado y por cuanto no circulará corriente entre emisor y colector sin llegar a activar el relé; en el segundo caso es superada la tensión de umbral, por lo que circulará corriente entre emisor y colector llegando a activar el relé, con una caída de tensión entre colector y emisor de aproximadamente 0.2v [31].

Lo ideal en este tipo de circuitos es trabajar con 02 fuentes de alimentación diferentes, una para alimentar el circuito del relé y otra de control proveniente del circuito electrónico TTL, de esta manera se evita que se dañe el circuito electrónico cada vez que se desactiva el relé, debido a las interferencias generadas por la bobina de este, la cual genera una extra-tensión por el campo magnético inducido de polaridad opuesta. Para evitar esto es que se utiliza el diodo rectificador en paralelo al relé, de tal manera que absorba los picos. Por una cuestión estética se agrega un led de 3mm que prenderá cada vez que el relé se active protegiéndolo con una resistencia en serie de 1.8K.

Para el cálculo de la resistencia base se consideró utilizar el transistor

bipolar NPN BC547, cuya I máxima es de 100 mA, h_{fe} de 110 y voltaje umbral 0,6 voltios; dado que el circuito de control tiene un V_{in} de 5 voltios y, considerando que la I del relé no superará los 50 mA y que el h_{fe} será 100 como máximo; entonces:

$$R_{base} = (V_{in} - 0,6) \cdot h_{fe} / I_{relé} = (5 - 0,6) \cdot 100 / 0,05 = 4,4 \text{ K Ohmios.}$$

Cuando los microcontroladores se inicializan, suelen reconocer sus pines de salida entregando diversos valores que pueden confundir al transistor activando el relé, para esto se utiliza una R entre base y tierra lo suficientemente grande para lograr que el circuito se estabilice, este valor es recomendable que sea del orden de los 47K ohmios. Por lo expuesto se recomienda el uso del circuito de la figura 11.

2.3.8 Gabinete de alarma

Un sistema de alarma es un elemento de seguridad pasiva. Esto significa que no evitan una situación anormal, pero sí son capaces de advertir de ella, cumpliendo así, una función disuasoria frente a posibles problemas. Para esto, la alarma tiene que tener conexiones de entrada para los distintos tipos de detectores, y conexiones de salida para activar otros dispositivos que son los que se ocupan de hacer sonar la sirena, abrir los rociadores o cerrar las puertas. Las Conexiones de entrada suelen ser lazos cerrados ó contactos normalmente cerrados (NC).

2.4 Software

2.4.1 Entorno MatLab

MatLab, "laboratorio de matrices", es una herramienta de software matemático que ofrece un entorno de desarrollo integrado (IDE) con un lenguaje de programación propio (lenguaje M). Entre sus prestaciones básicas se hallan:

la manipulación de matrices, la representación de datos y funciones, la implementación de algoritmos, la creación de interfaces de usuario (GUI) y la comunicación con programas en otros lenguajes y con otros dispositivos hardware. El paquete MatLab dispone de dos herramientas adicionales que expanden sus prestaciones, a saber, Simulink (plataforma de simulación multidominio) y GUIDE (editor de interfaces de usuario - GUI). Se pueden ampliar las capacidades de MatLab con las cajas de herramientas (toolboxes) como el Toolbox de Image Processing [32].

2.4.2 Interface gráfica de usuario (GUI)

La interfaz gráfica de usuario, conocida como GUI, es un programa informático que actúa de interface de usuario, utilizando un conjunto de imágenes y objetos gráficos para representar la información y acciones disponibles en la interface. Su principal uso, consiste en proporcionar un entorno visual sencillo para permitir la comunicación con el sistema operativo de un ordenador. GUIDE proporciona un conjunto de herramientas para crear un interface gráfica de usuario (GUI). Estas herramientas simplifican el proceso de diseño y creación del interface gráfico, en la ventana gráfica se pueden disponer de paneles, botones, cuadros de texto y barras de desplazamiento. Genera automáticamente dos ficheros, uno con extensión **.fig** que contiene la información sobre el aspecto visual del interface, y otro fichero **.m** en el que se codifica la respuesta a las acciones del usuario sobre los controles [33].

2.4.3 Toolbox de “Image Processing”

El Toolbox Image Processing, (IPT) proporciona un amplio conjunto de funciones para la manipulación de imágenes, el análisis, la imagen digital, la visión por ordenador y la imagen digital. IPT incluye archivos de imagen de E/S las transformaciones del espacio de color, filtros lineales, morfología

matemática, análisis de textura, reconocimiento de patrones, estadísticas de imagen y otros. El IPT contiene un manual de referencia completo con descripciones matemáticas de varios algoritmos y más de 100 ejemplos de código de los usos de las funciones. Los comandos más utilizados en este proyecto de investigación se describen en la Tabla 01 [34]:

Instrucciones	Acción a realizar
vision.VideoFileReader	Recupera un video del directorio Matlab
strel('arbitrary',ones(6))	Arreglo matricial usado como elemento estructural
clear	Limpiar la ventana de datos
while – end	Ciclo condicionado de ejecución de comandos
~isDone(.....video.....)	Manipula video mientras este no concluya
step	Recupera un cuadro de la secuencia de imágenes
mod (a,b)	Operación matemática que muestra el residuo
im2uint8(image)	Cambia formato de datos de imagen a (UINT8)
bitand (imagen,128)	Extrae plano de bits más significativo si es un byte.
imclose (A1,EE) :	Transformación morfológica tipo cerradura
im2double	Cambia formato de datos de imagen a (DOUBLE).
for.....end	Ciclo de ejecución de comandos condicional.
figure()	Crea una ventana para mostrar imágenes
subplot	Crea múltiples ventanas como un arreglo matricial
imshow	Comando usado para mostrar una imagen.
title(' ')	Imprime en parte superior de la ventana caracteres
A(:,:,1)	Definición de una matriz A con componente ROJO
release(video)	Cierra archivo de video anteriormente recuperado
pause()	Matlab se detiene lo expresado en segundos

Tabla 01: Comandos MatLab utilizados frecuentemente.

2.4.4. Comandos de MatLab para arduino

Los comandos descritos en la Tabla 02 permiten al ordenador reconocer al interface arduino en su terminal USB asignándole un número de “COM”.

Instrucciones	Acción a realizar
Install_arduino	Instalación virtual de placa Arduino en PC.
a=arduino(COM10)	Asigna valor de sujeto para ‘a’.
a.pinMode(Nº,OUTPPUT)	Asigna cualidad al pin Nº del sujeto ‘a’.
a.digitalWrite(Nº, 0)	Asigna un valor LOW al puerio del sujeto ‘a’.

Tabla 02: Comandos MatLab para arduino.

2.4.5. Comandos de programación para arduino UNO r3

Los comandos de la Tabla 03 forman parte de un microprograma que se registran en el microcontrolador a través de Matlab, generando un control de sus pines de ingreso y salida de datos independientemente del ordenador.

Instrucciones	Acción a realizar
Serial.begin	Estable el ratio de comunicación serial.
pinMode	Designará una cualidad a un número de pin.
digital Write	Asigna un nivel (HIGH o LOW) en su terminal pin elegido.
DigitalRead :	Lee el nivel presente en su terminal a mencionar.
Delay	Comando para hacer una pausa y se expresa en milisegundos.
LOW	Nivel Bajo o ‘0’ lógico (0 voltios DC).
HIGH	Nivel Alto o ‘1’ lógico (5 voltios DC).
void setup	Estructura necesaria para definir condiciones de los terminales.
void loop	Estructura necesaria para las instrucciones del programa.
if	Condicional que agrupa instrucciones a ejecutarse.

Tabla 03: Comandos de programación para Arduino.

CAPÍTULO III: TÉCNICAS DE PROCESAMIENTO Y ALGORITMOS DE RECONOCIMIENTO

Esta investigación se desarrolló aplicando una secuencia de técnicas de procesamiento espacial, empezando por la captura de imágenes de la recepción de un edificio multifamiliar donde se encuentra el personal de vigilancia, con una cámara de video vigilancia con tecnología CCD de 700 líneas de televisión y lente variable; las imágenes fueron registradas en un DVR Nova de 04 canales de grabación, del cual se extrajo una secuencia de aproximadamente 120 segundos de video, consistente en 3600 cuadros de grabación en un USB el cual se descargó en un ordenador que contiene la aplicación MatLab y un conjunto de instrucciones para procesar imágenes como es el Toolbox de Image processing; para luego aplicar en uno de los cuadros técnicas de extracción de plano Rojo, extracción de plano de bits más significativo, inversión de imagen, recorte de imagen y transformaciones morfológicas, que dieron como resultado una silueta binarizada, a partir de la cual se elaboró un algoritmo matemático consistente en contar bits significativos (unos) en perfiles de línea creados y en áreas específicas donde se desplazan los brazos, interpretando así la intención del usuario.

El análisis inicialmente se hizo en imágenes ó cuadros estáticos, analizando los atributos generados en cada paso del procesamiento espacial, observándose mejores resultados con la componente de color Rojo, al igual que en las transformaciones morfológicas dio como mejor resultado los elementos estructurales en transformaciones morfológicas tipo cerradura.

Una consideración muy importante fue la iluminación artificial, dado que la luz natural varía con el tiempo mientras que la artificial es constante en el transcurso del mismo; adicionalmente en pruebas realizadas se observó que una iluminación ligeramente superior o más intensa detrás del sujeto de

análisis, crea un plano de bits con mayores niveles de grises, y al seleccionar el plano de bits más significativo discrimina la parte posterior creando una silueta color negra sobre un plano blanco, con algunos pixeles blancos que son eliminados posteriormente con una transformación morfológica.

Los gestos corporales elegidos se debe a que representan un lenguaje inconsciente, es decir no necesita de previa capacitación, gestos que representan la condición de algún evento u ocurrencia.

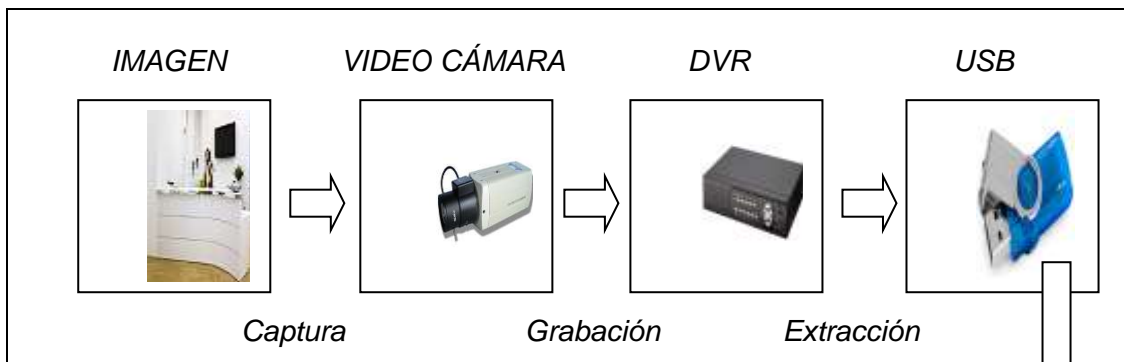
Previamente se muestra un diagrama de bloques para su mejor entendimiento, graficando los equipos utilizados, necesarios para el procesamiento de imágenes, y la aplicación de técnicas de procesamiento espacial, que presentan la imagen con atributos óptimos para desarrollar un algoritmo matemático que logra reconocer los tres gestos mencionados .

Posteriormente se plasma el procesamiento espacial, en un diagrama de flujo, cuyos resultados finales son entregados a una interface externa que cuenta con elementos de comunicación y protocolos necesarios para la recepción de señales eléctricas a través de un puerto de comunicación, como es el USB.

