



FACULTAD DE INGENIERÍA Y ARQUITECTURA
ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

**DISEÑO DE UN SISTEMA DE DETECCIÓN DE INCIDENCIAS EN
LOS PARADEROS DEL METROPOLITANO, UTILIZANDO
TÉCNICAS DE PROCESAMIENTO DIGITAL DE IMÁGENES**

PRESENTADA POR

OSCAR ROSALES OSPINAL

ERIK VERGARA LIVIA

TESIS

PARA OPTAR EL TÍTULO PROFESIONAL DE INGENIERO ELECTRÓNICO

LIMA – PERÚ

2014



**Reconocimiento - No comercial - Sin obra derivada
CC BY-NC-ND**

El autor sólo permite que se pueda descargar esta obra y compartirla con otras personas, siempre que se reconozca su autoría, pero no se puede cambiar de ninguna manera ni se puede utilizar comercialmente.

<http://creativecommons.org/licenses/by-nc-nd/4.0/>



USMP

UNIVERSIDAD DE
SAN MARTÍN DE PORRES

**FACULTAD DE
INGENIERÍA Y ARQUITECTURA**

ESCUELA PROFESIONAL DE INGENIERÍA ELECTRÓNICA

**DISEÑO DE UN SISTEMA DE DETECCIÓN DE INCIDENCIAS
EN LOS PARADEROS DEL METROPOLITANO, UTILIZANDO
TÉCNICAS DE PROCESAMIENTO DIGITAL DE IMÁGENES**

TESIS

PARA OPTAR EL TÍTULO DE INGENIERO ELECTRÓNICO

PRESENTADO POR

ROSALES OSPINAL, OSCAR

VERGARA LIVIA, ERIK

LIMA – PERÚ

2014



Dedicatoria

A Dios y a la Virgen María, porque ellos nos dieron las fuerzas para poder culminar la carrera. A nuestras familias ya que con su gran apoyo hemos podido alcanzar una de nuestras metas profesionales.



Agradecimiento

A Dios por estar en nuestras vidas, a nuestros padres por apoyarnos y poder culminar esta carrera con esfuerzo, a nuestras esposas por comprender el gran sacrificio, y a nuestras hijas que son el motor de todo lo que hacemos.

INDICE

	Página
RESUMEN	x
ABSTRACT	xi
INTRODUCCIÓN	xii
CAPÍTULO I: MARCO TEÓRICO	16
1.1 Antecedentes	16
1.2 Estado del arte	18
1.3 Bases teóricas	29
1.4 Propuesta de la Solución	41
CAPÍTULO II: METODOLOGÍA	42
2.1 Material	42
2.2 Métodos	46
CAPÍTULO III: DESARROLLO DEL PROYECTO	60
3.1 Justificación	60
3.2 Diagrama de Flujo	63
CAPÍTULO IV: PRUEBAS Y RESULTADOS	67
4.1 Diseño de para la extracción del área de interés	68
4.2 Diseño para la detección del faro de color rojo del bus	69
4.3 Diseño para la extracción del área de interés	74
4.4 Diseño para la detección de personas	75
CAPÍTULO V: DISCUSIÓN Y APLICACIÓN	100
CONCLUSIONES	101
RECOMENDACIONES	102
GLOSARIO	103
FUENTES DE INFORMACIÓN	104
ANEXOS	105

Lista de tablas

		Página
Tabla 1	Incidentes mensuales	18
Tabla 2	Características de PC	43
Tabla 3	Características de la cámara	44
Tabla 4	Distribución de la muestra	49
Tabla 5	Tipología textual	50
Tabla 6	Puntuación de ítems	52
Tabla 7	Listado de costos	97
Tabla 8	Listado de tareas	98



Lista de figuras

	Página	
Figura 1	Cuadro de accidentes	17
Figura 2	Imágenes con detección	22
Figura 3	Adyacente por frontera y por esquina	31
Figura 4	Vecindad de 4 y de 8	31
Figura 5	Generación de colores en el espacio RGB 8	32
Figura 6	Histograma de una imagen	33
Figura 7	Menor, media y mayor resolución	35
Figura 8	Imagen normal e imagen en escala de grises	36
Figura 9	Igual media y diferente dispersión, viceversa	38
Figura 10	Cámara IP, router, pc e internet	40
Figura 11	Laptop	44
Figura 12	Cámara IP	45
Figura 13	MatLab	46
Figura 14	Sistema operativo Windows 7	46
Figura 15	Office 7	46
Figura 16	Comienzo de tamaño de la Imagen (1,1)	48
Figura 17	Fin de tamaño de la imagen(640,480)	48
Figura 18	Primera área de interés de la imagen total	48
Figura19	Imagen resultado	50
Figura 20	Imagen con segundo resultado	50
Figura 21	Segunda área de interés	51
Figura 22	Imagen sin persona	53
Figura 23	Imagen con presencia de persona	53
Figura 24	Diagrama de bloques	55
Figura 25	Diagrama de bloques	56
Figura 26	Diagrama de bloques	57
Figura 27	Diagrama de bloques	58
Figura 28	La varianza	59
Figura 29	Diagrama de Flujo	63

Figura 30	Diagrama de Bloques	64
Figura 31	Diagrama pictográfico de la propuesta	65
Figura 32	Imagen a color y con las 3 matrices	66
Figura 33	Llegada del bus y extracciones del área de interés	67
Figura 34	Imagen a color a procesar	68
Figura 35	Imagen a color en sus 3 matrices	69
Figura 36	Imagen discriminada	69
Figura 37	Imagen discriminada de 0 a 255	70
Figura 38	Imagen solo del color rojo	70
Figura 39	Imagen solo de 0 a 255	71
Figura 40	Imagen binarizada del color rojo	72
Figura 41	Extracción del área de interés	73
Figura 42	Imagen a escala de grises	74
Figura 43	Embarque vacío	75
Figura 44	Histograma	75
Figura 45	A escala de grises	76
Figura 46	Con persona en embarque	77
Figura 47	Histograma de área de interés	77
Figura 48	Escala de grises	78
Figura 49	Números de tonalidades en escala de gris	79
Figura 50	Área de interés vacío	80
Figura 51	Área de interés con persona	80
Figura 52	Imagen tomada	81
Figura 53	Imagen procesada	82
Figura 54	Imagen tomada	82
Figura 55	Imagen procesada	83
Figura 56	Imagen tomada	83
Figura 57	Imagen procesada	84
Figura 58	Imagen tomada	84
Figura 59	Imagen procesada	85
Figura 60	Imagen tomada	85
Figura 61	Imagen procesada	86
Figura 62	Imagen tomada	86

Figura 63	Imagen procesada	87
Figura 64	Imagen tomada	87
Figura 65	Imagen procesada	88
Figura 66	Imagen tomada	88
Figura 67	Imagen procesada	89
Figura 68	Imagen tomada	89
Figura 69	Imagen procesada	90
Figura 70	Imagen tomada	90
Figura 71	Imagen procesada	91
Figura 72	Imagen tomada	91
Figura 73	Imagen procesada	92
Figura 74	Imagen tomada	92
Figura 75	Imagen procesada	93
Figura 76	Imagen tomada	93
Figura 77	Imagen procesada	94
Figura 78	Imagen tomada	94
Figura 79	Imagen procesada	95
Figura 80	Imagen tomada	95
Figura 81	Imagen procesada	96
Figura 82	Imagen tomada	96
Figura 83	Imagen procesada	97

Lista de anexos

	Página
Anexo 1 Programa para la toma de imágenes utilizando cámara IP y matlab	105
Anexo 2 Programa para elegir la primera área de interés	106
Anexo 3 Primer procesamiento de área de interés	107
Anexo 4 Segundo procesamiento de área de interés	108
Anexo 5 Detección del bus	109
Anexo 6 Comandos para hallar la varianza del área de interés	110
Anexo 7 Para detección del Bus	111
Anexo 8 Para detección de bus	113
Anexo 9 Para detección de persona en embarque	116
Anexo 10 Red de fibra óptica Metropolitano	118
Anexo 11 Estaciones del Metropolitano	118
Anexo 12 Instalación de cámaras vs embarque	120
Anexo 13 Características de la cámara IP seleccionada para la implementación	121
Anexo 14 Segunda zona de interés	122

RESUMEN

La presente investigación, tiene como objetivo dar a conocer un diseño tecnológico que pueda solucionar el problema de control de los incidentes ocurridos en los embarques de cada paradero de las estaciones del Metropolitano. Se sabe que los accidentes al momento de abordar un medio de transporte terrestre son muy comunes, tanto a nivel local como a nivel mundial.

Para poder llegar al objetivo se ha utilizado como metodología el procesamiento digital de señales, específicamente de imágenes. Que consiste en la captura de video mediante una cámara IP, para luego dividirla en imágenes o fotos. Cada imagen será procesada mediante el uso del programa Matlab, el cual nos ayudara a realizar un procesamiento de cada foto y video de acuerdo al programa diseñado.

Luego de realizar la toma de imágenes se tuvo como resultado en un primer momento de la detección del bus cuando éste llegaba al paradero, y después se detecto en el mismo embarque la presencia de las personas cuando bajaban del bus.

Por lo tanto se concluye que el diseño tecnológico debería ser aplicado en nuestro país, como primer paso en Lima metropolitana.

Palabras claves: Diseño, sistema de detección, incidencias, paraderos, embarques.

ABSTRACT

The present research aims to present a technological design that can solve the problem of control of incidents occurring in the shipments of each location of the Metropolitan stations. Accidents when dealing with land transport are known to be very common, both locally and globally.

In order to reach the target, digital signal processing, specifically of images, has been used as methodology. It consists of capturing video using an IP camera, then dividing it into images or photos. Each image will be processed using the Matlab program, which will help us to perform each photo and video processing according to the designed program.

After taking the images, the result was the detection of the bus when it reached the destination, and then the presence of people when they got off the bus was detected.

Therefore it is concluded that technological design should be applied in our country, as a first step in metropolitan Lima.

Key Words: Design, detection system, incidences, stops, shipments.

INTRODUCCIÓN

En la actualidad el tema del tráfico en las ciudades del país es bastante preocupante, en Lima se implementó el sistema del Metropolitano que ayuda a gran parte de la población a mejorar su transporte en determinadas zonas de la ciudad, pero como todo sistema, tiene su deficiencia, uno de estos es la detección de incidentes en los paraderos.

Durante el gobierno de Susana Villarán, se subieron los pasajes del Metropolitano de S/. 2.00 a S/. 2.50, aun así no se controlaban ni monitoreaban los incidentes que ocurría en los embarques de los paraderos.

Adicionalmente se puso en marcha la interconexión entre el Metropolitano y el metro de Lima, lo que hizo que el riesgo de incidentes se incremente. Siendo el propósito de la tesis, no solo prevenir accidentes, sino monitorear lo que ocurre en los embarques.

La presente tesis está estructurada en 5 capítulos, en el primero se da a conocer el Marco Teórico en el cual se presentan los antecedentes que motivaron el desarrollo de esta tesis, a su vez se dará un vistazo de cómo se tomo el problema en otras partes para poder dar a conocer el estado del arte, efectuada la investigación se indicará cuál es la propuesta de solución y se fijarán los objetivos generales y específicos.

En el segundo capítulo, mostraremos la metodología seguida para lograr a modelar las áreas de embarque y así mostrar cómo se resuelven los problemas, utilizando la tecnología de nuestro lado, así como la detección de los buses.

Siguiendo en el tercer Capítulo, para resolver el problema se desarrollará en fases, así como las técnicas que nos ayudaran al desarrollo de la tesis, así como las técnicas y poder lograr los objetivos propuestos.

En el cuarto Capítulo, como toda solución a los problemas, necesitaremos realizar pruebas y simulaciones que nos confirmen la solución que hemos optado. Las simulaciones se realizaron para esta primera etapa,

considerando la luminosidad diurna como prueba principal, así como el correcto estado de los buses.

Para culminar en el quinto Capítulo, detallaremos las discusiones y aplicaciones para poder brindar otras soluciones para otros problemas dentro del mismo campo.



1. Planteamiento del problema

Durante las investigaciones, se planteo como principal problema, los incidentes ocurridos en los paraderos del Metropolitano y su falta de atención en el momento de los mismos.

2. Objetivos

2.1. Objetivo general

Diseñar un sistema de detección y monitoreo de incidentes ocurridos en los paraderos de la red de transporte del Metropolitano, utilizando Software de Simulación. Para su posterior toma de acción de Centro de Control.

2.2. Objetivos específicos

- a) Diversos Objetivos especificos han sido planteados, en función de lograr nuestro objetivo principal estos son:
- b) Establecer estadísticas de incidencias en toda la red del metropolitano.
- c) Elegir las estaciones del metropolitano para la realización de pruebas.

Diseñar la programación del sistema en Matlab de detección.

3. Limitación

Como limitación que se obtuvo dentro del desarrollo de la tesis, fue en el aspecto tecnológico, ya que como principal herramienta de apoyo solo se conto con una cámara No profesional, y con esta se implementó los programas en el software elegido.

4. Justificación

El 23% de los incidentes que ocurren en el Metropolitano implica a pasajeros presionados en las puertas de los buses, dato revelado por el gerente de Operaciones de Protransporte, Hernán Navarro.

Él explicó que estos problemas se deben porque los usuarios no abordan o descienden de las unidades de una manera adecuada. A raíz de esto, La empresa Protransporte inició la campaña "Esté Atento", que buscó informar y concientizar a los ciudadanos para que respeten el aviso de cierre de puertas de los buses del sistema y eviten problemas. De tal forma que se concientice sobre la importancia de estar atentos a las luces rojas y al sonido que emite el bus a la hora de abrir y cerrar las puertas. Estos incidentes son informados al Centro de Control, de manera ineficiente, debido a que no existe personal monitoreando cada plataforma de embarque.