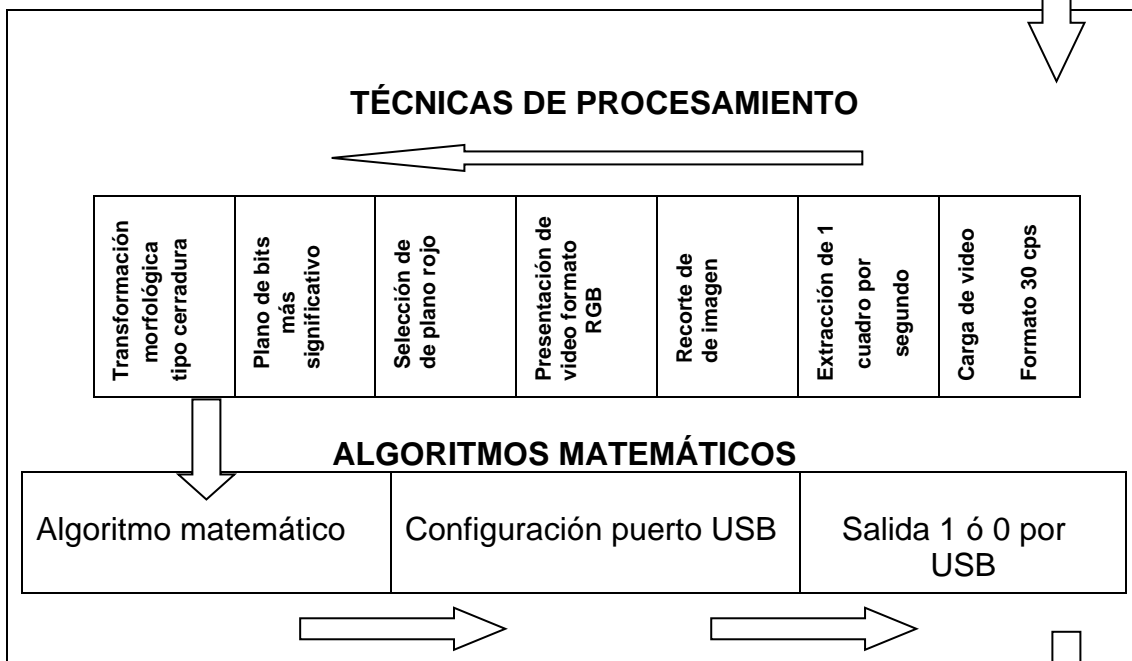
3.1 Diagrama de bloques general:

En la figura 12 se muestra el proceso descrito con anterioridad en 03 bloques: Captura de imagen, Ordenador e Interface externa, describiendo los equipos y/o procesos que los componen guiados por una señal, culminando con una sirena que forma parte de la central de alarma.

EQUIPOS DE CAPTURA DE IMÁGENES



COMPUTADORA PERSONAL Y MATLAB



INTERFACE EXTERNA

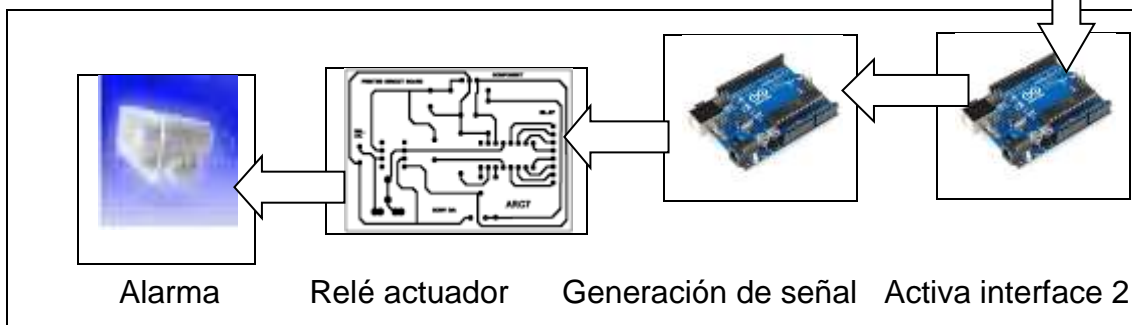


Figura 12: Diagrama de bloques.

3.2 Gestos corporales específicos

Para el desarrollo de este proyecto se realizaron encuestas previas con la finalidad de determinar las “Ocurrencias que vulneran seguridad del edificio” (anexo 01), y que no existe forma de advertirlas dado que el nivel tecnológico no lo permite. Estas ocurrencias se agruparon en los siguientes tipos de gesto:

Tipo 01: Personal de recepción que abandona su puesto sin dejar reemplazo.

Tipo 02: Necesidad de activar alarma por situaciones de peligro en recepción.

Tipo 03: Apagar sistema de alarma desde la recepción.

Para que el algoritmo matemático pueda reconocer estos eventos, es conveniente procesar espacialmente imágenes utilizando gestos de forma inconsciente, no oral y que pertenezca a un lenguaje corporal de significado universal, realizando un movimiento corporal, de tal manera que la comunicación sea eficiente para lograr la recepción del mensaje, estos gestos asociados son:

1. Para el tipo 01: Counter o recepción sin personal.
2. Para el tipo 02: Levantar las 02 manos, a la altura del rostro o por encima de este.
3. Para el tipo 03: Levantar solo 01 mano, cualquiera que sea, a la altura del rostro o por encima de este.

En cualquiera de los casos mencionados el usuario puede adoptar diversas posiciones de trabajo como parado, sentado ó inclinado, y desde allí transmitir el gesto necesario que advierta la situación en que se encuentra, como se muestran en las figuras 13 a la 19, donde se observan diferentes posiciones que adopta el sujeto de análisis.



Figura 13: Counter vacio.



Figura14
Señal de apagar
parado y mano a la
altura del rostro.



Figura15
Señal de apagar
parado y mano
encima del rostro.



Figura 16
Señal de apagar
sentado y mano
encima del rostro.



Figura 17
Señal activar
sentado y manos a la
altura del rostro.



Figura 18
Señal activar
sentado y manos
encima del rostro.



Figura 19
Señal activar
parado y manos a la
altura del rostro.

3.3 Captura de imágenes

Teóricamente el reconocimiento de gestos es el conjunto de técnicas aplicadas a una imagen para que el ordenador reconozca gestos corporales mediante

algoritmos matemáticos aplicados a la imagen procesada espacialmente, usando filtros u operaciones matemáticas para extraer algunos atributos requeridos como contornos, suavizar bordes, eliminación de ruidos, definición de siluetas, entre otros.

El requerimiento para este proyecto es obtener una silueta color blanca, con fondo negro, porque el algoritmo aplicado consiste sumar los bits significativos es decir “unos” en determinadas áreas o líneas de perfil. Para lo cual es necesario eliminar todo pixel negro en la zona interna a la silueta blanca y todo pixel blanco en el fondo del cuadro negro, como se muestra en la figura 20.

A partir de un cuadro a color obtenido por la videocámara; resulta esta silueta y en este estado de procesamiento se puede aplicar los algoritmos matemáticos que definen el estado de alarma que corresponde.

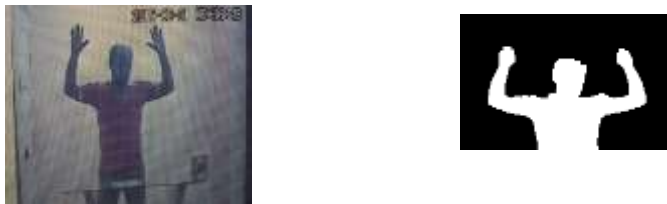


Figura 20: Imagen capturada a color e imagen procesada final.

El procesamiento digital se inicia con la captura de la imagen desde una videocámara de tipo CCD, cuyas características se encuentran en la datasheet del equipo, que se adjunta en la sección anexo 02. La elección de una cámara con lente variable como la mostrada en la figura 21, se debe a la necesidad de ajustar los cuadros de imágenes al sector de interés, como se indica en la figura 22, en la que se muestran algunos atributos como la emisión de la fecha y hora en la parte superior, esta señal es una referencia, que indica el límite superior al cual deben llegar las manos, dado que al aplicar una

transformación morfológica puede anular parte de la información necesaria para ubicar las manos. Se especifica “d” como un dato adicional que permite regular el lente de la cámara quedando a 2/3 el centro de gravedad del sujeto de análisis.



Figura 21: Cámara CCD [35]. Figura 22: Posición de la línea 140.

$$\textcircled{d} = 2/3 \text{ altura total}$$

Las consideraciones que se tuvieron para que el algoritmo desarrollado funcione correctamente, fueron:

1. Una buena iluminación artificial del ambiente posterior al usuario, es decir a su espalda, de manera tal que la silueta del usuario tenga un nivel de gris más bajo que el fondo, de esta manera al seleccionar un plano de bits más significativo, existirá un umbral mayor en el fondo asignando píxeles blancos a este fondo y píxeles negros a la silueta en análisis. La luz natural es variable en el tiempo por lo que no es de buena utilidad para el algoritmo a desarrollar.
2. La línea140 es una característica que permite orientar la videocámara en una posición de manera tal, que la línea en mención, se encuentre por encima del counter, y lo suficientemente cerca a la vez, y por otro lado, el usuario con brazos extendidos, quede dentro del cuadro de análisis, como se muestra en la figura 22.

3. Lente variable entre 2.8 mm y 30 mm para ajustar desde el punto de instalación de la cámara, el cuadro de análisis de interés, lo que significa que la cámara puede estar situada entre 4 a 8 metros de distancia del counter, quedando el sujeto de análisis dentro del cuadro a procesar.

4. Las cámaras fueron de tecnología tipo CCD, ya que estas entregan su información en un formato matricial, ideal para su análisis en el entorno Matlab, las cámaras con tecnología CMOS se pueden incluir en este proceso, con la restricción que esta suelen ser muy sensible a la luz, de allí la preferencia por las cámaras con tecnología CCD.

Las imágenes capturadas por la cámara CCD fueron registradas en un DVR, el mismo que se configuró acorde a las características de la cámara, para esto el DVR cuenta con menús desplegables que permite configurar características como: calidad de imagen (CIF), determinar número de cuadros por segundo a grabar (30cps), la resolución (240x352), entre otras. El formato de video para los DVR es h.264, lo que determina que las grabaciones realizadas se capturan con la extensión .DAV.

En el caso particular de este proyecto, los videos procesados son entregados al ordenador mediante un USB y en formato .DAV, sin embargo el entorno MatLab procesa videos con extensión .AVI, de allí la necesidad de hacer la conversión previa de un formato DAV a AVI. En este caso particular, el reproductor Player para archivos DAV tiene incorporada la función de conversión requerida.

3.4 Técnicas de procesamiento y algoritmo matemático de reconocimiento

En esta sección se describe las técnicas de procesamiento espacial utilizadas durante la investigación con un breve comentario de los resultados obtenidos, cabe resaltar que el proceso se realizó a determinados cuadros, para luego hacerlo extensivo al video capturado, el mismo que está compuesto por una secuencia de cuadros.

3.4.1 Técnicas de procesamiento espacial aplicadas

Los videos digitales están formados por una secuencia de cuadros de imagen a razón de 30 cuadros por segundo para el formato NTSC; para lograr el reconocimiento de los gestos se seleccionó 01 cuadro por segundo de cada 30 cuadros que comprende el video; esta cualidad puede variarse cambiando los parámetros establecidos en el desarrollo del programa, eligiendo 02 cuadros por segundo, solo en caso de situaciones que es necesario analizar movimientos mas rápidos.

Elegido un cuadro, cuya resolución es una característica de la cámara CCD, que es de 240x352 píxeles, donde 240 (valor máximo de x) es el número de columnas y 352 (valor máximo de y) es el número de filas, se representa por una matriz de 240x352, es decir $M(x,y) = M(240,352)$, además $f(x,y)$ es la función que contiene el valor de la intensidad de cada pixel en la coordenada (x,y) , es decir la fila 'x' con la columna 'y'. Cada cuadro está compuesto por 03 matrices independientes: $f_1(x,y)$, $f_2(x,y)$ y $f_3(x,y)$, donde f_1 guarda la intensidad o nivel de grises del color Rojo (R), f_2 la intensidad o nivel de grises del color Verde (G) y f_3 la intensidad del nivel de grises del color Azul (B).