Es por ello que se vio viable la utilización de un sistema de monitoreo en tiempo real, que pueda alertar inmediatamente sobre algún incidente en las plataformas de embarque.

5. Alcances

EL alcance de la presente tesis es la detección del bus, y luego detectar si existe alguna persona, para luego enviar una alerta. Todo esto en condiciones normales a la luz del día. Se utilizo un video grabado con la cámara IP obtenida para las pruebas.

CAPÍTULO I

MARCO TEÓRICO

1.1 Antecedentes

En el Perú, cada año, las lesiones causadas por accidentes de tránsito son motivo de un alto número de fallecimientos y de atenciones en los servicios de emergencias hospitalarios.

El conocimiento de la magnitud y tendencias del problema es el punto de partida para el estudio de la situación de las lesiones causadas por accidentes de tránsito, de este modo se podrá conocer la carga y respuesta del país frente a las lesiones ocasionadas por estos.

De este modo, si se tiene información confiable y oportuna es posible desarrollar intervenciones orientadas a los factores de riesgo causantes de los accidentes de tránsito. Es importante romper el paradigma de que son producto del azar y actuar en consecuencia.

El Metropolitano es el nuevo sistema integrado de transporte público para Lima que cuenta con buses articulados de gran capacidad que circulan por corredores exclusivos bajo el esquema de autobuses de tránsito rápido.

El primer corredor del Metropolitano conecta Lima Sur con Lima Norte, recorriendo 16 distritos de la ciudad desde Chorrillos hasta Comas.

El objetivo de este moderno sistema es elevar la calidad de vida de los ciudadanos, al ahorrarles tiempo en el traslado diario, proteger el medio ambiente, brindarles mayor seguridad, una mejor calidad de servicio y trato más humano, especialmente a gestantes, mujeres con niños en brazo, niños, adultos mayores y personas con discapacidad.

El proyecto del Metropolitano fue ejecutado con fondos propios de la Municipalidad de Lima y el financiamiento del Banco Interamericano de Desarrollo con el Banco Mundial, constituye un eje transformador de la

ciudad ya que su diseño contempla además el mejoramiento del mobiliario urbano, construcción de nuevas pistas de transporte privado, cambio de redes de servicios de luz, agua y telefonía además del tratamiento paisajístico de toda la zona de intervención del sistema.

A diferencia de los sistemas que funcionan en ciudades como Bogotá, Curitiba o México, nuestro sistema es el primero que opera a gas natural vehicular, lo que permitirá contribuir a la reducción de la contaminación que genera el parque automotor.

El 23% de los incidentes que ocurren en el Metropolitano implica a pasajeros presionados en las puertas de los buses, así lo revela el gerente de Operaciones de Protransporte, Hernán Navarro.

Él explicó que estos problemas se deben porque los usuarios no abordan o descienden de las unidades de una manera adecuada. Razón por la cual se busca una solución eficiente que pueda dar pronta respuesta a estos incidentes y así tratar de prevenir los mismos, para poder minimizar el número de usuarios perjudicados.

A continuación se muestran las estadísticas de accidentes ocurridos en el sistema metropolitano:

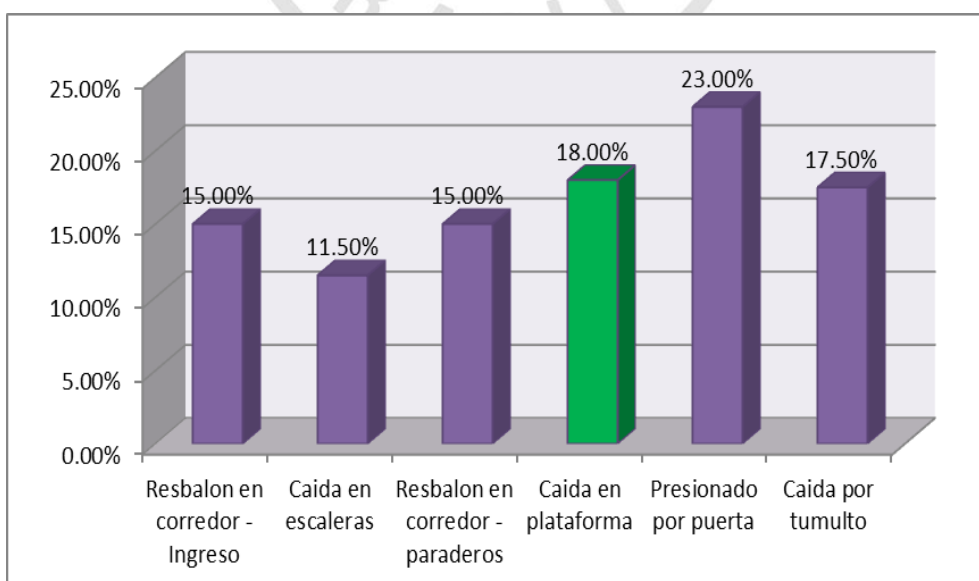


Figura 1: Cuadro de accidentes
Fuente: Base de datos del metropolitano

Adicionalmente mencionamos las estadísticas de caída en plataforma del último año:

Tabla N° 01: Incidentes mensuales.

Mes	Incidentes Mensuales											
	Enero	Febrero	Marzo	Abril	Mayo	Junio	Julio	Agosto	Setiembre	Octubre	Noviembre	Diciembre
Caida en plataforma	2	0	0	2	1	0	3	0	1	3	0	2

Fuente: Base de datos del Metropolitano.

Problema ingenieril

Básicamente se hizo dos preguntas para llegar a los objetivos

¿Es posible realizar la detección del bus cuando este llegue a los paraderos utilizando tecnología actual? ¿Cómo?

¿Es posible realizar la detección de las personas, cuando el bus haya partido con visión artificial? ¿Cómo?

1.2 Estado del arte

Para Molina (2005):

El estado del arte es una modalidad de la investigación documental que permite el estudio del conocimiento acumulado (escrito en textos) dentro de un área específica. Sus orígenes se remontan a los años ochenta, época en la que se utilizaba como herramienta para compilar y sistematizar información especialmente el área de ciencias sociales, sin embargo, en la medida en que estos estudios se realizaron con el fin de hacer balances sobre las tendencias de investigación y como punto de partida para la toma de decisiones, el estado del arte se posicionó como una modalidad de investigación de la investigación. Es por eso que el estado del arte ayuda a nuestro proyecto o tesis a poder darle un camino hacia el objetivo final, guiándonos de otras informaciones y poder cumplir con el alcance.

Se ha investigado acerca del metro en otros países, y no se ha encontrado una solución similar a la planteada en la presente tesis. Si bien es cierto Perú vive uno de los mayores caos, en lo que le corresponde al transporte, parte de esta situación se debe a una cultura mal formada, que con el tiempo tendrá que ir alineándose. Es por ello que hemos reflejado nuestro Estado del Arte a las técnicas utilizadas para la detección de objetos mediante el procesamiento de señales. (p. 73)

1.3.1 Visión global de los sistemas de video vigilancia

Para Vignolo (2008):

Las señales se representan matemáticamente como funciones de una o más variables independientes. La variable independiente más común es el tiempo o, y algunas señales que dependen de él son, por ejemplo, la voz, una onda de radio, un electrocardiograma, etc. Otras señales, tales como las imágenes, son funciones de 2 variables independientes, ya que contienen información de brillo o de colorido en función de las coordenadas X e Y de un plano. Las señales se representan matemáticamente como funciones de una o más variables independientes. La variable independiente más común es el tiempo, y algunas señales que dependen de él son, por ejemplo, la voz, una onda de radio, un electrocardiograma, etc. Otras señales, tales como las imágenes, son funciones de 2 variables independientes, ya que contienen información de brillo o de colorido en función de las coordenadas X e Y de un plano. Las señales que trataremos en los procesos son del tipo imágenes que serán procesadas por el simulador Matlab. (p. 19)

Se entiende por sistema de video vigilancia a toda aquella actividad que implique la colocación de una o varias cámaras de grabación, fijas o móviles, que tengan la finalidad de monitorizar el comportamiento y actividades de un espacio o personas. Los sistemas de video vigilancia pueden estar compuestos, simplemente, por una o más cámaras de vigilancia conectadas

a uno o más monitores o televisores que reproducen las imágenes capturadas por las cámaras. Los sistemas de video vigilancia pueden clasificarse por el tipo de sensor (infra rojo, audio y vídeo), por la multiplicidad del sensor (monocular o estéreo) y por el emplazamiento del sensor (centralizado o distribuido).

La forma más apropiada de clasificar los sistemas de video vigilancia es clasificarlos en tres generaciones que se distinguen por la tecnología empleada en ellos. La primera generación la forman los sistemas de circuito cerrado de televisión o CCTV. Se les conoce por el nombre de circuito cerrado porque todos los componentes que lo forman están enlazados entre sí. Este conjunto de sistemas están formados por un grupo de cámaras distribuidas que se sitúan en la zona de vigilancia y se conectan a un conjunto de monitores que normalmente se ubican en una sala central. Este tipo de sistemas son completamente analógicos y necesita de la supervisión de personas para hacer posible la detección de sucesos de interés. A pesar de sus limitaciones proporcionan una baja tasa de errores y son sistemas ampliamente utilizados debido en gran parte a la madurez de la tecnología que emplean. Sin embargo utilizan técnicas analógicas para el tratamiento, distribución y almacenamiento de las imágenes, y dependen demasiado de la actividad humana para detectar las anomalías que suceden en el entorno.

Los sistemas de segunda generación combinan la tecnología de los sistemas CCTV analógicos con los sistemas digitales de vigilancia-IP. El objetivo es intentar reducir la dependencia que existe de la actividad humana interpretando automáticamente los eventos y comportamientos que se producen en el entorno monitorizado. Estos sistemas permiten incrementar la eficiencia de los sistemas de seguridad, observando y analizando un mayor número de situaciones al mismo tiempo. También reducen la presencia del factor humano para detectar situaciones anómalas. Actualmente no existe una solución que permita realizar un razonamiento general sobre cualquier situación, solo existen soluciones parciales que interpretan soluciones muy concretas. Además existen situaciones anómalas imprevisibles que podrían no ser detectadas, y algunas de las soluciones

propuestas no son demasiado robustas, dando lugar a un número elevado de falsas alarmas.

1.1.2 Aplicaciones de sistemas de video vigilancia

Existen diferentes tipos de sistemas de video vigilancia entre los que destacan:

➤ **Vídeo detección de intrusos**

Un sistema de vídeo detección de intrusos, es una tecnología capaz de detectar accesos no autorizados por personas en áreas de seguridad o restringidas como aeropuertos, centros de detención de máxima seguridad o instalaciones nucleares. La función principal de este tipo de sistemas es la de detectar anomalías que puedan ser indicios de intrusión o alarmas. Para hacer posible la detección de intrusos, se recopila información de múltiples fuentes del sistema, analizando los datos obtenidos de diferentes maneras. Algunas de estas consisten en hacer un análisis estadístico sobre la información, buscando patrones que indiquen actividad anormal.

➤ **Vídeo Tracking**

Se entiende por Video Tracking al proceso de localización de un objeto en movimiento (o varios) y su posterior seguimiento en el tiempo mediante el uso de cámaras de vídeo. Se utiliza en gran cantidad de aplicaciones, por ejemplo: interacción hombre-máquina; seguridad y vigilancia; compresión y comunicación de vídeo; realidad aumentada; control de tráfico, imágenes médicas y edición de vídeo. Los sistemas visuales de seguimiento tienen dos principales componentes: representación y localización del objetivo, y filtrado de la información asociada. El primero utiliza una gran variedad de algoritmos para identificar el objeto en movimiento, por ejemplo: "Blob tracking", "Kernel-based tracking", "Contour Tracking", "Feature matching". La segunda necesita conocer información a priori de la escena o del objeto para poder hacer su seguimiento. Este algoritmo permite seguir objetos complejos incrementando notablemente la complejidad computacional. Los algoritmos más comunes son: filtro de Kalman y "Particle filter".

➤ **Detección de objetos abandonados**

Esta tarea consiste en localizar automáticamente objetos que fueron abandonados en una escena. Normalmente este tipo de objetos son bastante pequeños comparados con las personas y frecuentemente son ocultados por otras personas o vehículos que se mueven alrededor de la escena. Su funcionamiento se basa en utilizar algoritmos de segmentación que actúan en el primer plano de la secuencia. Los píxeles que no son parte del fondo y no se mueven se utilizan para encontrar regiones que contienen los posibles objetos abandonados. Sin embargo este tipo de sistemas son vulnerables a los problemas causados por fluctuaciones de iluminación, sombras y niveles de contraste de la escena.



Figura 2: Imágenes con detección

Fuente: <http://tecnocapacitacion.com/wp-content/uploads/2012/12/analisis2.jpg>

➤ **Conteo de objetos**

Los sistemas de conteo de objetos en entornos de vídeo son aquellos que son capaces de llevar la cuenta del número de objetos similares que aparecen en una escena en un determinado espacio de tiempo. Estos sistemas se basan en las características del objeto para su identificación. Existe un amplio campo de aplicaciones para este tipo de sistemas entre las que destacan el análisis de tráfico, el conteo de personas en zonas video vigiladas y procesos de fabricación.

1.1.3 Estado actual de sistemas que detectan movimiento desde cámaras en movimiento

Para Pearson (2004):

Una característica importante que remarca en el diseño de los procesamientos de imágenes es el nivel significativo de pruebas y experimentación que normalmente se requiere antes de llegar a una solución aceptable. Esta característica implica que la capacidad de formular enfoques y rápidamente prototipos de soluciones candidatas generalmente desempeña un papel importante en la reducción de costos y el llegar a un sistema viable de implementación. Es por ello la importancia de utilizar una herramienta importante que nos pueda ayudar a simular el diseño, como los resultados por poder realizar la implementación con costos mejorados. (p. 1)

La mayoría de los algoritmos de detección automática y de seguimiento están diseñados para configuraciones con cámara fija. Las técnicas basadas en sustracción de fondo o los algoritmos de diferenciación temporal no pueden usarse directamente para detectar movimiento con cámara activa. En los últimos años ha habido intentos para desarrollar sistemas de video vigilancia que sean capaces de operar con cámaras en movimiento dando como resultado una serie de algoritmos que, con restricciones, son capaces de resolver el problema.

1.2.3.1 Métodos que diferencian imágenes consecutivas alineadas

Estos métodos consisten en alinear previamente las imágenes para después hacer una diferenciación entre ellas. Básicamente la idea consiste en eliminar el movimiento aparente de la cámara para que en la diferenciación se descubran realmente los objetos que se mueven. Para que este tipo de algoritmos tenga éxito la mayor parte del movimiento aparente entre las dos imágenes debe ser causado por el movimiento de la cámara. De lo contrario se estaría intentando alinear con respecto a los objetos móviles que queremos detectar causando importantes errores en la detección del

verdadero movimiento. El alineamiento de las imágenes puede realizarse de tres formas distintas:

a) Alineamiento global

Este tipo de alineamiento se caracteriza por tener en cuenta toda la información de la imagen para realizar el alineamiento. A cada píxel de la imagen se le aplica una transformación global que permite poner en correspondencia las imágenes que se traten de alinear. La forma de obtener dicha transformación varía según el método que se aplique.

b) Template Matching

Este método consiste en aplicar una transformación directa a una imagen objetivo (Target) para adecuarla a una imagen de "plantilla"(Template). Las imágenes template y objetivo suelen ser consecutivas para facilitar la transformación que se produce con el movimiento de la cámara. Para transformar la imagen se parte de unas restricciones iniciales acerca del tipo de transformación que se desea hacer. Existen muchas transformaciones diferentes entre las que destacan las transformaciones simples (tres parámetros: traslaciones y rotaciones), transformaciones afines (seis parámetros) y transformaciones proyectivas (nueve parámetros). Una vez transformada la imagen se necesita medir que tan bueno ha sido el alineamiento. Para ello se utilizan unas medidas de similitud que nos darán información acerca de lo parecida que es la imagen alineada con respecto al Template. Existen muchas medidas de similitud entre las que destacan la correlación, el error cuadrático medio y la información mutua. Finalmente necesitamos un algoritmo de búsqueda que encuentre el valor de los parámetros de la transformación que hace que la similitud entre las dos imágenes sea máxima. El problema de este método suele ser el tiempo que conlleva el realizar la búsqueda de los parámetros que optimizan la medida. Conforme se quieran transformaciones más complejas, mayor número de parámetros se ven implicados en ella y por tanto más costosa es la búsqueda ya que se incrementan exponencialmente las combinaciones entre ellos. Sin embargo esta técnica puede ser adecuada para imágenes con baja

resolución, debido a la ventaja que se obtiene al utilizar toda la información para el alineamiento.

c) Derivada Temporal:

El método más utilizado en este tipo de solución es el de obtener el flujo óptico de la imagen para que a través de técnicas de agrupamiento detectemos la parte de la imagen correspondiente al fondo para su posterior alineamiento. El flujo óptico es el patrón de movimiento que existe entre una imagen y otra. Con él estimamos las velocidades instantáneas asumiendo que la variación de la intensidad entre las imágenes es pequeña. Una vez obtenido el flujo dominante (fondo) se utiliza la velocidad de dicho flujo para alinear el fondo y a continuación detectar con la diferencia entre las imágenes, los objetos de interés.

Para obtener las velocidades se realiza el desarrollo de Taylor y se deja el resultado en función de las velocidades V_x y V_y . Obtenemos una ecuación en la que se deben buscar las velocidades V_x y V_y que hacen que esa ecuación sea igual a cero. A partir de este punto tendremos una ecuación (velocidades) por cada píxel, es decir, la variación de intensidad entre una imagen y otra en las dimensiones x e y pueden relacionarse con sus componentes de velocidad V_x y V_y , por tanto tenemos un sistema de ecuaciones que no necesariamente tienen que tener solución única. Este sistema se puede resolver de diferentes formas. Entre los métodos que destacan en la literatura los más utilizados son los métodos que utilizan aproximaciones por mínimos cuadrados, o el método de Lucas Kanade, que para resolver el sistema impone unas restricciones considerando que el flujo situado en la vecindad del píxel que estamos considerando es constante

d) Modelo Elástico

Los anteriores métodos se basan en transformar directamente toda la imagen para así poder alinearla con su antecesora. El método del modelo elástico propone asumir que la imagen puede ser un material elástico, es decir, las deformaciones de una zona de la imagen no tienen por qué afectar a otras, por lo que la imagen que pretendemos alinear puede adaptarse

completamente a la imagen previa deformando, según convenga, unas zonas u otras con independencia de las deformaciones en el resto de la imagen. La imagen se va a flexionar y estirar y su cantidad estará caracterizada por la energía del material elástico. El objetivo no será otro que el de minimizar esta energía. Esto con lleva una gran carga computacional volviendo de nuevo al problema del coste que supone utilizar un algoritmo que deformando la imagen converja a un óptimo proporcionado por algún tipo de medida.

e) Alineamiento por características

A diferencia del alineamiento global, el alineamiento por características ya no utiliza toda la información de la imagen para realizar el alineamiento. En su lugar, se utilizan algoritmos que buscan puntos claves en las imágenes que queremos alinear. Los puntos deben ser lo más robustos posible frente a las variaciones entre imágenes para que la puesta en correspondencia sea más eficaz. Tal y como aparece en la literatura este método se puede dividir en tres fases.

- **Extracción de puntos característicos**

Existen diversos métodos en la literatura que extraen puntos de la imagen. Entre ellos destaca el algoritmo de Harris Stephens. Este algoritmo se basa en el principio de que en una esquina la intensidad de una imagen cambiará en gran medida en múltiples direcciones. Para detectar ese cambio de intensidad se utiliza una ventana que se desplaza alrededor de la esquina para detectar la dirección de donde proviene el cambio. Otro algoritmo muy utilizado es el detector de SUSAN. La idea consiste en suponer que en torno a la esquina existen dos áreas bien definidas: un área correspondiente a los píxeles que pertenecen a la esquina y otra área correspondiente a los píxeles que están fuera de la esquina. Se define un círculo que va situando su centro en los píxeles de la imagen y cuando el área que intersecciona con el círculo compuesta por los píxeles parecidos a ese centro es mínima se detecta esa zona como una esquina. Los algoritmos que más se suelen utilizar son los detectores SIFT/SURF. En el algoritmo de SIFT los puntos característicos son los máximos y los mínimos de las diferencias entre funciones gaussianas aplicadas a distintas escalas en la imagen suavizada.

Los puntos candidatos de bajo contraste y los de borde son descartados para finalmente asignar un descriptor para cada punto con su orientación y escala.

- **Correspondencia entre puntos (Matching)**

Una vez extraídos los puntos en las dos imágenes, para que puedan alinearse se necesita identificar el par de puntos/esquinas que son iguales en las dos imágenes. Como solución se proponen fundamentalmente dos tipos de métodos:

Correspondencia basada en flujo óptico: consiste en utilizar la idea del flujo óptico para realizar el “matching” de los puntos. En lugar de obtener una ecuación para cada píxel de la imagen, se obtiene una ecuación por cada punto característico de la imagen para después resolver el sistema utilizando algún método resolutivo como el de Lucas-Kanade o mínimos cuadrados.

Correspondencia basada en similitud de descriptores: para que funcione este método cada punto característico obtenido debe tener asociado un descriptor. Este descriptor contiene información acerca de la escala y orientación. Seguidamente se calculan las distancias entre descriptores de un punto característico en la imagen fuente frente a todos los puntos característicos de la imagen destino. Aquella distancia mínima de entre todas servirá para identificar el par de puntos correspondidos entre la imagen fuente y la de destino. El proceso se repite para el resto de puntos característicos de la imagen fuente.

Transformación de la imagen. Establecida la correspondencia entre puntos, buscaremos una transformación que ponga en correspondencia unos puntos con otros. Según el tipo de transformación que queramos hacer (simple, afín y proyectiva) necesitaremos más o menos puntos para que la calidad de la transformación sea buena y así tener un buen alineamiento.

Alineamiento por parámetros intrínsecos de la cámara: La última técnica que utiliza alineamiento de imágenes como paso previo para detectar movimiento consiste en utilizar parámetros intrínsecos de la cámara para estimar el movimiento que se produce en ella. La relación que existe entre cada posición de los píxeles en las dos imágenes se puede obtener mediante la fórmula de Kanatani . Para pequeños desplazamientos de la lente central

desde el centro de rotación de la cámara, la fórmula de Kanatani relaciona con éxito los píxeles de las dos imágenes. Dada una inclinación inicial θ del sistema de la cámara, desplazamiento α e inclinación de rotación γ y donde f es la distancia focal. Sabiendo f , θ , γ , y α para cada posición (x_t, y_t) en la imagen actual, podemos calcular la posición (x_{t-1}, y_{t-1}) del correspondiente píxel en la imagen previa.

1.2.3.2 Métodos que detectan movimiento directamente en la imagen

Existe una segunda clase de método que no necesita alinear las imágenes para detectar los objetos en movimiento en cámaras móviles. Esta técnica intenta encontrar entre dos imágenes movidas los píxeles que no encajan con el patrón del movimiento general de la imagen. A esta técnica se la conoce como detección basada en distorsión de flujo y sirve para distinguir entre objetos estáticos y en movimiento. En las imágenes capturadas con la cámara en movimiento, ambos tipos de objetos generan flujo óptico. La idea es aprovechar el flujo para obtener el movimiento de los objetos directamente. Este método asume que los píxeles que se corresponden con el flujo global pertenecen al fondo de la imagen. Este flujo global se extrae ajustando un sistema de ecuaciones que relaciona los flujos de las dos imágenes. La parte de flujo no ajustado se considera el movimiento real de la escena de modo que la detección del movimiento es directa sin ningún tipo de alineación previa.

Según las técnicas analizadas hemos visto que existen múltiples métodos para detectar movimiento con cámaras activas.

La primera estrategia que hemos encontrado en la literatura consiste en deshacer el movimiento de la cámara para eliminar el movimiento aparente (Métodos A). Si la cámara nos ofrece parámetros acerca de la rotación, desplazamiento e inclinación, nos será más fácil y rápido alinear las imágenes utilizando dichos parámetros. Sin embargo en la mayoría de las situaciones esto no ocurre, por lo que deberemos analizar la imagen en busca de parámetros externos a la cámara. Las técnicas de alineamiento global utilizan todos los píxeles de la imagen para obtener una transformación que alinee la imagen con su antecesora. Por otro lado, las técnicas de alineamiento por puntos utilizan zonas aisladas de la imagen con

determinadas propiedades que facilitan la correspondencia entre ellas y permiten aplicar posteriormente una transformación que alinee las imágenes. Según la literatura las dos técnicas de alineamiento son igualmente validas para alinear las imágenes y corregir de este modo el movimiento de la propia cámara.

La tercera estrategia que hemos encontrado en la literatura abandona la idea de alinear las imágenes y se centra en la detección directa del objeto en movimiento (Métodos B). Aprovecha la distorsión del flujo óptico para identificar el movimiento real que ocurre en la secuencia grabada. Aunque no necesita alineamiento, el algoritmo es muy costoso debido a la cantidad de ecuaciones que deben resolverse simultáneamente. Además esta técnica introduce ruido en los píxeles de la imagen, ruido que deberá ser corregido por algún método adicional.

1.3 Bases teóricas

1.3.1 Introducción a las técnicas de procesamiento digital de señales

La imagen digital es la representación bidimensional de una imagen fuente empleando bits, unidad mínima de información compuesta por dígitos binarios (1 y 0), que se emplea a instancias de la informática y cualquier dispositivo de tipo digital.

Mientras tanto, la imagen digital podrá ser obtenida a partir de dispositivos de conversión analógica digital, tal es el caso de los escáneres y las cámaras fotográficas digitales, o en su defecto a través de programas informáticos pertinentes como ser realizando dibujos con el mouse.

Cabe destacar que es absolutamente posible la modificación de las imágenes digitales, mediante filtros se pueden añadir o suprimir determinados elementos que no se tienen o por el contrario quitar aquellos que no se quieren, asimismo, se puede modificar el tamaño de una imagen y en caso de ser necesario hasta grabar la misma en un dispositivo de almacenaje, tal es el caso de un CD o del propio disco duro de la computadora.

Respecto de su conformación, la mayoría de los formatos digitales respetan una misma estructura: cabecera con atributos: dimensión de la imagen y el tipo de codificación, entre otros; datos de la imagen en sí. Tanto los atributos como los datos de la imagen serán diferentes en cada formato.