Siendo el objetivo del procesamiento lograr la silueta mostrada en la figura 20, se eligió solo una de las matrices, que en este caso particular fue la que contiene la componente del color Rojo, definiendo la variable 'cuadro' como una matriz $\text{cuadro} = M(240,352,1)$, obteniendo el plano Rojo con niveles de grises con se muestra en la figura 23. También se hizo uso de la instrucción $M(240,352,2)$ para el verde y $M(240,352,3)$ para el azul, donde estas dos últimas no permitieron reconocer en un alto grado como puede observarse en las figuras 3, 4 y 5, pues el nivel de grises en cada componente (RGB) difiere en alto grado entre una componente y otra. De allí la elección de la componente color rojo para obtener mejores resultados. En la figura 23 observamos una mejor definición de la silueta según lo comentado.

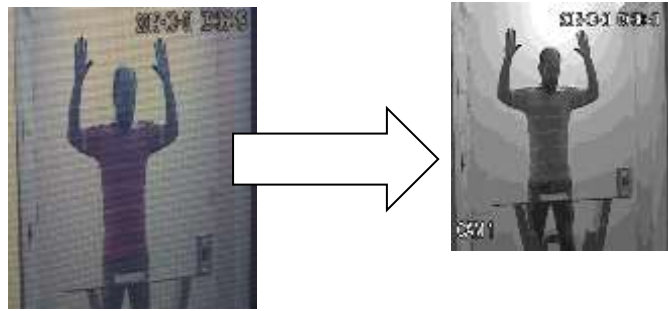


Figura 23: Extracción de plano rojo.

El plano rojo obtenido, contiene múltiples niveles de grises comprendidos entre los valores '0' a '255', por lo que se aplicó una técnica adicional consistente en la extracción del plano de bits más significativo, denominada 'Plano de Bits'. Es necesario que las variables de la imagen estén en formato de bits, la conversión se consigue con el comando `cuadro = im2uint8 (M)`, estando ya en formato de 08 bits (byte) se puede extraer el plano más significativo con la instrucción `cuadro = bitand(M,128)`, obteniendo como resultado lo mostrado en la figura 24. Al elegir el plano de bits más significativo equivale a aplicar un umbral 128, es decir los niveles de grises superiores a 128 toman el valor de 128 y los píxeles con valor menor a 128 tomarán valor cero. Cabe aclarar que los niveles de grises obtenidos en el procesamiento mostrado en la figura 24 son de 0 ó 127, tomando el valor máximo y mínimo.

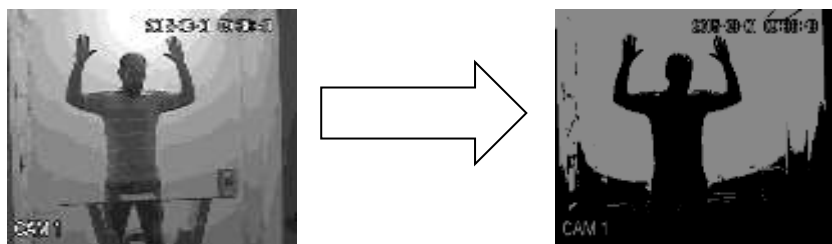


Figura 24: Obtención del plano de bits más significativo.

Cuando se requiere una imagen binarizada, el cuadro tomará valores extremos es decir “0” o “1”, o que en este cuadro equivale a decir “0” o “255”. Para lograr estos niveles debe realizarse una operación de compensación consistente en multiplicar por 2 todos los elementos de la matriz M, esta operación se logra con la instrucción Cuadro = Mx2, obteniendo como resultado una imagen binarizada, es decir con tonos blancos y negros, tal como se muestra en la figura 25.

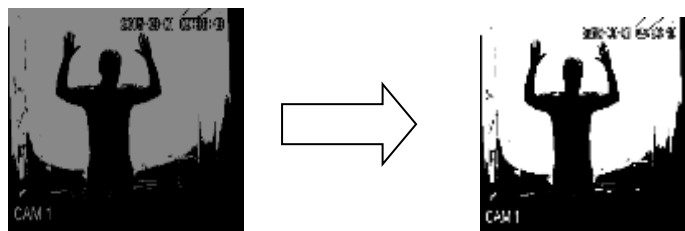


Figura 25: Obtención de imagen binarizada.

Dado que el sujeto de análisis es la silueta binarizada, existe la necesidad de eliminar información innecesaria en el cuadro, la misma que se encuentra alrededor del sujeto de análisis. Estos píxeles se eliminaron en gran parte, realizando un recorte de la imagen, para lo cual elegimos las columnas comprendidas entre 45 y la 258, y las filas comprendidas entre la 0 y 140, estos datos se obtuvieron durante la ejecución del aplicativo Matlab, deslizando el cursor sobre el cuadro de análisis, donde se indica las coordenadas de cada pixel, este procedimiento se logra definiendo la variable cuadro = M (1:140,45:258,1); obteniendo como resultado el cuadro de imagen recortado mostrado en la figura 26. La instrucción mencionada exhibe los píxeles comprendidos entre la fila 1 y la 140, y las columnas comprendidas entre la 45 y 258. Estos parámetros son estables por lo que se debe tener las consideraciones mencionadas para ubicar adecuadamente la cámara en una posición frontal y una buena regulación del lente variable.

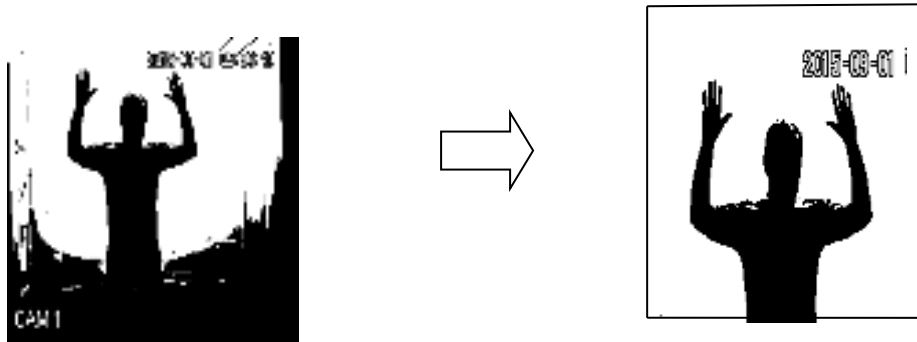


Figura 26: Recorte de imagen.

En la figura 26 obtenida se presencia píxeles blancos dentro de la silueta negra y píxeles negros en el fondo blanco, siendo una imagen que dificulta la aplicación de algún algoritmo matemático, para eliminar estos píxeles se aplicaron diferentes técnicas de transformación morfológica como erosión, dilatación, apertura, y cerradura, obteniendo mejores resultados con la transformación tipo cerradura, utilizando un elemento estructural arbitrario conformado por una matriz 6x6 de unos en forma de cuadrado, ya que matrices de menor orden no eliminaban por completo algunos píxeles innecesarios, obteniendo la imagen mostrada en la figura 27. Este proceso se logra mediante las instrucciones $\text{Cuadro} = \text{imclose}(M, EE)$, donde EE es el elemento estructural, la forma elegida es un cuadrado de 6x6. Según se muestra en la figura 27 también se logró eliminar un atributo del DVR como es la visualización de la fecha y hora quedando en forma óptima para aplicar el algoritmo matemático que realice el conteo de bits.

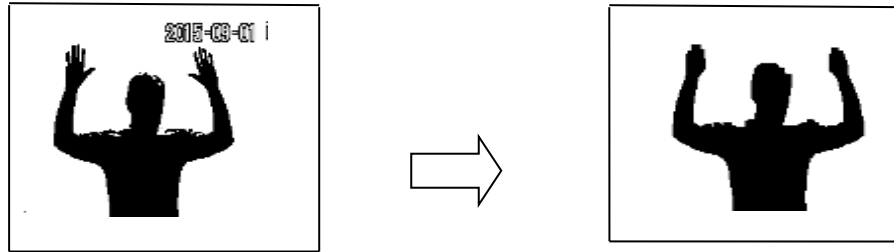


Figura 27: Transformación morfológica.

Para la aplicación de algoritmos matemáticos es más ventajoso obtener la imagen inversa, es decir, que la silueta sea de color blanco y un fondo negro, obteniendo el resultado deseado mediante la operación matemática: $\text{Cuadro} = 255 - M$, mostrado en la figura 28.

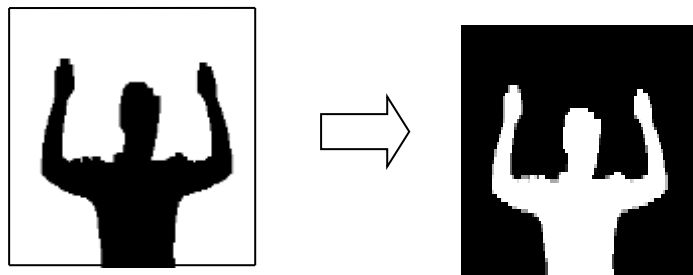


Figura 28: Imagen inversa.

3.4.2 Algoritmo matemático de reconocimiento

Los videos digitales están formados por una secuencia de cuadros de imagen a razón de 30 cuadros por segundo para el formato NTSC; para lograr el reconocimiento de los gestos es solo necesario escoger 01 cuadro por segundo

de cada 30 cuadros que comprende el video.

El algoritmo matemático que llegó a reconocer los gestos anteriormente descritos, a partir de la imagen inversa de la figura 28, y después de un análisis de cada cuadro del video se determinó conveniente dividir los cuadros en 03 columnas de la siguiente manera :

Columna 01: comprendida entre los píxeles de la columna 01 a la 72
Es la zona donde se desplaza el brazo derecho de arriba hacia abajo.

Columna 02: comprendida entre los píxeles de la columna 73 a la 142
Es la zona donde se desplaza el brazo izquierdo de arriba hacia abajo.

Columna 03: comprendida entre los píxeles de la columna 143 a la 214
Es la zona donde se desplaza el rostro de arriba hacia abajo.

En las columnas de píxeles, (derecha, centro e izquierda) se procede a aplicar un algoritmo en la columna central, consistente en recorrer de arriba hacia abajo y de izquierda a derecha como diente de sierra, donde se irá sumando en un contador el valor de cada pixel, es decir sumando Ceros. Cuando el valor del pixel sea diferente de Cero, es porque habrá recorrido la fila que pasa por la parte superior del rostro, en este momento, la fila tomará un valor equivalente a $d = f(\text{rostro})$.

A partir de la detección de esta fila $d = f(\text{rostro})$ se crea una segunda situada 20 filas mas abajo (20 píxeles), es decir $d+20 = f(\text{rostro}) + 20$. Esta última fila será la a analizar en cada cuadro, la misma que cambia de valor si el usuario se sienta o se inclina, ya que constantemente de cuadro en cuadro, el algoritmo está calculando el valor de $f(\text{rostro})+20$. Por lo descrito con anterioridad obtendremos un cuadro con atributos como se muestra en la figura 29.