Una vez digitalizada la imagen, puede considerarse como monocromática, susceptible de ser representada como una función $f(x,y)$, donde x e y denotan coordenadas espaciales, y el valor de f (intensidad) en cualquier punto (x,y) es proporcional al nivel de gris (brillo) de la imagen en ese punto. Una imagen digital bidimensional es una imagen $f(x,y)$ que ha sido separada en coordenadas espaciales y brillo (nivel de gris). Así mismo, una imagen digital puede ser convenientemente representada por una matriz I de tamaño $M \times N$ cuya fórmula es:

$$I = \begin{matrix} I(1,1) & I(1,2) & \dots & I(1,M) \\ I(2,1) & I(2,2) & \dots & I(2,M) \\ \vdots & \vdots & & \vdots \\ I(N,1) & I(N,2) & \dots & I(N,M) \end{matrix}$$

Los elementos de la matriz (píxeles) en una imagen monocromática típica son del orden 2 a la 8 o 256 niveles de gris, por lo tanto, pueden ser representados como caracteres en la mayoría de los lenguajes de programación. Las columnas y los renglones de las matrices de imágenes digitales.

1.3.2. Vecindad entre píxeles

Un píxel p en las coordenadas (x,y) tiene 4 vecinos horizontales y 4 verticales, cuyas coordenadas están dadas por la fórmula:

$$(x+1,y),(x-1,y),(x,y+1),(x,y-1)$$

Este grupo de píxeles se nota como $N_4(p)$. Así mismo, las vecindades diagonales con el punto (x,y) se notan como $N_d(p)$, y sus coordenadas son:

$$(x+1,y+1),(x+1,y-1),(x-1,y+1),(x-1,y-1)$$

Para definir de forma adecuada el concepto de vecindad, es necesario revisar el de adyacencia. Dos pixeles son adyacentes si, y solo si, tienen en común una de sus fronteras, o al menos una de sus esquinas.

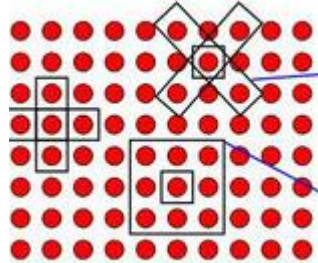


Figura 3- Adyacente por frontera y por esquina

Fuente: <http://procesamientografico-tiy.jimdo.com/laboratorios-1/laboratorio-1/>

Dos píxeles son vecinos si cumplen con la definición de adyacencia. Si los píxeles comparten sus fronteras, se dice que los mismos son vecinos directos; si solo se tocan en una de sus esquinas, se llaman vecinos indirectos.

Para ilustrar las definiciones anteriores, se puede observar en la figura, vecindades de 4 y vecindades de 8 está formada tanto por vecinos directos como indirectos.

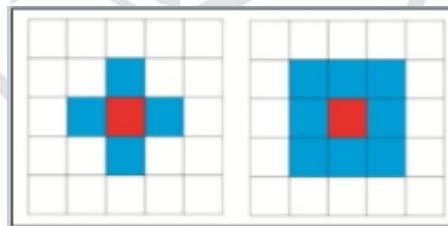


Figura 4- Vecindad de 4 y de 8

Fuente: <http://es.slideshare.net/lonely113/procesamiento-digital-de-imagenes-con-matlab>

En el desarrollo de técnicas de procesamiento de imágenes que involucren el análisis de una determinada región de la escena digital, es posible encontrar vecindades de 5x5 y hasta de 9x9; básicamente la definición de las dimensiones de la matriz vecindad depende de la técnica que se esté desarrollando.

1.3.3. El espacio RGB y su almacenamiento

Uno de los espacios de color más utilizado es el RGB. Es un sistema basado en mapa de bits y la capacidad de los conos en el ojo humano de distinguir claramente el color rojo, verde y azul. Es un sistema de color aditivo, es decir la suma de todos los colores genera el blanco.



Figura 5- Generación de colores en el espacio RGB 8
Fuente: <http://www.grimaldos.es/cursos/imgdig/color.html>

Una imagen en color RGB basa su funcionamiento en la matriz bidimensional de una imagen en blanco y negro. Esta matriz está compuesta por los valores de luminancia de cada pixel que compone la imagen.

Trabajando con 8 bits se tiene un total de 256 valores posibles para cada pixel, que se traducen en diferentes tonalidades de grises. De esta manera se cubre toda la escala de grises, desde el valor 0, correspondiente al negro, hasta el valor máximo 255, que pertenece al blanco.

1.3.4. Histograma

Lo primero que hay que decir del histograma es que no es un concepto fotográfico, sino estadístico. Así, se trata de una herramienta para representar la frecuencia con que aparecen valores dentro de una serie, de manera que cada línea o barra vertical indica la frecuencia con la que una variable determinada toma un valor concreto.

En nuestro caso, en horizontal tenemos los valores de luminosidad, y en vertical el porcentaje de pixeles de la imagen que poseen ese valor. Para una imagen de "n" bits de color, los posibles valores en horizontal serán 2^n . Así, un JPG (8 bits) tendrá 256 valores.

En algunos casos esta información se muestra dividida en pasos de exposición, lo cual es muy práctico si queremos expresar al máximo la información que puede capturar nuestro sensor.

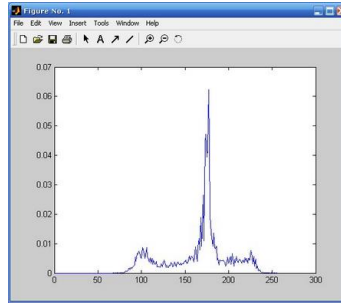


Figura 6- Histograma de una imagen
Fuente: Matlab.

1.3.5. Imagen digital

Una imagen digital es la representación bidimensional de una imagen empleando bits, unidad mínima de información compuesta por dígitos binarios (1 y 0), que se emplea en la informática y cualquier dispositivo del tipo digital.

De acuerdo a la resolución que presenta la imagen se podrá hablar de mapa de bits, representa una rejilla rectangular de pixeles o punto de color, la cual se puede suavizar en un monitor, en un papel o en cualquier otro dispositivo de representación que se emplee.

La imagen digital podrá ser obtenida a partir de dispositivos de conversión analógica digital, tal es el caso de los escáneres, las cámaras fotográficas y de video.

1.3.6 Pixel

Un píxel o pixel, es la menor unidad homogénea en color que forma parte de una imagen digital, ya sea esta una fotografía, un fotograma de vídeo o un gráfico.

Es la unidad mínima a partir de la cual se forma cualquier imagen en mapa de bits. Para la representación de un pixel se puede utilizar 1 bit, 2 bits, 4 bits... de información. Este valor determinará el número de colores que

puede representar dicho píxel; como es evidente, a más bits, más colores y, por tanto, mejor calidad de imagen, mayor tamaño de la misma.

En un mapa de bits el contenido se representa mediante pequeños puntos rectangulares denominados píxeles. Ampliando lo suficiente una imagen digital (zoom) en la pantalla de un ordenador, pueden observarse los píxeles que componen la imagen.

En las imágenes de mapa de bits o en los dispositivos gráficos cada píxel se codifica mediante un conjunto de bits de longitud determinada (la llamada profundidad de color); por ejemplo, puede codificarse un píxel con un byte (8 bits), de manera que cada píxel admite 256 variaciones (28 variaciones con repetición de 2 valores posibles en un bit tomados de 8 en 8). En las imágenes de color verdadero, se suelen usar tres bytes para definir un color; es decir, en total podemos representar un total de 224 colores, que suman 16.777.216 opciones de color (32 bits son los mismos colores que 24 bits, pero tiene 8 bits más para transparencia).

1.3.7 Resolución espacial

Es la finura de detalles visibles en una imagen, cuanto menor es el área terrestre representada por cada píxel en una imagen digital, mayores son los detalles que pueden ser captados y mayor es la resolución espacial. En las Figs. 26 a, b y c se observan imágenes de una misma región a tres diferentes resoluciones espaciales (baja, mediana y alta, correspondiendo respectivamente al MODIS 1.1 km, LANDSAT-5, 30m y QUICKBIRD, 0.60m. Existen diferentes factores que influyen en la resolución espacial, además de los obviamente asociados al sensor, como ser el poder resolutivo del sistema óptico, debemos considerar las influencias atmosféricas, presencia de humo, neblina, bajos niveles de iluminación, etc.

También existen factores asociados a las características del objeto a detectar y de su entorno, un buen contraste y bordes nítidos del objeto favorecen la detección. La selección de la resolución espacial necesaria para un proyecto debe analizarse cuidadosamente. En efecto, pretender

resoluciones mayores que la necesaria encarece el costo de las imágenes y la carga de proceso.

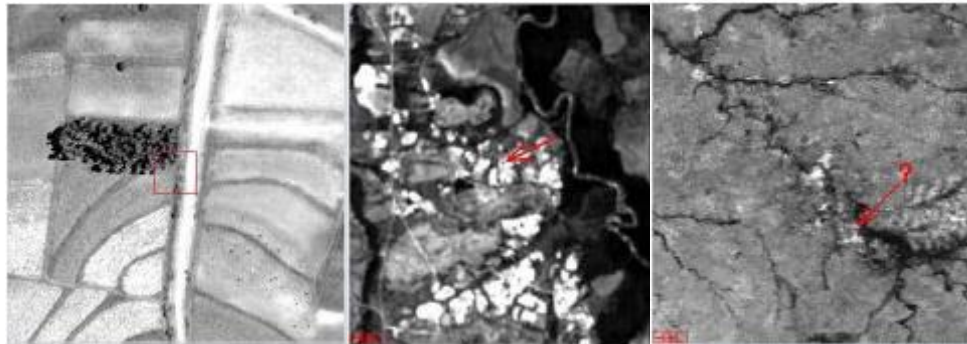


Figura 7- Menor, media y mayor resolución

Fuente: <http://www.teledet.com.uy/tutorial-imagenes-satelitales/satelites-resolucion-espacial.htm>

1.3.8 Profundidad de color

Las imágenes digitales son una superficie rectangular dividida en pequeños cuadrados llamados píxeles, teniendo cada uno un color definido. La cantidad de píxeles que se usa para representar la imagen se llama resolución.

El color de cada píxel es asignado por medio de bits, como sucede en todo lo que se digitaliza.

Un bit es la unidad mínima de medida de memoria de los archivos informáticos. Es un número binario: puede tener dos valores, cero o uno.

Llamamos Profundidad de color a la cantidad de colores que pueden tener una imagen digital. Esta "medida" de las imágenes es de suma importancia.

Las resoluciones disponibles están relacionadas con el modo de color y son:

- De 1 bit por píxel para los mapas de bits.
- De 8 o 16 bits por canal para las imágenes en escala de grises o color RGB.

Una imagen de un bit por píxel tendrá un solo número binario para definir el color de cada píxel; sólo podrán tener dos colores, usualmente blanco (0) y negro (1). Con 2 bits por píxel la imagen podrá contar con cuatro colores

(0,0 - 0,1 - 1,0 - 1,1). Análogamente podemos tener imágenes de 3, 4 o más bits por pixel, con una creciente cantidad de tonos de gris. Este tipo de imágenes no son muy usuales y pocos formatos digitales las soportan.

Cuando se asignan 8 bits por píxel tenemos una imagen en escala de grises. Un número binario de ocho cifras permite definir 256 tonos de gris, desde el blanco hasta el negro: una cantidad suficientemente grande de tonos para describir una foto en blanco y negro.

Las imágenes en color se describen apelando a algún sistema o espacio de color. Los más usados son los RGB y CMYK. Una imagen RGB está compuesta por tres canales, uno por cada color primario luz (rojo, verde y azul). La superposición de los tres colores primarios permite describir una amplia gama de colores. En la figura 3 vemos una misma imagen en diferentes profundidades de color.

1.3.9 Modelo de escala de grises

El modelo de color Escala de grises define los colores solo con un componente, la luminosidad, y se mide en valores de 0 a 255. Cada color de la escala de grises tiene los mismos valores de los componentes rojo, verde y azul del modelo de color RGB.



Figura 8- Imagen normal e imagen en escala de grises

Fuente: <http://www.cristalab.com/tutoriales/los-modos-de-color-en-photoshop-c105006/>

1.3.10 La media

La media o valor medio de una distribución de probabilidad se representa por μ , y se define por:

$$\mu = \begin{cases} \sum x_i f(x_i) & , \text{si } X \text{ es discreta} \\ \int x f(x) dx & , \text{si } X \text{ es continua} \end{cases}$$

En estas dos fórmulas $f(x)$ es la función de probabilidad y la función densidad de probabilidad respectivamente de la variable aleatoria X en consideración.

Conviene mencionar que la media μ se conoce como esperanza matemática de X o, brevemente, esperanza de X , y se representa por $E(X)$.

Que quiere decir que 3.5 es el valor esperado, lo que significa que 3.5 es el valor central de la distribución. Obsérvese que no es necesario que el valor esperado sea un valor posible de la variable aleatoria. También se interpreta en el sentido de que en 10 ejecuciones del experimento, por ejemplo, se espera que la suma de los números obtenidos sea de $(10) \cdot (3.5) = 35$.

1.3.11 Varianza

La varianza de una distribución se representa mediante σ^2 y se define por:

$$\sigma^2 = \begin{cases} \sum (x_i - \mu)^2 f(x_i) & , \text{si } X \text{ es discreta} \\ \int (x - \mu)^2 f(x) dx & , \text{si } X \text{ es continua} \end{cases}$$

Donde $f(X)$ representa a la función de probabilidad y a la función densidad de probabilidad, respectivamente, de la variable aleatoria.

Claramente $\sigma^2 \geq 0$ porque $(X - \mu)^2 \geq 0$, para todo X , y $f(X) \geq 0$, para todo X . A la raíz cuadrada positiva de la varianza la llamamos desviación estándar y se denota s .

La varianza es una medida de dispersión o variabilidad que no tiene interpretación física ya que está en unidades cuadradas.

Si en las fórmulas anteriores desarrollamos el cuadrado del binomio y aplicamos propiedades de las sumatorias (integrales) se llega a una expresión más conveniente para realizar los cálculos:

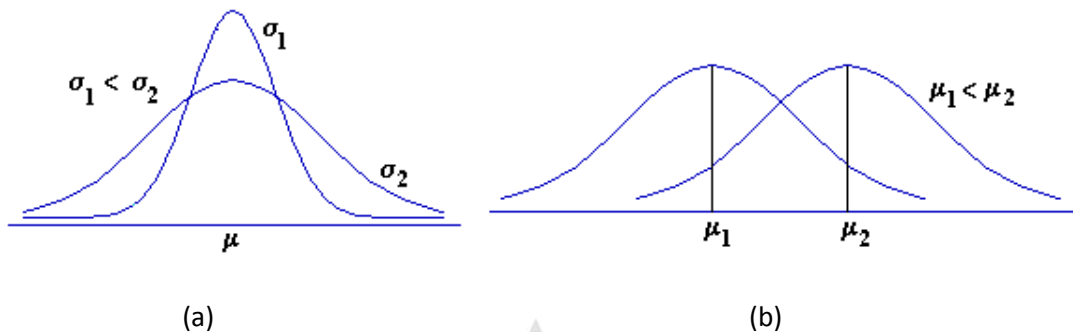


Figura 9- Igual media y diferente dispersión, viceversa

Fuente: Propia.

1.3.12 Cámara IP

Una cámara IP (también conocidas como cámaras web o de red) son videocámaras especialmente diseñadas para enviar las señales (video, y en algunos casos audio) a través de internet desde un explorador (por ejemplo el internet explorer) o a través de concentrador (un hub o un switch) en una red local (LAN).

En las cámaras IP pueden integrarse aplicaciones como detección de presencia (incluso el envío de mail si detectan presencia), grabación de imágenes o secuencias en equipos informáticos (tanto en una red local o en una red externa (WAN), de manera que se pueda comprobar el porqué ha saltado la detección de presencia y se graben imágenes de lo sucedido.

Lo más importante para poder usar una cámara IP es disponer de una conexión a Internet si tenemos intención de poder las imágenes en una red externa, para ello conecto la cámara IP a un Router ADSL, XDSL, o Cable modem (o a un HUB) u otros sistemas de banda ancha.

1.3.13 Componentes

Básicamente una cámara IP se compone de:

La " cámara " de video tradicional (lentes, sensores, procesador digital de imagen, etc.

Un sistema de compresión de imagen (para poder comprimir las imágenes captadas por la cámara a formatos adecuados como MPEG4.

Un sistema de procesamiento (CPU, FLASH, DRAM y un módulo Wireless ETHERNET/WIFI). Este sistema de procesamiento se encarga de la gestión de las imágenes, del envío al modem. Del movimiento de la cámara (si dispone de motor), de la detección de movimiento.

Con todo esto únicamente necesitamos conectar la cámara al Router ADSL y a la alimentación eléctrica y no necesitamos nada más o si pensamos usar la cámara en una red local, lo conectamos a un HUB/SWITCH y pasa a ser un equipo más que se comunica con el resto de la LAN (y con el exterior si la LAN dispone de conexión a Internet).

1.3.14 Cámara IP sobre un CCTV convencional

Una cámara IP te aporta grandes ventajas frente al tradicional CCTV:

Posibilidad de acceso desde cualquier sitio del mundo. Un CCTV es, como su nombre indica, "cerrado", por ello hay que estar en el lugar del CCTV para poder ver las imágenes.

Las cámaras IP son más baratas que las cámaras analógicas comunes.

Instalar cámaras IP es muy sencillo ya que es como instalar una red local LAN o conectarla directamente al Router (inalámbrico o con cables, existen ambas opciones).

No se necesita las complicadas y caras instalaciones de CCTV.

Es muy sencillo añadir más cámaras IP a un sistema, mientras que en un CCTV necesitamos duplicar sistemas de monitorización durante la ampliación del sistema.

Las cámaras IP y los servidores de vídeo solamente necesitan conectarse directamente a un PC mediante un cable de red "cruzado" cuando se instalan por primera vez.

Una vez instalada, cualquier modificación de la configuración, de los ajustes de calidad de imagen, de las contraseñas de acceso, etc. se realizará de forma remota desde cualquier punto del mundo, bastará con conectarse a la cámara en modo "administrador".

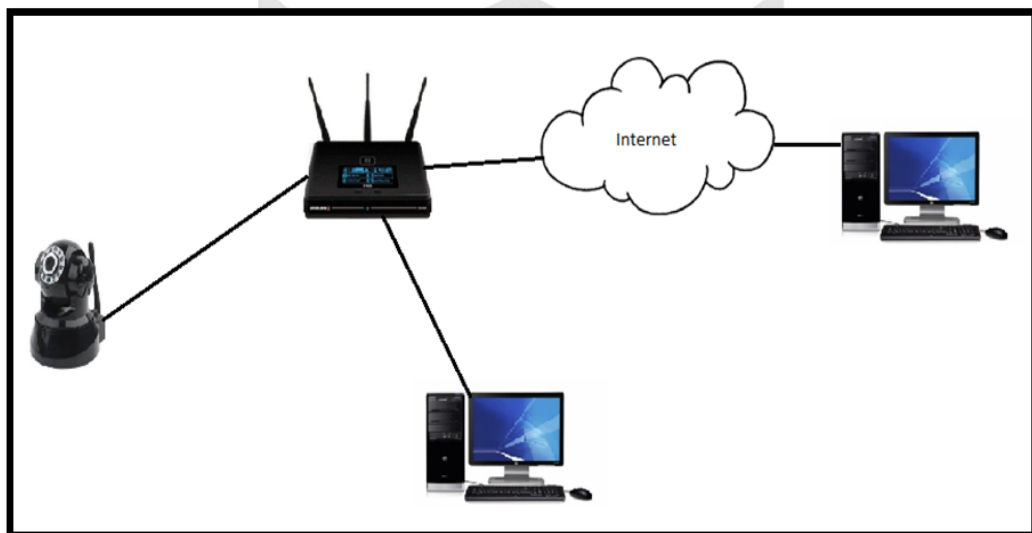


Figura 10- Cámara IP, router, pc e internet
Fuente: Propia.

1.3.15 Incidente

Un incidente es aquello que se interpone en el transcurso normal de una situación o de un evento. El sistema detectara el bus, y si existe una persona en el embarque, luego que el bus haya partido.

Al existir una persona en el embarque, luego de la partida del bus, se considera como un incidente, ya que no debe existir.

Es por eso que se toma alguna acción rápida al detectarla, para que no ocurra algún accidente.

1.4 Propuesta de la Solución

Para la presente tesis, se investigó cómo se podría monitorear de manera eficiente los incidentes en los embarques del Metropolitano. Gran porcentaje de accidentes ocurren en estas plataformas, debido a que la gente se queda algunas veces atrapadas entre la puerta del paradero y la puerta del Metropolitano, y otro grupo quiere ser el primero en subir.

Para estos dos casos, la probabilidad que pase un accidente es alta, es por ello que se diseñó un sistema de tal forma que detecte presencia de personas en los embarques, cuando estos no tengan algún bus estacionado. O de lo contrario cuando este ya se haya retirado.

El sistema consiste básicamente, con ayuda de una cámara IP y para el caso de estudio, utilizando el Matlab, el sistema nos pueda avisar cuando llega un bus al paradero, y luego nos informa si existe alguna persona en la plataforma, luego que el bus se retiró.

De tal forma que cuando se tenga el caso de detectar una persona, automáticamente alerte al centro de control, quien controla las cámaras de seguridad, y se ponga en pantalla principal, la cámara con el incidente, así el personal pueda dar alerta que existen personas en la plataforma, o mandar ayuda en caso pasase un accidente.

Alcance

EL alcance de la presente tesis es la detección del bus, y luego detectar si existe alguna persona, para luego enviar una alerta. Todo esto en condiciones normales a la luz del día. Se utilizo un video grabado con la cámara IP obtenida para las pruebas.

CAPÍTULO II

METODOLOGÍA

2.1 Material

Los materiales que se usaron para poder realizar las simulaciones del sistema propuesto, fueron hardware y software.

2.1.1 Laptop Lenovo T430

Tabla N° 02: Características de PC.

Machine name: LAPEVERGARA
Operating System: Windows 7 Professional 32-bit (6.1, Build 7601)
Service
Pack 1 (7601.win7sp1_gdr.140303-2144)
Language: Spanish (Regional Setting: Spanish)
System Manufacturer: LENOVO
System Model: 2349TKA
BIOS: Phoenix BIOS SC-T v2.2
Processor: Intel(R) Core(TM) i5-3230M CPU @ 2.60GHz (4 CPUs), ~2.6GHz
Memory: 4096MB RAM
Available OS Memory: 3308MB RAM
Page File: 4569MB used, 2044MB available
Windows Dir: C:\windows
DirectX Version: DirectX 11
DX Setup Parameters: Not found
User DPI Setting: Using System DPI
System DPI Setting: 96 DPI (100 percent)
DWM DPI Scaling: Disabled
DxDiag Version: 6.01.7601.17514 32bit Unicode:
Display Devices

Card name: Intel(R) HD Graphics 4000
Manufacturer: Intel Corporation
Chip type: Intel(R) HD Graphics Family
DAC type: Internal
Device Key:
Enum\PCI\VEN_8086&DEV_0166&SUBSYS_21F317AA&REV_09
Display Memory: 1461 MB
Dedicated Memory: 64 MB
Shared Memory: 1397 MB
Current Mode: 1366 x 768 (32 bit) (50Hz)
Monitor Name: ThinkPad Display 1366x768
Monitor Model: unknown
Monitor Id: LEN40A0
Native Mode: 1366 x 768(p) (59.998Hz)
Sound Devices
Description: Altavoces (2- Dispositivo de High Definition Audio)
Default Sound Playback: Yes
Default Voice Playback: Yes
Hardware ID:
HDAUDIO\FUNC_01&VEN_10EC&DEV_0269&SUBSYS_17AA21F3
&REV_1002
Manufacturer ID: 1
Product ID: 65535
Type: WDM
Driver Name: HdAudio.sys
Driver Version: 6.01.7601.17514 (Spanish)
Driver Attributes: Final Retail
WHQL Logo'd: Yes
Date and Size: 11/20/2010 16:29:03, 304128 bytes

Fuente: Marca Lenovo Laptop



Figura 11- Laptop Marca Lenovo

Fuente: <https://pixabay.com/es/ordenador-port%C3%A1til-port%C3%A1til-equipo-158648/>

Cámara IP WansView

Especificaciones técnicas

Tabla N° 03: Características de la cámara.

Camera	Wireless IP Camera
Model	NC541/W
Resolution	640x480(VGA)/320x240(QVGA)/160x120(QVGA)
Color	Black
Image Sensor	1/4" Color CMOS Sensor
Display Resolution	640x480 Pixels (300k Pixels)
Lens	f: 4.5mm, F:2.0 (IR Lens)
Mini Lumination	0 Lux
Lens Type	Glass Lens
Viewing Angle	67 Degree
Input	Built-In Microphone
Output	Built-In Speaker
Audio Compression	ADPCM
Image Compression	MJPEG
Image Frame Rate	15fps(VGA), 30fps (QVGA)

Bit Rate	128kbps~5Mbps
Flip Mirror Images	Verical / Horizontal
Light Frequency	50Hz, 60Hz or Outdoor
Video Parameters	Brightness, Contrast
Ethernet	10Base-T/100base-TX
Wireless Standards	IEEE 802.11 b/g
Wireless Security	WEP & WPA Encryption
Pan/Tilt Angle	Horizontal: 270°ree; & Vertical: 120°ree
Infrared Light	10 IR, LEDs, Night visibility upto 5 meters
Gross Weight	No
Net Weight	No
CPU	No
Memory Size	No
Supported OS	Microsoft Windows 2000 / XP / Vista / 7
Browser	IE 6.0, IE 7.0, Firefox and other standard browser
Certification	CE, FCC



Figura 12- Cámara IP Marca VanWiew

Fuente: <https://soloelectronicos.com/2013/07/05/acceso-a-tu-camara-ip-desde-internet-con-tu-home-station/>

Software

Matlab (V.8.1.0.604)



Figura 13- MatLab
Fuente: Matlab.

Sistema operativo windows 7 professional



Figura 14- Sistema operativo Windows 7
Fuente: PC.

Microsoft office 2007

Microsoft office word 2007

Microsoft office excel 2007

Microsoft office power point 2007

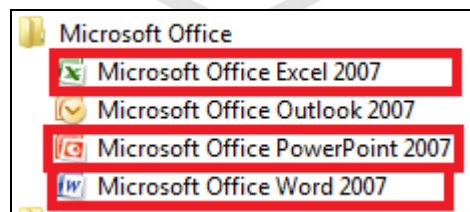


Figura 15- Office 7
Fuente: PC.

2.2 Métodos

2.2.1 Método de captura de la imagen, utilizando una cámara IP.

Primeramente antes de poder realiza cualquier procesamiento de señales digitales, necesitamos poder realizar las tomas desde la cama IP de ensayo,

de tal forma que sea compatible con el software utilizado para procesarla, en este caso el Matlab.