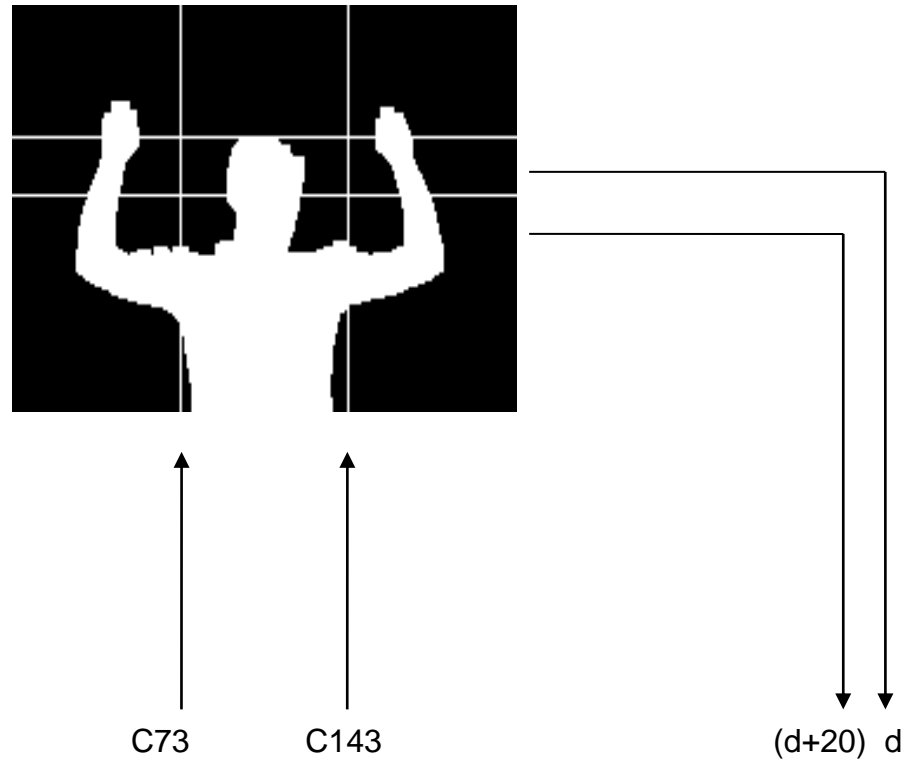


Figura 29: Aplicación de algoritmos.

Tomando como referencia la línea $f(\text{rostro})+20 = d+20$, el algoritmo analiza la presencia de bits con valor UNO en su recorrido. De tal forma que se presenta los siguientes condicionales:

1. Si solo la columna derecha contiene píxeles blancos, es porque levantó el brazo izquierdo, y se está solicitando que la alarma se apague.
2. Si solo la columna izquierda contiene píxeles blancos, es porque levantó el brazo derecho, y se está solicitando que se apague la alarma.
3. Si ambas columnas derecha e izquierda contienen píxeles blancos, es porque se está solicitando que se active la alarma.
4. Cuando el valor de $f(\text{rostro}) + 20$ excede el valor de 120, el sistema deja de calcular e interpreta que al no encontrar un rostro, es porque el usuario se ausentó de su puesto emitiendo una señal de activación de alarma.

El rango de valores para un rostro en la columna central varía entre 25 y 40 píxeles, mientras que cada brazo varía entre 5 y 25 píxeles, en caso estos valores excedan los rangos establecidos, el sistema lo interpretará como si un intruso a ingresado al área de procesamiento de imágenes emitiendo una señal de alarma.

En la figura 30 se muestra un diagrama de flujo del algoritmo desarrollado y ejecutado en el ordenador para reconocer los tres gestos corporales a partir de la imagen binarizada. El código desarrollado a partir de este diagrama de flujo que se adjunta en el anexo 03 “Programa computacional desarrollado en Matlab y arduino” se inicia con la asignación de variables de los diferentes contadores que intervienen en el reconocimiento. El proceso se inicia con la lectura del video que se carga con un módulo USB para luego recortar la imagen, de esta manera se reduce el número de píxeles a analizar, ahorrando un tiempo considerable en el procesamiento espacial posterior. Los 03 contadores que se muestran son los que realizan el conteo de bits en un perfil de línea que cubre las 03 áreas donde se deslizan las extremidades del usuario. Posteriormente los resultados y valores adoptados son llevados a un cuadro que contiene un grupo de condicionales, y de acuerdo al número de bits registrado, el algoritmo tomará la decisión de enviar un “1” o un “0” al pin 13 de la plataforma arduino conectada en el terminal USB del ordenador.

El algoritmo hace uso de un grupo de instrucciones de Matlab especialmente desarrolladas para el entorno Arduino.

3.4.3 Diagrama de flujo:

Como resultado del algoritmo matemático descrito con anterioridad, representamos el proceso en el diagrama de flujo a continuación para poder desarrollar el código fuente en el entorno Matlab.

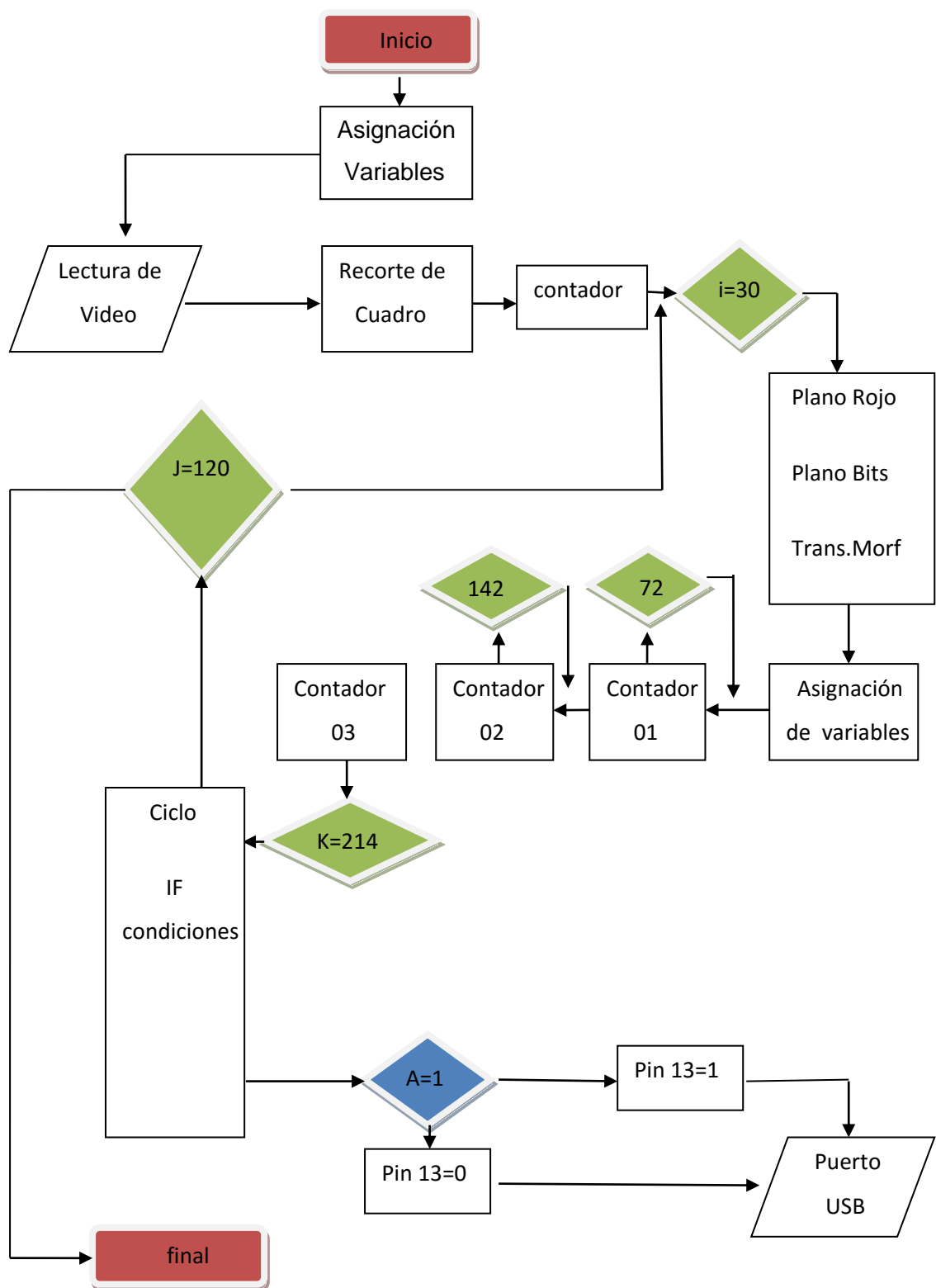


Figura 30: Diagrama de flujo.

3.4.4 Configuración de puerto USB

La conexión física entre una interface externa y un puerto serial, paralelo o USB de la PC requiere de un protocolo de comunicación, para la transmisión y/o recepción de datos entre ambos hardware. El puerto de comunicación elegido para este proyecto es el USB, dado que se encuentra en cualquier computador personal y su uso es muy frecuente. Elegido el tipo de puerto, se diseñó un hardware adecuado que contaba con la conexión física tipo USB y el software que con un protocolo de comunicación compatible con MatLab y los comandos necesarios para el desarrollo de un conjunto de instrucciones que nos permitió enviar una señal, como resultado del procesamiento espacial y el algoritmo matemático de las imágenes procesadas.

Por lo expuesto anteriormente se procedió a utilizar la plataforma de prototipos electrónicos de código abierto Arduino UNO R3, basado en hardware y software flexibles, de fácil entendimiento. Su Hardware está basado en microcontroladores AT-MEGA8 ó AT-MEGA168 de Atmel y su software, en el lenguaje de programación AVR C, que puede ser expandido con librerías C++.

Para poder realizar el enlace Arduino – MatLab, fue necesario descargar y descomprimir tres archivos en nuestro computador personal, estos son:

1. Arduino 1.0.4 windows: contiene el software para Arduino e instrucciones para elaborar programas en Arduino, en la página Arduino: http://arduino.googlecode.com/files/arduino_1.0.4_windows.zip.
2. ArduinoIO: programa extensión de MatLab para realizar enlace con Arduino, ya que MatLab actuará como administrador del dispositivo, se ubica como: http://www.mathworks.com/matlabcenter/fileexchange/32374_matlab_support_package_for-arduino_aka_arduinoIO.zip.

3. Drivers: descargando de la página Arduino.Home Page 'Arduino.cc' haciendo click en la barra download, donde se eligió la opción Windows Installer (zip). Versión en Inglés. Se recomienda deshabilitar cualquier puerto serial.

El proceso de instalación para un correcto funcionamiento se lleva a cabo de la siguiente manera:

1. Abrir MatLab, descargar y descomprimir los dos archivos mencionados.
2. Abrir la carpeta de archivos 'arduino_1.0.4' y luego el símbolo aplicación 'arduino'.
3. Conectar físicamente la placa Arduino en el puerto USB.
4. Verificar a que puerto de comunicación 'COMX' está conectado, caso contrario actualizar el controlador mediante el siguiente proceso: En 'Inicio', situarse en 'Equipos' y dar click derecho para elegir 'Propiedades', luego abrir 'Administrador de Dispositivos' y elegir 'Otros dispositivos', de no encontrarse Arduino aquí, es porque ya tiene un puerto COM asignado en 'Puertos COMX'; de encontrarse arduino en 'otros dispositivos', debemos encontrar su COMX haciendo click derecho en 'Arduino UNO' para abrir 'Actualizar Controlador' y luego la opción 'Buscar Software de Controlador de Equipo'; haciendo click en 'examinar', en este momento se apertura el directorio donde abrimos la carpeta de archivos 'arduino 1.0.4' para solo elegir 'Drivers' y luego 'Aceptar' ó también presionar 'siguiente' y la opción 'Instalar este software de controlador de todas formas'. De esta manera arduino pasará de la sección 'Otros dispositivos' a 'Puertos COM' donde se verifica el número de COM asignado.
5. Ir a la aplicación abierta 'arduino', para elegir la tarjeta Arduino UNO y

el puerto serial asignado COMX en windows, los que se encuentran en la opción 'Herramientas' de la barra superior, que al desplegarse se elige 'Tarjeta' ó 'Puertos', y el dato a cambiarse.