Se tuvo varios intentos fallidos al momento de poder reconocerla, ya que se podía hacer fácilmente con una webcam o la misma cámara de la laptop usada.

En este intento tuvimos tiempo invertido, y al final lo pudimos realizar.

Debido al funcionamiento de la cámara, pudimos observar que esta, guardaba las fotos tomadas en un archivo en la memoria de computador, cada vez que se mandaba la orden de toma.

Lo que se hizo mediante comandos de Matlab, fue hacer que mande la orden de toma, vaya a la carpeta de guardado, y mostrarla mediante el matlab.

Ya teniendo la imagen capturada, la procesamos de tal forma que podamos tener tomas en casi tiempo real, por un mínimo de desfase de tiempo, ya que hay un pre-procesamiento para poder realizar las capturas en línea. Y seguir con los procesamientos.

Se muestra el programa en el anexo 01.

2.2.2 Método de plano cartesiano para análisis de zona de interés.

Para poder realizar el procesamiento del video, se hizo el procesamiento a las imágenes por las cuales está compuesto el video.

En este caso los frames por segundo con esta cámara es de 25. Esto quiere decir que tenemos que hacer el procesamiento a cada frame.

Lo primero que tenemos que realizar teniendo los frames tomados por el Matlab, es poder segmentar de toda la imagen, solo el área de interés, que para nuestro caso tendremos 02 aéreas de interés.

Primer Área de Interés: Esta área de interés está enfocada a poder detectar el Bus, cuando este llega a su embarque correspondiente.

El enfoque está dado al faro posterior del autobús, debido a que de todo el lado trasero del bus, esta mascarilla del faro es rojo, que luego será procesado.



Figura 16- Comienzo de tamaño de la Imagen (1,1)

Fuente: Propia.



Figura 17- Fin de tamaño de la imagen (640,480)

Fuente: Propia.

Del total de la imagen la primera área de interés estará dado por las coordenadas en Matlab de (373:450,480:570)



Figura 18- Primera área de interés de la imagen total

Fuente: Propia.

Los comandos para poder elegir esta área se muestra en el anexo 02:

2.3.3 Método de binarización para poder encontrar el peso del color rojo en el área de interés.

Mediante este método lo que hacemos es binarizar la imagen, pero antes aplicamos 02 procesamientos, para poder ir filtrando el color rojo.

Tomamos un muestreo de los pixeles de que están en dentro del área de interés, y luego mandamos a cero los que contengan la componente R en menor cantidad.

Utilizando un muestreo para poder hallar los umbrales.

Tabla N° 04: Umbrales.

R	G	B
138	53	54
132	53	54
126	52	52
133	56	53
123	49	57
138	54	56
132	54	53
126	54	53
133	52	52
133	53	56
123	57	53
138	56	53
133	53	54
123	53	54
138	52	52

Fuente: Propia.

Los comandos se muestran en el anexo 03:

Obtenemos esta imagen como resultado:



Figura 19- Imagen resultado.
Fuente: Propia.

2.3.2 Primera Imagen procesada

Como vemos aún tenemos componentes que debemos filtrar, para eso aplicamos un segundo procesamiento, teniendo como resultado el siguiente:



Figura 20- Imagen con segundo resultado.
Fuente: Propia.

2.3.2 Segunda imagen procesada

Aquí la tenemos el solamente el color rojo de la imagen de la primera área de interés.

Los comandos se muestran en el anexo 04.

Luego de esto binarizamos la imagen para poder hallar la energía de la misma, y poder tener como resultado un número escalar.

Teniendo como resultado de varias muestras la siguiente tabla:

Tabla N° 05: Resultados.

Hay Rojo	No hay Rojo
340	122
350	270
325	210
398	100
310	222

367	210
378	111
320	250
310	240
398	260
310	280
398	210
310	203
367	211
309	269

Fuente: Propia.

Con esta tabla podemos discriminar cuando hay rojo (existe el bus), o cuando no hay rojo (no hay bus). Indicando que si la energía es mayor a 300 si hay bus, y cuando es menor no lo hay.

Los comandos se muestran en el anexo 05.

La siguiente área de interés es el área del primer embarque:

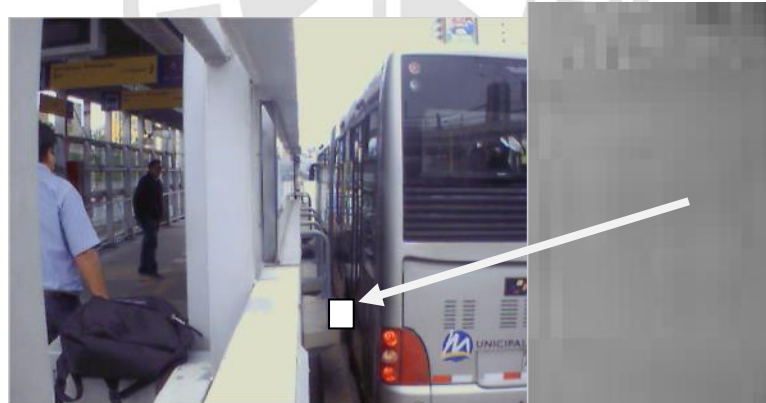


Figura 21- Segunda área de interés.

Fuente: Propia.

2.3.4 Método del cálculo de la varianza en el área de interés

Primero lo que procesamos en el área de interés, lo pasamos a escala de grises, para poder sacar su histograma.

Una vez sacado el histograma podemos hallar su varianza, teniendo como resultado otro número escalar, del cual sacamos la siguiente tabla con los resultados obtenidos.

Tabla N° 06: Listado de varianzas.

Persona No Presente	Persona Presente
43.523	502.123
55.342	452.221
109.221	22111.345
107.212	604.212
91.213	795.123
51.245	604.987
34.112	502.433
55.345	521.683
55.363	9888.234
110.346	4577.89
90.232	605.345
109.234	888.54
69.604	580.32
110.23	602.11
55.303	873.33

Fuente: Propia.

Por esta razón tomamos como umbral el escalar 450.

Los comandos se muestran en el anexo 06.

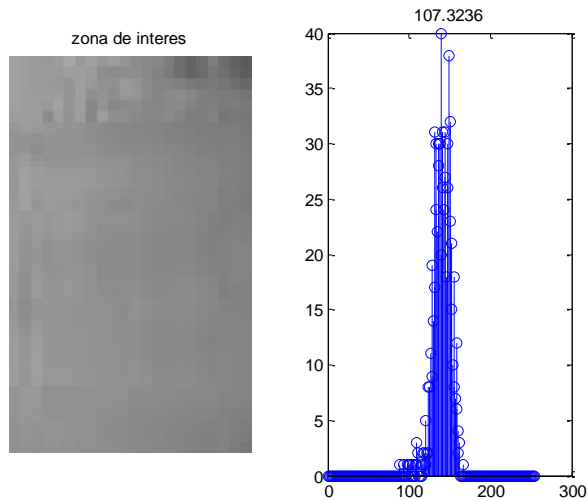


Figura 22- Imagen sin persona.
Fuente: Propia.

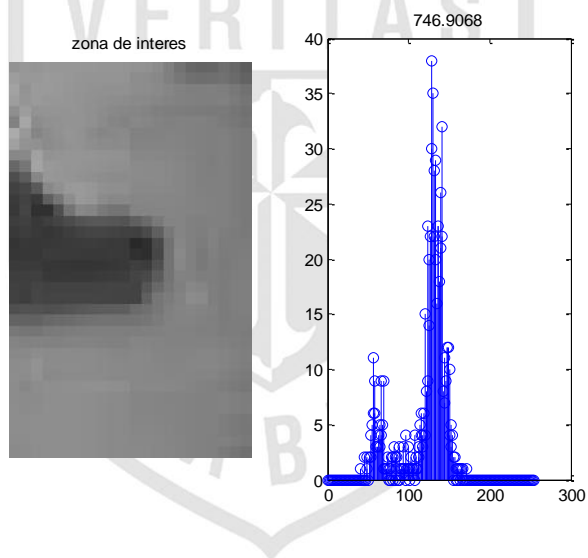


Figura 23- Imagen con presencia de persona.
Fuente: Propia.

Metodología de elección de la cámara IP

Inicialmente se hizo la elección de una cámara web para poder hacer las pruebas, pero al ver el campo de acción, se decidió por recurrir a una cámara IP.

Esta cámara debía cumplir lo mínimo para poder hacer el ejercicio y demostrar que la propuesta funcione, adicionalmente por su tamaño de

pixeles de los frames (640x480), no tendría sobrecarga en los procesamientos. Además no tener un costo elevado.

Por estas razones se optó por usar para el desarrollo de la tesis la cámara IP.

Metodología de elección de área de interés

Antes de recurrir al área de interés, se hizo la elección de la posición de la cámara para que pueda de alguna forma, enfocar directamente al área de acción, el cual era la plataforma de embarque a los buses.

Una vez hallada esta toma, se seleccionaron dos áreas de interés, la primera fue la máscara del faro de los buses, de color rojo, de tal forma que usamos las técnicas de procesamiento de imágenes para poder hallar ese rojo y confirmar que el bus estaba en su lugar correcto.

La segunda área de interés, es la plataforma, para poder detectar que existe una persona.

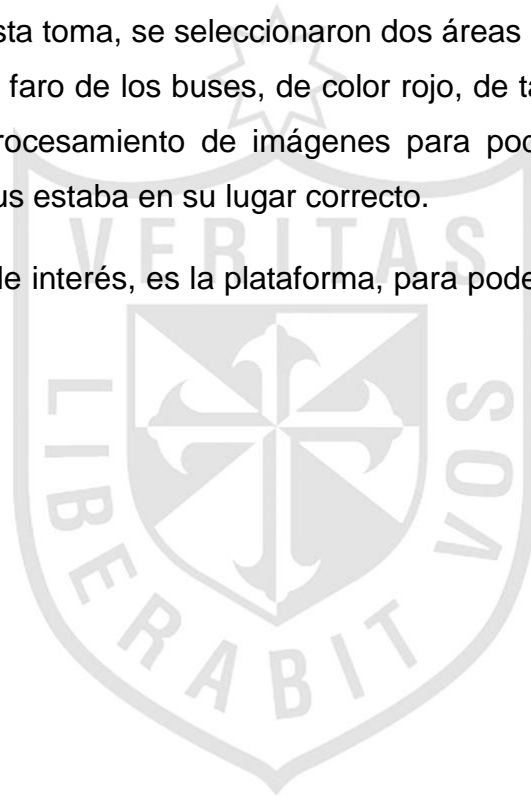


Diagrama de bloques de todo lo procesado

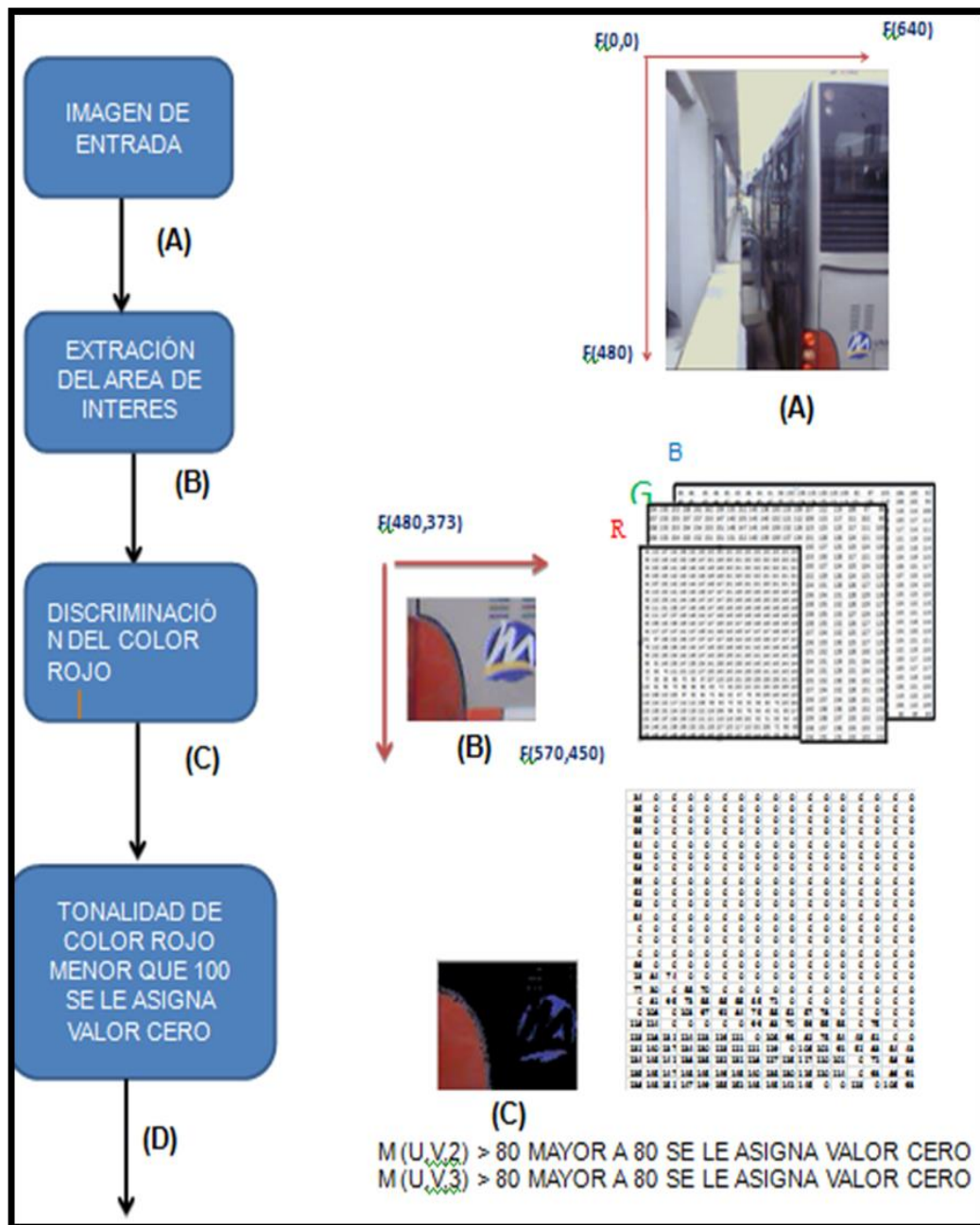


Figura 24- Diagrama de bloques.