6. Para una conexión virtual eficiente entre arduino y MatLab, se introduce un sketch en la placa Arduino, eligiendo en la barra de la aplicación Arduino la opción 'Archivo' y luego 'Abrir' donde se elige el archivo 'adio.pde' que se encuentra en: 'ArduinolO>pde>adiosrv>adiosrv.pde ó adio.pde'. Habiendo cargado el archivo adio.pde en la aplicación Arduino, elegir 'Cargar' de la opción 'Archivo' ó presionar la flecha verde, esperando a que aparezca la frase 'carga terminada'.
7. Se procede a cerrar la aplicación Arduino y se abre MatLab como administrador desde Inicio, haciendo click derecho en 'MatLab' para elegir la opción 'Ejecutar como Administrador'.
8. En la aplicación MatLab se procede a abrir el directorio 'ArduinolO' en el current folder para luego correr (run) las siguientes instrucciones:

```
>> install_arduino      % instalación virtual de Arduino con MatLab%
>> a = arduino('COM11') % se crea la variable a que es la tarjeta
arduino%
% a partir de aquí, todo comando se precederá por la abreviatura 'a.' %
      Sólo, si esta referido a la tarjeta arduino
```

Ejemplo, si queremos declarar el pin 13 de la plataforma arduino como salida:

```
Antes (desde aplicación Arduino): >> pinMode (13,'output');
```

```
Ahora (desde aplicación MatLab): >> a.pinMode (13,'output');
```

Resultado de esta conexión física y virtual, el programa computacional desarrollado para este proyecto, que se expone en el Anexo 04, el ordenador transmite a la plataforma Arduino una señal de Nivel Bajo (LOW) (0 voltios) cuando se desea apagar alarma, ó Nivel alto (HIGH) (5 voltios) cuando se desea activar alarma. Esto se logra con los comandos:

```
a.digitalWrite(13,LOW); % para apagar alarma %
```

```
a.digitalWrite(13,HIGH); % para activar alarma %
```

CAPÍTULO IV: DISEÑO Y PROGRAMACIÓN DE INTERFACE EXTERNA

La interface de alarma tiene como propósito capturar las señales transmitidas por el ordenador en su puerto USB, procesarlas y luego generar 02 tipos de señales: apagar o activar. Para su diseño considerar los siguientes aspectos:

1. La interface cuenta con un protocolo de comunicación, que permite la transmisión y recepción de datos con el ordenador mediante el puerto USB, en forma ordenada.
2. La interface genera 02 tipos de señales eléctricas, consistente en un tren de pulsos con magnitudes de 0 y 5 voltios DC y frecuencias predeterminadas.
3. Se elige la plataforma Arduino, porque su microcontrolador cumple con los requerimientos mencionados en los puntos 1 y 2.
4. La plataforma arduino obedece a un conjunto de instrucciones que provienen del ordenador ó de su microcontrolador, pero no en forma simultánea, lo que nos obliga a trabajar con 02 plataformas; la primera que ejecute instrucciones proveniente del ordenador y la segunda que ejecute instrucciones proveniente de su microcontrolador.
5. Los componentes externos que se acoplen a la placa arduino no deben consumir corrientes superiores a los 40 mA.
6. El gabinete de alarma se activa cuando su terminal pasa de un estado normalmente cerrado (NC) a un estado normalmente abierto (NA) y se apaga en un proceso inverso. Esta cualidad es propia de un relé.

Dado que su consumo aproximado es de 40 mA, no se debe acoplar directamente a la plataforma arduino, por lo que es necesario un circuito

adicional que tenga un transistor de bajo consumo, que amplifique la señal emitida por uno de los pines terminales del arduino, con una fuente externa; y de esta manera activar el relé.

Teniendo en cuenta los considerandos mencionados anteriormente, el proceso se subdivide en 04 etapas de interfaces:

4.1 Interface 01: Comunicación de ordenador con plataforma Arduino

El objetivo de esta interface es contar con un protocolo de comunicación que permita la transmisión y recepción de datos en forma ordenada con el ordenador. La plataforma arduino cuenta con un sketch o programa denominado adiosrv.pde ó adio.pde el cual se instala en su microcontrolador cuando este, se configura con la aplicación arduino a través del puerto USB. Habiendo configurado desde MatLab el pin13 de la plataforma arduino como un pin de salida (OUTPUT), MatLab enviará a través del pin 13 del arduino, dos tipos de señales: LOW (0 voltios) cuando se requiera apagar la alarma ó HIGH (5 voltios) cuando se requiera activar la alarma. Esta plataforma no necesita de una fuente de alimentación externa ya que se alimenta del puerto USB. Esta señal se transmitirá a la interface 02.

4.2 Interface 02: Generación de 02 señales eléctricas pulsantes

4.2.1 Tabla de estado de Pines de Plataforma arduino

El objetivo de esta interface es generar señales eléctricas en su terminal pin 08 que se configura como pin de salida. El sketch que se detalla en el Anexo 04, es un microprograma desarrollado para generar las señales eléctrica, el cual se registra en el microcontrolador de la placa arduino desde la aplicación arduino en el ordenador, por lo que se conecta físicamente al puerto USB del ordenador para su configuración, siguiendo los pasos descritos en el ítem 3.4.4.

El programa compara el estado actual de la alarma (pin 12) con el estado que solicita MatLab (pin 13) y según el resultado, genera un tipo de señal eléctrica que la transmite por el pin 08, tal como se muestra en la Tabla de estado N° 04.

Estado del pin13 (solicitud de MatLab)	Estado del pin12 (Estado actual de Alarma)	Estado del pin 08 (Señal que transmite)
HIGH	HIGH	HIGH
HIGH	LOW	Señal para activar
LOW	HIGH	Señal para apagar
LOW	LOW	LOW

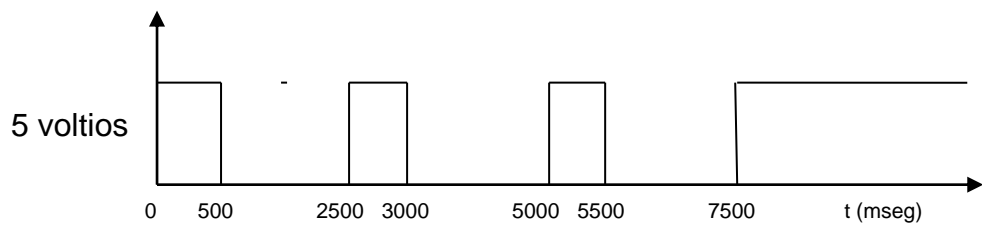
Tabla 04: Tabla de comparación entre pin12 y pin13

La presente tabla establece que el pin 08 conservará su estado cuando estando la alarma activada (HIGH) el ordenador le solicita un estado de activación de alarma (HIGH), mientras que cuando la alarma se encuentra apagada (LOW) y el ordenador le solicita apagarse (LOW) este permanecerá en el mismo estado. Caso diferente es la opción donde se solicita una directiva diferente al actual estado, es decir que estando apagada (pin 12 en LOW)

se requiere activar (pin13 HIGH) o cuando estando activada (pin 12 en HIGH) se requiere desactivarla (pin13 LOW).

Como resultado el pin 8 de la plataforma arduino correspondiente a la interface 02 entregará señales eléctricas como las mostradas en la figura 31, las cuales excitan a un relé conectado con el gabinete de alarma.

Señal para activar Alarma:



Señal para apagar Alarma:

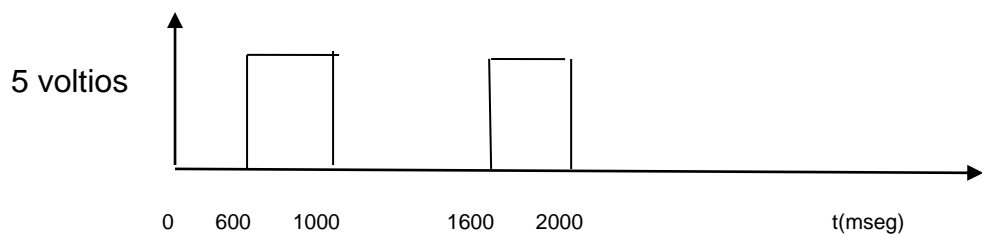


Figura 31: Señales eléctricas generadas en el pin 08 del arduino 02.

4.3 Interface 03: Circuito electrónico para activar relé.

El objetivo de la interface 03 es activar el relé mediante un circuito electrónico propuesto en la figura 32, el cual recibe señales eléctricas cuando es conectado al pin 08 de la interface 2, ya que directamente sin interface daña la plataforma arduino porque el relé tiene un consumo superior a los 40 mA.

De esta manera el relé transmitirá una señal NO (normalmente abierto) ó NC (normalmente cerrado) a través de sus terminales, tal y como se muestra en la figura 32, donde c es el terminal común del relé.

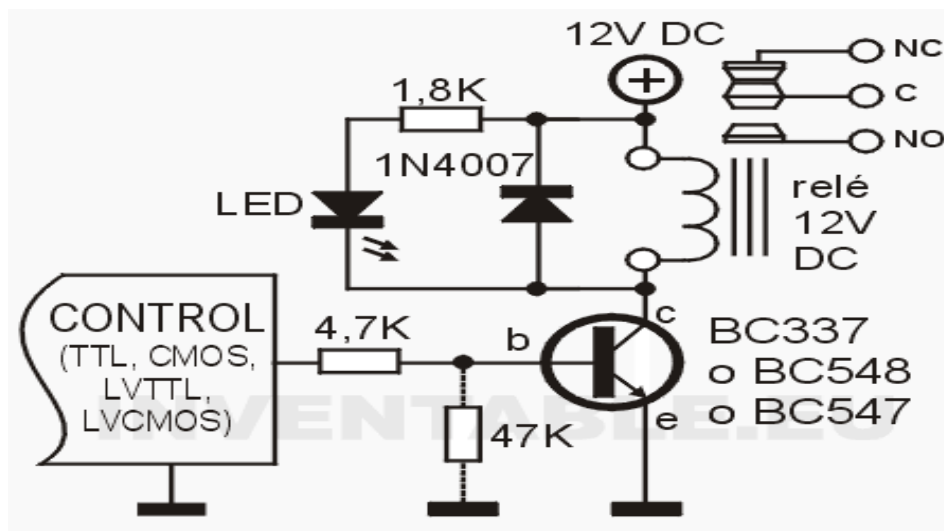
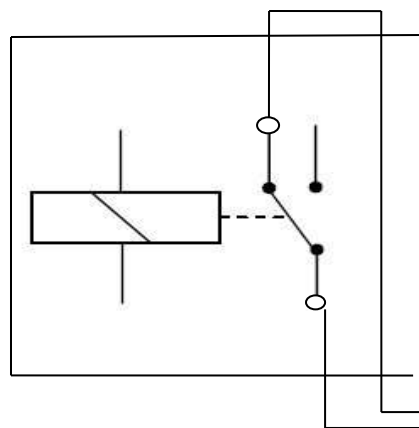


Figura 32: Circuito interface para activar relé [30].

4.4 Interface 04: Conexión eléctrica de relé con alarma

La Central de alarma requiere en sus terminales una conexión de tipo NO para activar la alarma y tipo NC para apagar la alarma. En algunos casos es acompañado por una resistencia en serie, cuyo valor depende de las especificaciones técnicas que se mencionen en el gabinete de alarma. La conexión se realiza directamente con alambre AWG N°24 entre el relé y sus terminales, como se muestra en la figura 33.



Relé



Placa de alarma

Figura 33: Conexión eléctrica entre relé y gabinete de alarma [36].

CAPÍTULO V: PRUEBAS DE CAMPO

En este capítulo se presentan los procedimientos realizados para la implementación física del proyecto, en un edificio multifamiliar.

5.1 Atributos del proyecto implementado

El proyecto implementado tiene ciertas características que determinan el grado de cumplimiento de los objetivos trazados. Estas características ó atributos se describen a continuación:

5.1.1 Eficiencia del Sistema de seguridad:

Eventos como los descritos en la encuesta a administradores de edificios del anexo 01, han sido resueltos en un 62.5 %, es decir 10 de los 16 eventos cotidianos que vulneran la seguridad, los mismos están relacionados con el abandono de puesto e implementación de sistema de alerta ó respuesta inmediata ante un evento que vulnera la seguridad del edificio. Como resultado de lo expuesto se aumentó a 81.25 % la eficiencia del sistema de seguridad.

5.1.2 Uso de ordenadores de baja velocidad:

Para el desarrollo del proyecto se usó como referencia la Tesis “Detección y seguimientos de manos en videos digitales utilizando computador”, la misma planteaba el uso de cámaras de alta resolución y ordenadores de alta velocidad de procesamiento de imágenes, de hasta 10 cps; aspecto que no puede implementarse en un edificio multifamiliar, de allí la necesidad de desarrollar un algoritmo con ordenadores más sencillos y por consecuente para ordenadores con baja velocidad de procesamiento, lográndose en pruebas de campo la activación de alarma en tiempo real con una velocidad de hasta 02 cuadros por segundo.