Fuente: Propia.

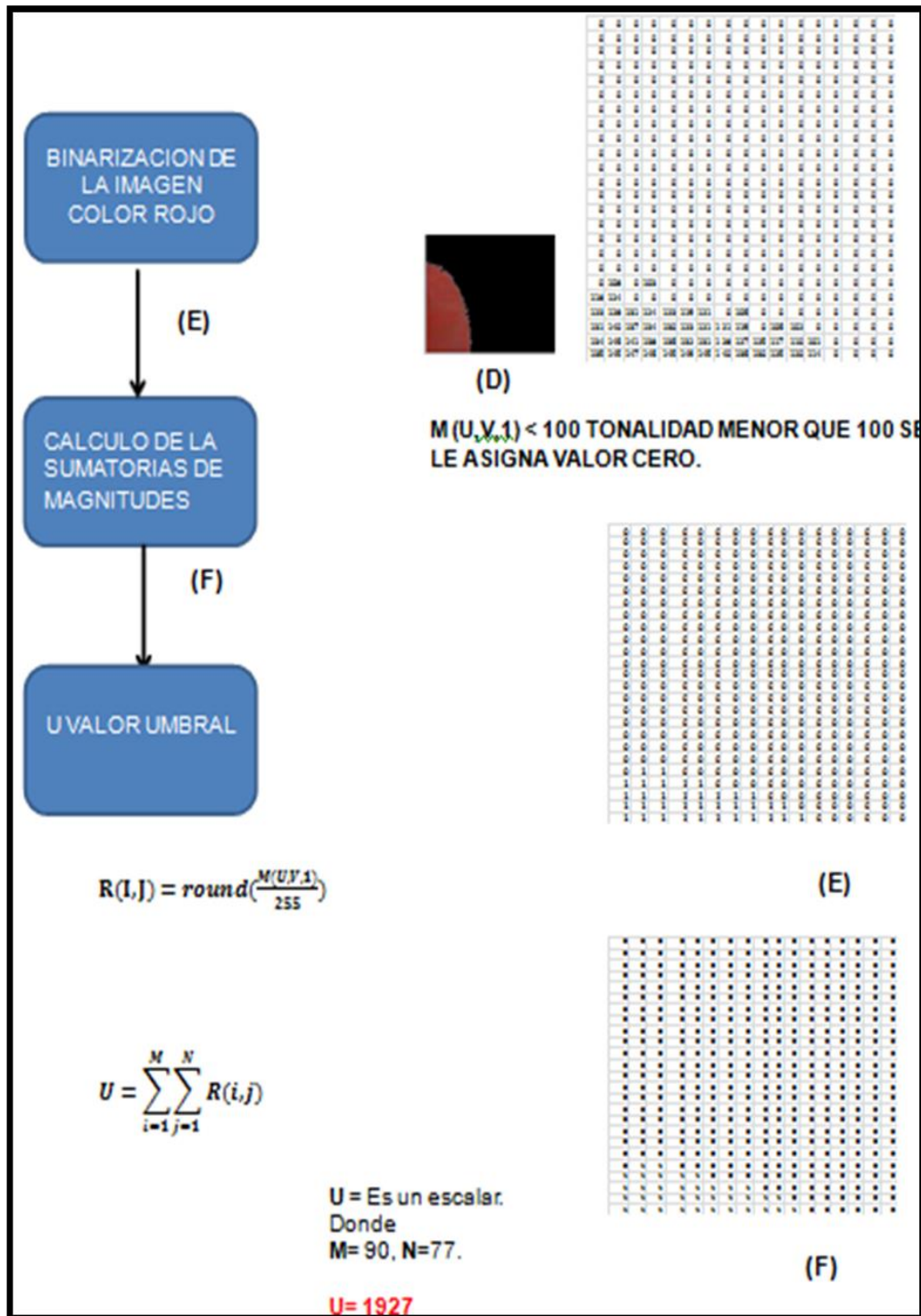


Figura 25- Diagrama de bloques.

Fuente: Propia.

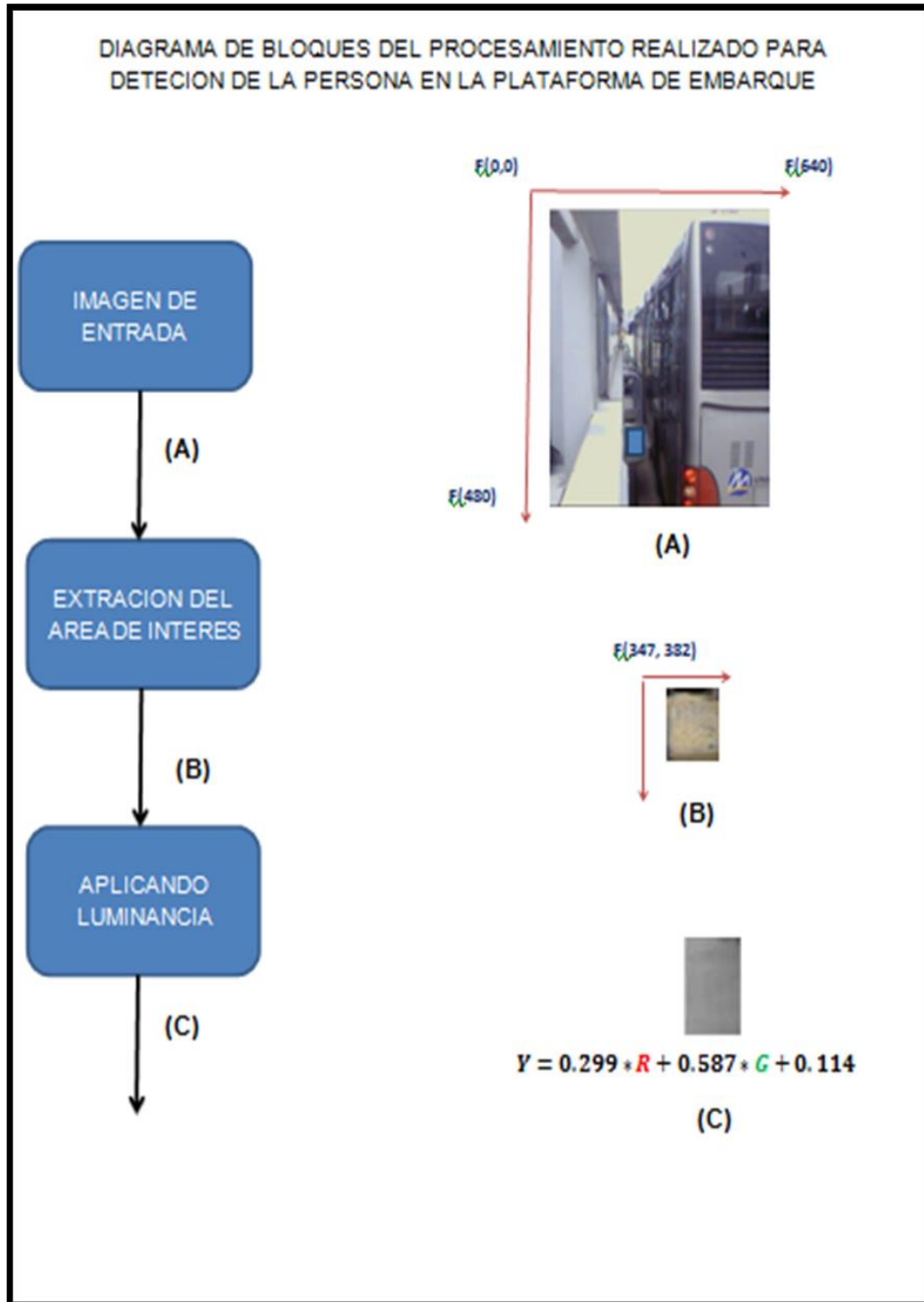


Figura 26- Diagrama de bloques.
Fuente: Propia.

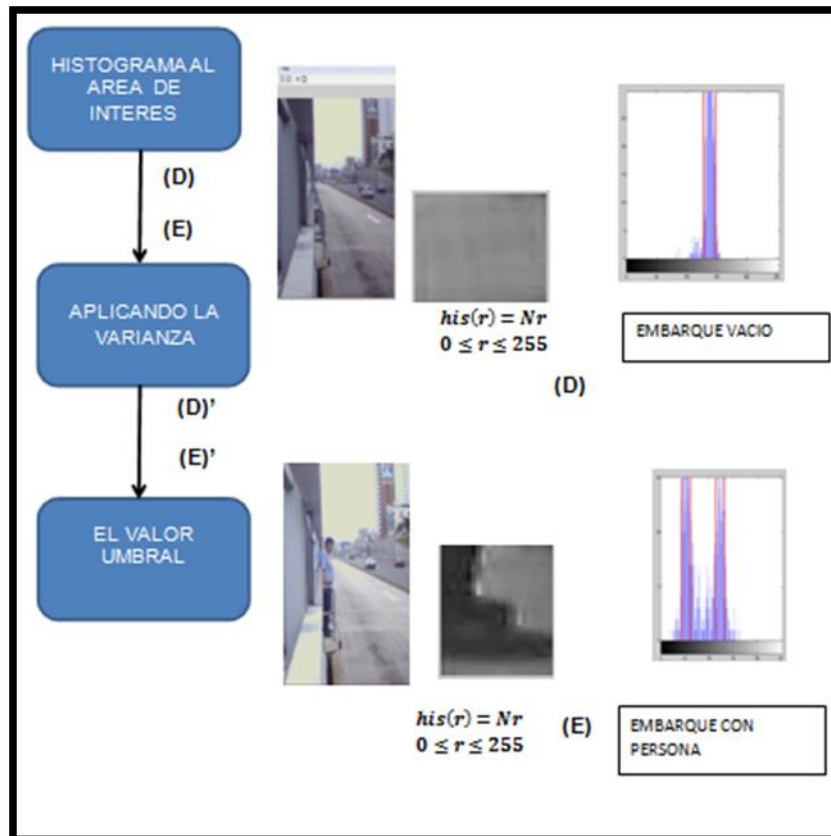


Figura 27- Diagrama de bloques.
Fuente: Propia.

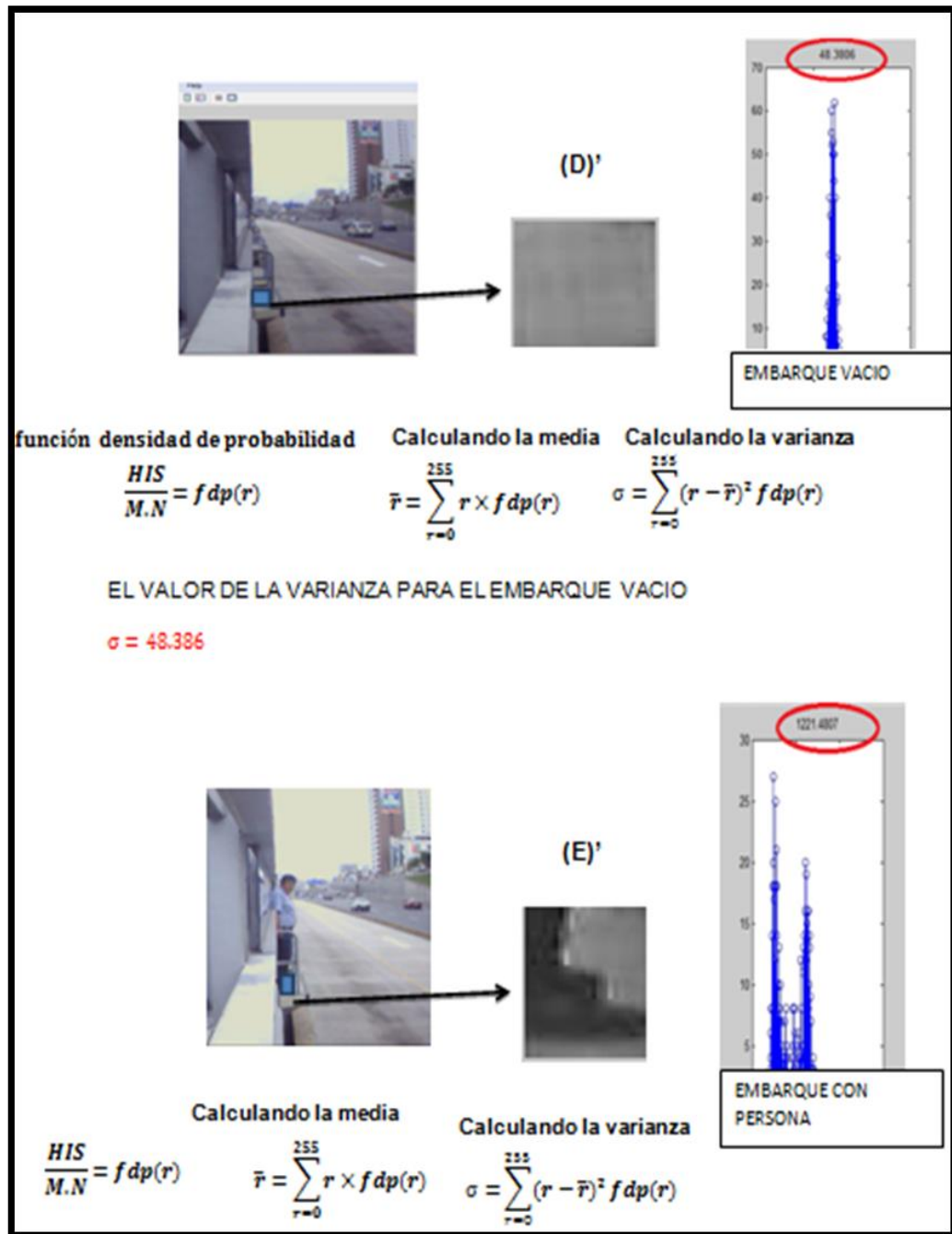


Figura 28- La varianza.