5.1.3 Activación automática del sistema de alarma:

Otras aplicaciones como Dataglove, la varita mágica de LG Electronics, Mouse Electrónico, resuelven parcialmente el problema haciendo uso de un objeto. Una gran ventaja del proyecto desarrollado es que en situaciones extremas de inseguridad, el personal no ubica los objetos disuasivos en forma inmediata; en las pruebas de campo se confirma que los gestos corporales aprendidos en forma inconsciente son efectivos en estas situaciones logrando activar la alarma en tiempo real sin hacer uso de herramienta alguna, por lo que se concluye que el tiempo de respuesta en tiempo real no excede los 02 segundos, que es el tiempo máximo de procesamiento de cuadros.

5.1.4. Iluminación artificial:

En el objetivo principal se condicionó al uso de luz artificial, por su atributo de ser constante e invariable en el tiempo a diferencia de la luz natural, se hizo pruebas con uso de luz natural, sin conseguir los objetivos buscados, de allí que se confirme el uso necesario de luz artificial, y para obtener mejores resultados es necesario contar con una iluminación más intensa a espaldas del usuario.

5.1.5. Costo de implementación:

Los edificios multifamiliares que requieran contar con el proyecto investigado tendrá un costo de implementación muy inferior a otras propuestas como la tecnología inalámbrica Wizee y los sistemas IVS que están entre los US\$ 2,500 y US\$ 4,000 dólares americanos, pues la necesidad de implementar el sistema desarrollado requiere un gasto adicional al sistema con que se cuenta en el edificio, es decir se asume que el interesado ya cuenta con un sistema de seguridad electrónica, describiendo en la tabla 05 los equipos y servicios complementarios para implementar el proyecto de investigación realizado.

CANT	Descripción	Costo en US\$ (Inc. I.G.V)
01	Ordenador personal	US\$ 580
01	Interface externa	US\$ 100
01	Implementación del software desarrollado	US\$ 200
01	Servicios generales de conexión de alarma	US\$ 200
	Costo total de implementación	US\$ 1,080

Tabla 05: Costo de implementación en un edificio multifamiliar.

5.2 Reconocimiento de gestos

5.2.1 Prueba de tiempo de respuesta: el código fuente selecciona mediante un contador un cuadro por segundo, se muestra en pantalla un cuadro de la imagen sin procesar al inicio del programa y otro cuadro procesado al final del programa, la diferencia de tiempo de presentación entre ambos cuadros representa el tiempo de procesamiento, el cual se mantiene constante desde el inicio del video hasta concluir los 120 segundo que comprende el mismo, por lo que se determina que las respuestas se da en tiempo real.

Al seleccionar 02 cuadros por segundo en el contador existe un ligero retraso en la presentación de las imágenes procesadas, de manera imperceptible, mientras que para 03 cuadros, el retraso es considerable, pues el video cargado en el ordenador no avanza en tiempo real, por lo que se determina que para un uso continuo del sistema se recomienda un máximo de 02 cuadros por segundo.

5.2.2 Prueba de ausencia de personal: en el video procesado el usuario procede a retirarse por unos segundo de la recepción, cuando el usuario se

traslada a un costado, el sistema advierte que el personal se está retirando de su puesto, y al retirarse completamente activa la alarma automáticamente. En igual forma cuando una persona ajena a la recepción ingresa al counter estando el vigilante en su puesto, el algoritmo detecta un aumento del área procesada, enviando un mensaje “ingresando otra persona”.

5.2.3 Prueba de desactivación de alarma: Se realizan gestos levantando un solo brazo, ya sea el izquierdo o el derecho, como consecuencia el ordenador envía una señal de desactivación de alarma, dicha acción se realiza sentado, parado e inclinado.

5.2.4 Prueba de activación de alarma: Se realizan gestos levantando dos brazos, a la altura del rostro y en otras situaciones por encima del rostro, como consecuencia el ordenador envía una señal de activación de alarma, dicha acción se realiza sentado, parado e inclinado.

Se implementó hardware propuesto simulando una recepción para analizar resultados en forma constante, como se muestra en la interface externa de la figura 12, y tras pruebas realizadas, el sistema responde en tiempo real a los requerimientos, según gestos propuestos como son: ausencia de personal, levantar mano para apagar alarma y levantar 02 manos para activar alarma.

5.2.5 Interferencias en el hardware: El proceso de implementación se realiza con la plataforma de hardware libre Arduino, el cual posee un microcontrolador que realiza las diferentes operaciones de lectura y escritura en los pines de la plataforma. Estas declaraciones, tanto de ingreso como salida de datos en los pines se realizan en el código fuente o programa, debe tenerse especial cuidado cuando se declara un puerto como entrada de datos, donde el microprocesador realiza lecturas, y esto debido a que estas plataformas son muy sensibles a cualquier tipo de interferencia, por lo que se recomienda que el cable de alimentación que transporta el dato a leer debe estar debidamente protegido contra interferencias, colocando el cable apantallado a tierra.

5.3 Prueba de activación de alarma en un edificio multifamiliar

En la implementación en un edificio familiar, se simuló la central de alarma propia del edificio con un buzzer, para evitar asumir algún tipo de responsabilidad por concepto de deterioro de equipos en los días posteriores a la prueba. En las pruebas de campo para un edificio multifamiliar se siguieron los siguientes pasos a considerar para un buen desarrollo del procesamiento de imágenes:

1. Una iluminación más intensa en la parte posterior del usuario.
2. Se levantaron los dos brazos, quedando las manos al borde superior y debajo del renglón de emisión de fecha mostrado por el DVR dentro del cuadro de observación de la cámara.
3. Se reguló el lente variable de la cámara de tal manera que el centro de gravedad del usuario quede a $2/3$ de la parte superior del cuadro a procesar.
4. Se Inicializó el circuito microcontrolador, hasta observar que se comporte en forma estable sin activar la alarma en forma innecesaria.
5. Se simuló la activación con un buzzer por razones de garantía.

En la figura 34 se muestra la implementación del hardware utilizado en las pruebas de campo, donde los leds indican el estado de alarma en cada interface para comprobar los estados requeridos en la tabla 04. A la izquierda la interface 01 es quien recibe datos del ordenador desde el puerto USB, enviando una señal de entrada a la interface 02 que genera el tipo de señal requerido, este cable de conexión debe ser apantallado para evitar interferencias como se muestra en la figura 35, y en la parte derecha la interface 03 es el circuito controlador de relé y buzzer.

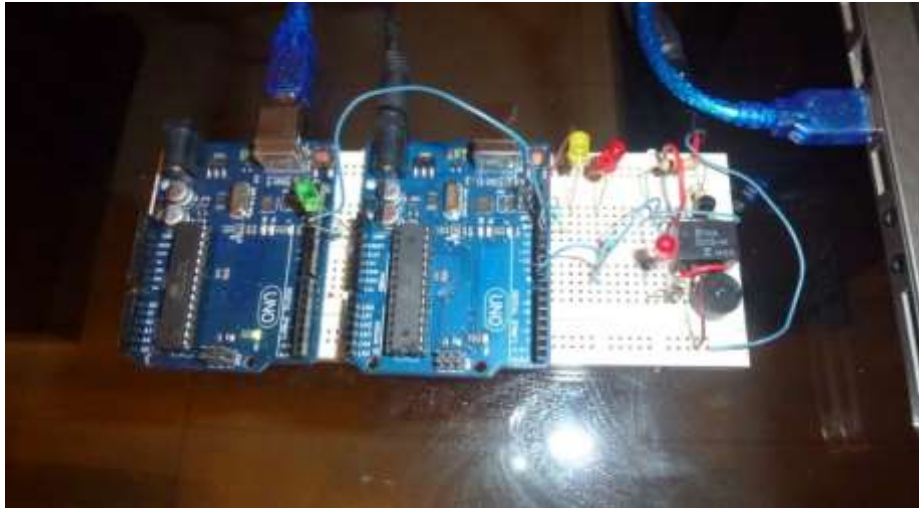


Figura 34: Interface implementada para pruebas de campo.



Figura 35: Interface con cable apantallado para pruebas de campo.

CONCLUSIONES

1. En el uso de técnicas matemáticas para el reconocimiento de gestos

El uso de diferentes técnicas matemáticas de procesamiento espacial de imágenes, capturadas en presencia de luz artificial, permitieron reconocer en tiempo real atributos de gestos corporales específicos planteados previamente, cuando se manejaba una velocidad de procesamiento de 01 cuadro por segundo. Usando una iluminación artificial más intensa en la parte posterior del objetivo de análisis, se obtuvo una silueta binarizada de contornos definidos, lo que reduce considerablemente las líneas del código fuente desarrollado en el entorno Matlab, el código desarrollado consiste de 64 líneas de los cuales 22 son para inicializar valores y 42 para un bucle para el constante procesamiento.

2. En la selección de la técnica matemática para tres gestos corporales

El plano de bits más significativo, es la técnica matemática de procesamiento espacial apropiada para el reconocimiento de tres gestos corporales, eligiendo el componente Rojo de un formato RGB, por ser el color que mejores atributos nos brinda. En el caso de las componentes G y B requiere técnicas adicionales de binarización donde se tiene que escoger un umbral de comparación, es decir un nivel de grises apropiado para obtener una imagen binarizada, este umbral varía según el color de las paredes del recinto y el color del uniforme del usuario. La necesidad de obtener una mejor definición y reconocimiento satisfactorio de la silueta binarizada, requirió complementar el procesamiento con técnicas de transformación morfológica. Para mejores resultados la cámara de video debe situarse de forma frontal al usuario entre 2 y 7 mts de distancia.

3. En la implementación del algoritmo en el entorno MatLab

La implementación algorítmica de las técnicas en el entorno MatLab es más

eficiente que otras aplicaciones, que son de uso general sin percibir detalles del procesamiento como el caso LabView, en el reconocimiento de gestos y en la generación de una señal eléctrica en el puerto USB del ordenador. Al poseer MatLab un lenguaje matricial propio y compatible con otras aplicaciones, facilitó la programación y generación del código fuente para el objetivo propuesto, y la configuración del puerto serial USB de uso recurrente.

4. En el diseño de una interface externa

La comunicación ordenada y sincronizada entre ordenador e interface permite la activación y desactivación del sistema de alarma, cuando se requirió su uso, para lo cual debe establecerse una velocidad de comunicación que en esta investigación fue de 9600 baudios. El eficiente diseño, se debió al uso de una plataforma económica Arduino R3, que contiene un microcontrolador, con un hardware y software libre compatible con MatLab. Al iniciarse el proceso el ordenador toma de 40 a 50 segundos realizar pasos de configuración y reconocimiento de la interface, manteniéndose estable en el transcurso del video. Parte del diseño se consideró la instalación de leds de colores diferentes en los terminales de la interface arduino, donde se puede apreciar el proceso de reconocimiento del código fuente del ordenador y el código fuente establecido en el microcontrolador. Los componentes utilizados para implementar el hardware suman un costo de US\$ 31 dólares americanos, pero se considera US\$100 (según tabla 05) para efectos de comercialización.

RECOMENDACIONES

1. El algoritmo consistió en hacer un seguimiento al usuario con desplazamientos verticales en su puesto, por consiguiente puede pararse, sentarse o inclinarse. Este algoritmo puede complementarse considerando seguimientos horizontales en caso se traslade hacia los costados.

2. Generar cuadro de horarios de ingreso y salida del personal del counter de recepción. Se realizaría una presentación virtual del último cuadro y el primer cuadro de video tomando como referencia el counter vacío. Ayudados por un atributo del DVR, el cual genera en su parte superior derecha la fecha y hora del video en cada cuadro.

3. Control de ingreso y salida del counter de personas. El sistema detecta cuando una persona se retira del counter y vuelve a ingresar, entonces puede imprimirse una relación de hora de ingresos y salidas adjuntando foto presente en el cuadro de video, con una presentación virtual.

4. El procesamiento digital de imágenes se desarrolló a partir de los niveles RGB, se recomienda usar otros tipos de niveles, que puedan favorecer a un desarrollo algorítmico más sencillo.

5. Cuando el personal se retira del counter, el sistema advierte con un enunciado lo sucedido, sin embargo el ordenador puede enviar una señal a otro pin de la interface arduino diferente del pin13, para activar un timer y darle un tiempo prudente para que el vigilante regrese a su puesto antes de que se active la alarma.

REFERENCIAS BIBLIOGRÁFICAS

- [1] A. Diaz, "Diseño de un sistema automatizado de Seguridad contra Intrusiones en edificios de departamento, utilizando el estándar de tecnología Inalámbrico Zigbee". Repositorio de Tesis PUCP, de Ing. Electrónica, Lima, Perú, Junio 2011.
- [2] F. Moreno, "Encuesta de ocurrencias en recepción de edificios multifamiliares", San Isidro, Lima, Perú, Junio 2015.
- [3] P. Cisneros, "Detección y seguimientos de manos en videos digitales utilizando computador". Repositorio de Tesis PUCP de Ing. Electrónica, Lima, Perú, Mayo 2014.
- [4] WiSee, "Tecnología inalámbrica gestos humanos", *University of Washington*. [Online]. Available: <http://wisee.cs.washington.edu>.
- [5] J. García "Sistemas de análisis de video", *Rev. Redseguridad.com*, Ed. España, Nov 2009.
- [6] F. Poyatos, *La comunicación no verbal: Cultura, lenguaje y conversación*. Madrid: Ed Istmo, 1994, ISBN 8470902806. vol. I, pp. 139.
- [7] J. Jiménez, *Lectura, expresión oral y escrita*. C.V. México: Ed. Nueva Imagen, S.A., 2008, 1º edición.
- [8] Definición ABC "gestos corporales" [Online]. Available http://www.definiciónabc.com/comunicación/gestos_corporales.php.
- [9] Desmond Morris, "el arte de la estrategia", *lenguaje corporal*, [Online]. Available http://www.elartedelaestrategia.com/lenguaje_corporal/.

- [10] P. Ekman, W. Friesen, *The Repertoire of Nonverbal Behaviour: Categories, Origins, Usage, and Coding*, 1969. *Semiotica* 1. pp. 49-98.
- [11] R. Ibañez, A. Teyseyre y M. Campo “ Evaluación de técnicas de Machine Learning para el reconocimiento de gestos corporales”, Argentina, 2014.
- [12] H. Rodriguez “La imagen digital y el código binario”, 2003, [Online]. Available: http://www.hugorodriguez.com/cursos/curso-idigital_01.htm.
- [13] Dr. B. Escalante, “Procesamiento Digital de Imágenes”, Agosto, 2006.
- [14] Prof. M. Martín Ortiz, “Procesamiento Digital de Imágenes”, Universidad Autónoma de Puebla. Facultad de Ciencias de la Computación, Enero, 2013.
- [15] “Escala de grises”, [Online]. Available <http://www.fotonostra.com/grafico/escalascolores.htm>.
- [16] Color5 “Escala acromática”. [Online]. Available: http://www.aloj.us.es/galba/digital/cuatrimestre_ii/imagen-pagina/2elementos4d.htm.
- [17] “Cubo RGB”, [Online]. Available <http://www.ipsi.fraunhofer.de/Kueppersfarbe/es/index.html>.
- [18] “Cubo RGB”, [Online]. Available <http://www.ipsi.fraunhofer.de/Kueppersfarbe/es/index.html>.
- [19] “Binarización de imágenes”, [Online]. Available <http://site-informatica.blogspot.pe/2010/09/binarizacion-de-imagenes.htm>.

- [20] “codificación en planos de bits”, *Definición*, [Online]. Available <http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/Imagenyvideo/compresion/3.1.3.htm>.
- [21] “codificación en planos de bits”, *bit mas significativos* [Online]. Available <http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/Imagenyvideo/compresion/3.1.3.htm>.
- [22] “Análisis morfológico en las señales gráficas”, Coruña, España, [Online]. Available <http://sabia.tic.udc.es/gc/Contenidos%20adicionales/trabajos/Imagenyvideo/morfologico/practica.htm>
- [23] Prof. M. Martín Ortiz, “Procesamiento Digital de Imágenes”, Universidad Autónoma de Puebla. Facultad de Ciencias de la Computación, Enero, 2013.
- [24] “Sensores con tecnología CCD vs CMOS”, *Sensor CCD*, Diciembre 2009. [Online]. Available <http://www.xatakafoto.com/camaras/sensores-con-tecnologia-ccd-vs-cmos>.
- [25] “Sensores con tecnología CCD vs CMOS”, *Sensor CCD*, Diciembre 2009. [Online]. Available <http://www.xatakafoto.com/camaras/sensores-con-tecnologia-ccd-vs-cmos>.
- [26] Definición.de “Definición de USB”, [Online]. Available <http://definicion.de/usb/> USB.
- [27] Símbolo de USB 2.0, [Online]. Available <http://windows.microsoft.com/es-es/windows-8/install-usb-3-usb-devices>.
- [28] Arduino Home. [Online]. Available <https://www.arduino.cc/>.

- [29] “Placa Arduino unoR3” [Online]. Available <http://mikroe.es/tag/manual-arduino-en-espanol/>.
- [30] Inventable.ed “como controlar un relé con transistor” *Circuito final completo*, [Online]. Available <http://www.inventable.eu/> .
- [31] Inventable.ed “como controlar un relé con transistor”, [Online]. Available <http://www.inventable.eu/>.
- [32] MATLAB - The Language of Technical Computing - MathWorks. [Online] Available www.mathworks.com/products/matlab/.
- [33] D. Barragán “Manual de Interfaz gráfica de usuarios en Matlab”, Ecuador, Agosto, 2007.
- [34] The MathWorks, Inc. “Toolbox Imagen Processing”, 2012. [Online]. Available <http://www.mathworks.com/products/?sec=categorydesc>.
- [35] “cámara CCD”, [Online]. Available <http://www.directindustry.es/prod/shihlin-electric-engineering-corporation/product-65405-452491.html>.
- [36] HS Sistemas, “Central de alarma”, 2009. [Online]. Available: <http://www.hssistemas.com/alarma.html>.
- [37] J. Pimentel, “*sistemas de video vigilancia con cámaras IP inalámbricas Pimentel*”, IE TEP P58, UPRP, Escuela Académica profesional de Ing. Electrónica, Lima, Perú, 2007.
- [38] “Sistema eficiente de reconocimiento de gestos de la mano”, Universidad politécnica de Madrid Escuela superior de Ingenieros de telecomunicaciones, Dpto. de señales, Sistema y radiocomunicación [Online]. Available: <http://silvela.org/jaime/pFC.pdf>.

- [39] L. Ferng, J. Bor “Reconocimiento robusto y en tiempo real de gestos”, Repositorios digitales Universidad de Chile, Chile, 2010 y Jong Bor 2010.
- [40] O. Marques, H. Viera “*Processamento Digital de Imagens*”, Rio de Janeiro, Brasil: Brasport, 1999. ISBN 8574520098.
- [41] M. Medina “Cómo escribir una bibliografía usando el formato IEEE” versión 1.1, [Online]. Available: mariomedina@udec.cl, Agosto,2014.
- [42] Hagroy Electronic, [Online]. Available www.hagroy.com/.
- [43] E. Pnufer “Técnicas de Procesamiento de imágenes”, Universidad del cauca, departamento de física, Colombia, 2012.

ANEXO 01

ENCUESTA: OCURRENCIAS QUE VULNERAN SEGURIDAD DEL EDIFICIO

Entrevistados: Adm. Lesli Barry Halfin.
Adm. Claudia Nuñez.
Adm. Carmen Contreras.
Adm. Lucía Caceres.
Adm. Lizzy Collado.
Adm. Javier Niño.

1. Tardanza en Hora de Ingreso.
2. Se toman tiempo extra para descanso o almuerzo.
3. No reportan todos los sucesos en cuaderno de ocurrencias.
4. Sus turnos son suplantados por otro personal.
5. Dejan el puesto para descansar en otro ambiente.
6. Dejan puerta principal de ingreso abierta.
7. Permiten ingreso de personas extrañas al edificio sin consultar.
8. No revisan el sistema circuito cerrado TV por ver televisión o películas.
9. Se retiran del edificio para realizar gestiones en bancos.
10. No están capacitados para ejercer puesto de seguridad o atención al público.
11. Atienden proveedores en otros ambientes sin dejar reemplazo en su puesto.
12. No cuentan con sistema de alarma o respuesta Inmediata en caso de emergencia.
13. Vulneran sistema de seguridad que controlan su estadía en el puesto de recepción.
14. Se presentan en estados no aptos para mantenerse despiertos.
15. Almuerzan en el puesto de recepción.
16. Suelen estar conversando con otro tipo de personal en recepción.

ANEXO 02

DATASHEET DE CÁMARA DE VIGILANCIA

Twida[®]

HONG KONG TENG WEI DA ELECTRONIC CO., LTD

Tel: +86-0755-83753640 Fax: +86-0755-83753684

Website: <http://www.sztwida.com>

TWD-ID202-700



VANDAL-PROOF VF IR DOME CAMERA

Specification

MODEL	TWD-ID202-700
Image Sensor	1/3-inch Sony Effio-E 960H Ex-view HAD II CCD
Horizontal Resolution	Color: 700TV Lines, B/W: 750TV Lines
Picture Pixels	PAL: 976(H)*582(V), NTSC: 976(H)*494(V)
LED No.	36pcs LEDs
Ranging Distance	35-40m
Infrared Wavelength	850nm
Sync System	Internal Synchronization
S/N Ratio	>52dB (AGC OFF)
Lens	4.9mm / 2.8-12mm 2.0 Megapixel IR varifocal Lens
Electronic Shutter	PAL: 1/50 -1/100,000 Sec; NTSC: 1/60 -1/100,000 Sec
Gamma Correction	$\gamma=0.45$
OSD menu	YES (8 Languages)
White Balance	Auto (ATW1, ATW2, AWC-SET, MANUAL)
BLC	BLC / HLC / ATR (WDR) OFF
AGC	High / Middle / Low (6dB-40dB)
Image Function	Digital WDR
Noise Reduction	2D-DNR
D/N Mode	Auto / Day / Night / Ext. Control for option
Minimum Illumination	Day: 0.01Lux/F1.2, Night: 0.001Lux/F1.2, 0Lux (IR ON)
Power Supply	DC12V \pm 20% (Dual power DC12V / AC24V is optional)
Case	IP66-rated black or white metal case with anti-reflection glass Focus/zoom are adjustable outside by manual
Size	Φ 140mm*100mm

ANEXO 03
PROGRAMA COMPUTACIONAL DESARROLLADO EN MATLAB Y
ARDUINO

3.1 Programa 'AlarmaGestosArduino':

```
<< clear
% configurando interface Arduino %

<< install_arduino
<< a=arduino('COM11');
<< a.pinMode(13,'output')

% declaración de variables %

<< video1 = vision.VideoFileReader('recepcion5.avi');
<< i = 1;
<< k = 1;
<< j = 1;
<< EE=strel('arbitrary',ones(6)); % elemento estructural %

% variables para calcular altura del rostro%

<< S = 0;           % suma de bits de un perfil de línea           %
<< d = 0;           % distancia entre rostro y arista superior     %
<< f = 1;           % número de fila que ubica parte superior del rostro%
                    % declarando variables para reconocer gestos que activan
                    alarma%

<< PD=0; % perfil derecho %
<< PI=0; % perfil izquierdo %
<< PC=0; % perfil central %
<< titulo='en espera';
```

```
%iniciando lectura de video captando cuadro por cuadro y recortándolo%
```

```
<< while ~isDone(video1);  
<< cuadro = step(video1);  
<< r_cuadro= cuadro(1:140,45:258,1);
```

```
% captura de 01 cuadro de cada 30 cuadros del video original %
```

```
<< if mod(i,30)==0 && j<120;
```

```
<< while S==0 && f<120 ;
```

```
% procesamiento espacial y transformación morfológica del cuadro %
```

```
<< A1=bitand(im2uint8(r_cuadro),128)*2; %extracción de plano de bits%  
<< A1dilatado=imclose(A1,EE); %transformación morfológica%  
<< A2=im2double(255-A1dilatado); % creando imagen inversa %
```

```
%ubicando la altura 'd' del rostro%
```

```
<< for c=73:142 ; S=S+A2(f,c) ; end  
<< d = f;  
<< f = f+1;  
<< end
```

```
%cálculo de variables del perfil de línea a la altura del rostro%
```

```
<< for k=1:72 ; PD=PD+A2(d+15,k);end  
<< for k=73:142 ; PC=PC+A2(d+15,k);end  
<< for k=143:214; PI=PI+A2(d+15,k);end
```

```
% analizar resultados que determinan activación de alarma%
```

```

<< if PC<40 && PC>25 ;titulo='conservar estado de alarma';end
<< if PD>5 && PD<25 && PI==0 && PC<40 && PC>25 ;
  << titulo='parar alarma';apagaralarma;
<< end
<< if PI>5 && PI<25 && PD==0 && PC<40 && PC>25 ;
  << titulo='parar alarma';apagaralarma;
<< end
<< if PI>5 && PI<25 && PD<25 && PD>5 && PC>25 && PC<40 ;
  << titulo='alarma activese';activaralarma;
<< end
<< if PD>25 ;titulo='ingreso otra persona';activaralarma;end
<< if PI>25 ;titulo='ingreso otra persona';activaralarma;end
<< if PD==0 && PI>25 && PC==0 ;
<< titulo='personal se retira';activaralarma;
<< end
<< if PD>25 && PI==0 && PC==0 ;
  << titulo='personal se retira';activaralarma;end
<< if PD==0 && PI==0 && PC==0 ;
  << titulo='alarma counter vacio';activaralarma;
<< end

% mostrando imagen procesada indicando estado de solicitud de alarma%
<< imshow(A2);title(titulo);
<< j=j+1;
<< end
<< i=i+1;
<< end
<< release(video1);
3.2 Programa 'apagaralarma ': << a.pinMode(13,LOW);
3.3 Programa 'activaralarma ': << a.pinMode(13,HIGH);

```

ANEXO 04
PROGRAMA ARDUINO PARA GENERAR 02 TIPOS DE SEÑALES
ELÉCTRICAS

```
Void setup()                % bloque de declaración de puertos %
{
  Serial.begin (9600);      % definir velocidad de comunicación en baudios%
  pinMode(8,OUTPUT);       % se declara pin 08 como salida hacia alarma%
  pinMode(12,OUTPUT);      % se declara pin 12 como pin de salida %
  pinMode(13,INPUT);       % se declara pin 13 como pin de entrada %
}
Void loop()
{
    %estando alarma apagada se pide que encienda %
  If ((digitalRead(12)==LOW && (digitalRead(13)==HIGH))
  {
    digitalWrite(12,HIGH );
    digitalWrite(8,HIGH ); delay (500);
    digitalWrite(8,LOW); delay (2000);
    digitalWrite(8,HIGH ); delay (500);
    digitalWrite(8,LOW ); delay (2000);
    digitalWrite(8,HIGH ); delay (500);
    digitalWrite(8,LOW); delay (2000);
    digitalWrite(8,HIGH );}
  % estando alarma prendida se pide que se apague %
  If ((digitalRead(12)==HIGH && (digitalRead(13)==LOW))
  {digitalWrite(12,LOW );
    digitalWrite(8,LOW ); delay (600);
    digitalWrite(8,HIGH ); delay (400);
    digitalWrite(8,LOW ); delay (600);
    digitalWrite(8,HIGH ); delay (400);
    digitalWrite(8,LOW );
  } }
```


ANEXO 05
COSTO DEL PROYECTO DE TESIS

La implementación del proyecto requirió de la adquisición de diversos componentes para la captura, procesamiento, activación y simulación de una alarma. Adicionalmente se contempla los requerimientos de tiempo para realizar la presente investigación y algunos materiales tipo administrativo para la elaboración de la tesis. Los detalles mencionados se registran en la tabla adjunta que conlleva a la implementación física para realizar las pruebas necesarias de esta investigación:

CANT	DESCRIPCIÓN	COSTO EN SOLES (INC I.G.V)
01	Cámara de seguridad Twida Nova	S/. 187
01	DVR 04 Canales Nova	S/. 463
01	Accesorios (Conectores , cables)	S/. 36
01	Fuentes de alimentación 12 v DC 220v AC	S/. 45
01	USB portátil	S/. 28
02	02 Plataforma Arduino R3	S/. 72
01	Interface transistor _ relé_ buzzer	S/. 27
01	Tiempo del tesista: 120 días,04 horas diarias, S/.20 por hora	S/. 9,600
01	Impresión de avances y tesis	S/. 475
	Costo total del desarrollo del proyecto de tesis	S/. 10,933

ANEXO 06

ENCUESTA: IMPLEMENTACIÓN TECNOLÓGICA EN EDIFICIOS DE SAN ISIDRO

Esta encuesta fue realizada en Junio del 2015 a una muestra de 32 edificios situados en un cuadrante del distrito de San Isidro, entre las calles Jacinto Lara, Alvares Calderón, Victor Maurtua y Angamos Oeste; con una población total de 168 edificios, con el objetivo de determinar el nivel tecnológico actual con que se cuenta en los edificios multifamiliares del presente distrito.

Las abreviaturas mencionadas en los diferentes ítems tienen los siguientes significados:

Cerco	Cerco eléctrico o inalámbrico que rodean al edificio.
A.I.	Alarma contra incendio.
CCTV	Circuito cerrado de televisión.
GRAB:	Sistema de grabación del sistema de cctv.
PC:	Computadora personal.
INT SERV	Servicio de internet.
TELF	Teléfono.
SIST RPTA	Sistema de respuesta inmediata ante un evento.
G.E	Grupo electrógeno.
ASC ACC	Acceso directo al ascensor.
ADM:	Servicio de administración de áreas comunes.
REV CUA	Revisión cotidiana de cuaderno de ocurrencias.
RV VD	Revisión cotidiana de grabación de videos en un DVR.
CAM REC	Instalación de cámara de vigilancia en recepción.
TUR	Número de turnos de servicio de vigilancia.
CAP	Capacitación presente en personal de vigilancia.

dirección	Cerco	A.I.	CCTV	GRAB	PC	INT SERV	TELF	SIST RPTA	G.E	ASC ACC	ADM	REV CUA	RV VD	CAM REC	TUR	CAP
Jacinto Lara 450	NO	NO	NO	NO	NO	NO	CELU	NO	NO	SI	NO	SI	NO	NO	02	SI
Jacinto Lara 465	NO	NO	NO	NO	NO	NO	FIJO	NO	NO	SI	SI	SI	NO	NO	02	NO
Jacinto Lara 440	NO	SI	NO	NO	NO	NO	NO	NO	NO	SI	NO	NO	NO	NO	NO	NO
Jacinto Lara 341	SI	SI	SI-04	SI	NO	NO	FIJO	CLAV 03	NO	SI	SI	SI	SI	SI	02	SI
Jacinto Lara 212	SI	SI	NO	NO	NO	NO	FIJO	PITO	NO	NO	SI	SI	NO	NO	02	SI
Jacinto Lara 206	SI	SI	SI-08	SI	NO	NO	FIJO	NO	NO	NO	SI	SI	NO	NO	02	NO
Jacinto Lara 193	SI	NO	SI-08	SI	NO	NO	FIJO	PULS	NO	NO	SI	SI	SI	SI	02	SI
Jacinto Lara 120	SI	SI	SI-08	SI	NO	NO	FIJO	NO	NO	NO	SI	SI	NO	NO	02	NO
Alvares Cald 670	NO	SI	SI-16	SI	NO	NO	FIJO	NO	SI	SI	NO	SI	NO	SI	02	NO
Alvares Cald 680	SI	SI	SI-16	SI	NO	NO	FIJO	NO	SI	SI	NO	SI	NO	SI	02	NO
Alvares Cald 498	SI	SI	SI-04	SI	NO	NO	FIJO	NO	NO	NO	SI	SI	NO	NO	02	SI
Alvares Cald 458	SI	SI	SI-08	SI	NO	NO	FIJO	NO	SI	NO	SI	SI	SI	NO	02	NO
Alvares Cald 365	SI	SI	SI-01	NO	NO	NO	FIJO	NO	NO	NO	SI	SI	NO	NO	02	NO

dirección	Cerco	A.I.	CCTV	GRAB	PC	INT SERV	TELF	SIST RPTA	G.E	ASC ACC	ADM	REV CUA	RV VD	CAM REC	TUR	CAP
Alvares Cald 308	NO	NO	NO	NO	NO	NO	NO	NO	SI	NO	SI	SI	NO	NO	03	SI
Alvares Cald 337	SI	SI	SI-04	NO	NO	NO	FIJO	PULS	NO	NO	SI	SI	NO	NO	02	NO
Victor Maurtua 225	NO	NO	NO	NO	NO	NO	FIJO	NO	SI	SI	SI	SI	NO	NO	02	NO
Victor Maurtua 300	NO	NO	SI-04	SI	NO	NO	FIJO	NO	NO	SI	SI	SI	NO	NO	03	NO
Victor Maurtua 324	SI	SI	SI-04	SI	NO	NO	FIJO	PULS	SI	SI	SI	SI	NO	NO	02	NO
Victor Maurtua 385	SI	SI	SI-08	SI	NO	NO	FIJO	ALAR	NO	SI	SI	SI	NO	NO	02	SI
Victor Maurtua 447	SI	SI	SI-04	SI	NO	NO	FIJO	NO	NO	NO	SI	NO	NO	NO	02	NO
Victor Maurtua 456	SI	SI	SI-08	NO	NO	NO	FIJO	PITO	NO	NO	SI	SI	NO	NO	02	SI
Sta Margart 273	NO	NO	SI-02	NO	NO	NO	FIJO	NO	NO	SI	SI	NO	NO	NO	02	SI
Blas Cerdeña 210	NO	NO	NO	NO	NO	NO	NO	NO	NO	SI	SI	SI	NO	NO	02	NO
Blas Cerdeña 197	SI	NO	SI-04	NO	NO	NO	FIJO	PITO	NO	SI	SI	NO	NO	NO	02	NO
Blas Cerdeña 485	SI	SI	SI-04	SI	NO	NO	FIJO	NO	NO	NO	SI	SI	NO	NO	02	NO
Blas Cerdeña 395	NO	NO	SI-08	SI	NO	NO	FIJO	NO	SI	NO	SI	SI	NO	NO	02	SI

dirección	Cerco	A.I.	CCTV	GRAB	PC	INT SERV	TELF	SIST RPTA	G.E	ASC ACC	ADM	REV CUA	RV VD	CAM REC	TUR	CAP
Blas Cerdeña 310	SI	NO	SI-08	SI	NO	NO	FIJO	NO	SI	SI	SI	NO	NO	SI	02	NO
Alf Salazar 673	SI	SI	SI-06	SI	NO	NO	CELU	PULS	NO	NO	SI	SI	NO	NO	02	NO
Alf Salazar 645	NO	SI	SI-10	SI	NO	NO	CELU	NO	NO	NO	SI	SI	SI	NO	02	NO
Carlos Graña 250	SI	NO	SI-06	SI	NO	NO	NO	PULS	NO	SI	SI	SI	NO	NO	02	SI
Carlos Graña 340	SI	SI	SI-04	NO	NO	NO	FIJO	NO	NO	NO	SI	SI	NO	NO	02	NO
Angamos 1624	SI	SI	SI-04	NO	NO	NO	FIJO	NO	NO	NO	SI	SI	NO	NO	02	NO