Fuente: Propia.

El valor de la varianza para el embarque con persona:

$\sigma = 1221.4807$

CAPÍTULO III

DESARROLLO DEL PROYECTO

3.1 Justificación

El 23% de los incidentes que ocurren en el Metropolitano implica a pasajeros presionados en las puertas de los buses, dato revelado por el gerente de Operaciones de Protransporte, Hernán Navarro.

Él explicó que estos problemas se deben porque los usuarios no abordan o descienden de las unidades de una manera adecuada.

A raíz de esto, La empresa Protransporte inició la campaña "Esté Atento", que buscó informar y concientizar a los ciudadanos para que respeten el aviso de cierre de puertas de los buses del sistema y eviten problemas. De tal forma que se concientice sobre la importancia de estar atentos a las luces rojas y al sonido que emite el bus a la hora de abrir y cerrar las puertas. Estos incidentes son informados al Centro de Control, de manera ineficiente, debido a que no existe personal monitoreando cada plataforma de embarque.

Es por ello que se vio viable la utilización de un sistema de monitoreo en tiempo real, que pueda alertar inmediatamente sobre algún incidente en las plataformas de embarque.

3.1.1 Justificación de Impacto tecnológico

Siendo la tecnología el conjunto de conocimientos técnicos, científicamente ordenados, que permiten diseñar y crear bienes y servicios que facilitan la adaptación al medio ambiente y satisfacer tanto las necesidades esenciales como los deseos de la humanidad.

Uno de los campos donde se desarrollo la tesis, se encuentra en el procesamiento digital de señales.

Actualmente el sistema de gestión de incidentes en las puertas de los paraderos del metropolitano, es manual y deficiente.

Debido a que no se controla en el momento que surgen estos incidentes, se utilizará la tecnología para reportar en tiempo real estos hechos relevantes.

Ayudando al sistema de metropolitano a contar con tecnología de punta para reportar estos casos y poder tener una respuesta inmediata a los incidentes.

Este impacto traerá como consecuencia un control y monitoreo de los incidentes en las plataformas de embarque en tiempo real.

3.1.2 Justificación de Impacto económico

Se ha realizado un estudio que ayudará a poder realizar un costo aproximado de la solución para implementar, que se consideraría como inversión, de los cuales básicamente contaría con lo siguiente:

- 01 cámara IP para cada embarque según líneas.
- Sistema DSP controlador.
- Sistema de comunicaciones.

3.1.3 Justificación de impacto ambiental

En este caso, siendo un sistema automatizado, se utilizará equipos electrónicos que no generaran:

Contaminación, que es la aparición en el medio ambiente de elementos perjudiciales para los organismos vivos en cantidad superior a la capacidad natural de reducción y absorción.

Agotamiento de los recursos energéticos, que es la utilización intensa de fuentes primarias de energía no renovables.

Razón por la cual no habrá un impacto ambiental que pueda poner en riesgo el medio ambiente.

3.1.4 Justificación de impacto social

El impacto social que produce el sistema, se debe a que teniendo un monitoreo en línea, se puede llevar un registro de todos los incidentes, y así poder con datos reales, realizar campañas de cuidado y seguridad, e ir cambiando la cultura a las personas, que siguen pensando que subir al metropolitano es como abordar una combi o cúster.

De esta forma, se tiene como objetivo minimizar los incidentes, y cambiar a las personas.

3.2 Diagrama de Flujo

A continuación se muestra el diagrama de flujo general de la solución:

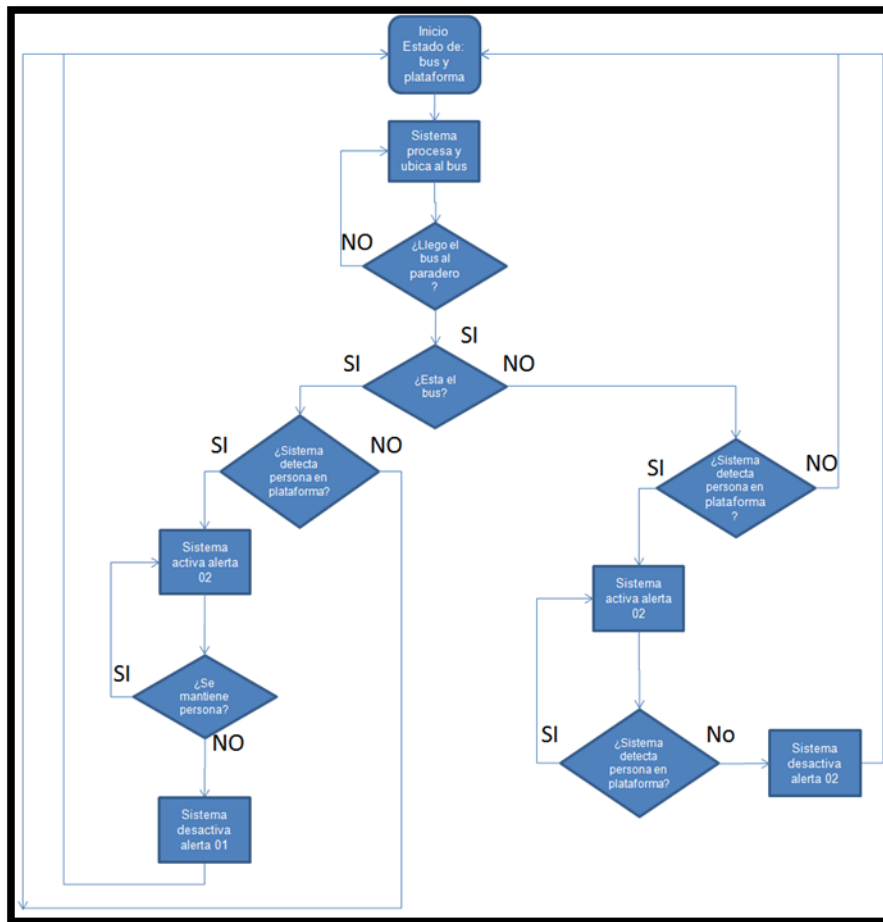


Figura 29- Diagrama de Flujo.

Fuente: Propia.

3.2.1 Diagrama de bloques general

El diagrama muestra los pasos del proceso del sistema:

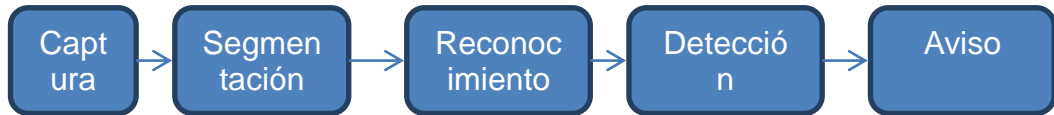


Figura 30- Diagrama de Bloques.

Fuente: Propia.

Técnicas utilizadas.



Captura

- Captura de la señal por cámara de video.
- Imagen digital.
- Digitalización de la señal.
- Imagen en color.
- Modelo RGB.

Reconocimiento

- Nivel de Luminosidad.
- Imagen en escala de grises.
- Histograma.
- Binarización.
- Varianza.

Segmentación

- Captura de primer área de interés.
- Captura de segunda área de interés.

Detección

- Umbral.

Alerta

- Aviso Sonoro

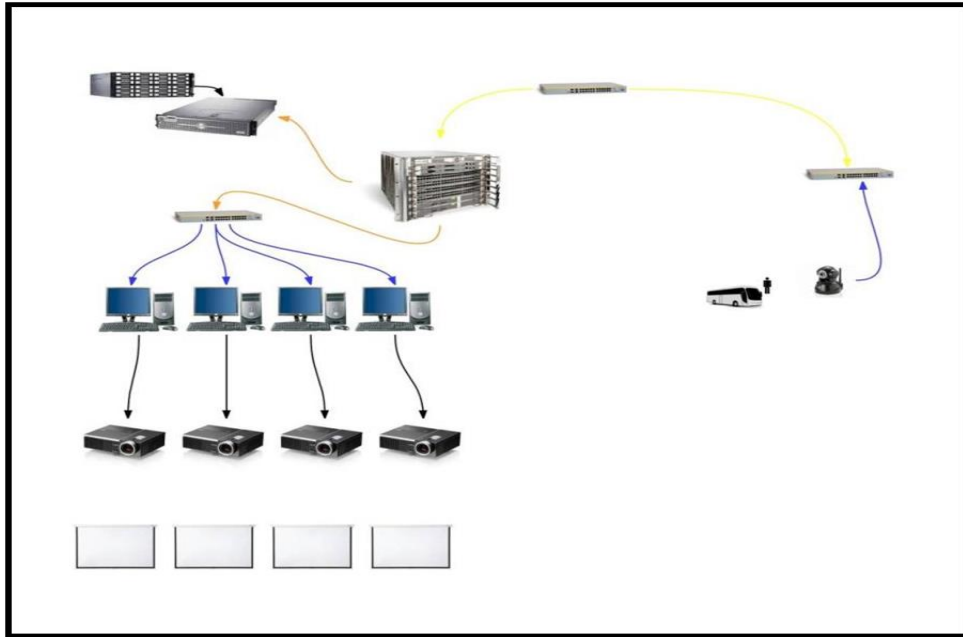


Figura 31- Diagrama pictográfico de la propuesta.
Fuente: Propia.

Componentes

Bus-persona-cámara IP-switch-router-switch core-servidor-storage-switch-PC-projector-ecran.

Alcance

El alcance de la solución simulada es lograr realizar la detección del bus y de las personas en el embarque. La red de Fibra se muestra en el anexo 10.

Luego en una segunda fase, podría converger en la red de comunicaciones del metropolitano, para poder así, tener en línea los sucesos que podrían pasar en estas áreas de las estaciones.

3.2 Programa en Matlab

En el anexo 07 se muestran los programas para la detección del bus 1, en el anexo 08 para de detección del bus 2, y en el anexo 09 para la detección de la persona en el embarque.

CAPÍTULO IV

PRUEBAS Y RESULTADOS

Conversión de una imagen a RGB

El proceso de captación de imagen, obtuvimos una imagen digital en el espacio de color RGB con sus tres matrices de intensidades para los colores rojo, verde y azul respectivamente. La figura 31 se muestran las 3 matrices correspondientes con intensidades para cada color de que van de 0 al 255.

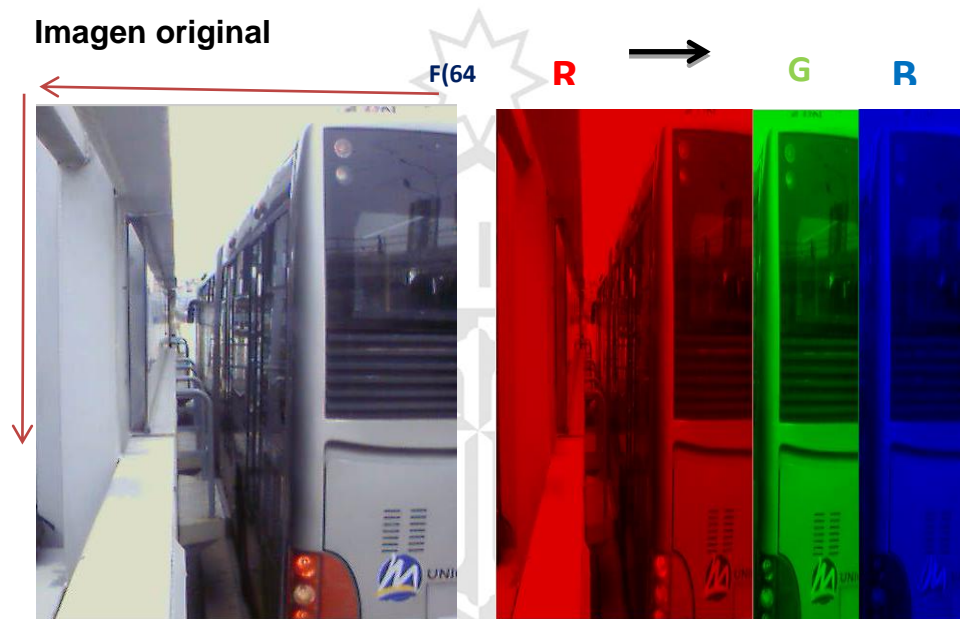


Figura 32- Imagen a color y con las 3 matrices.

Fuente: Propia.

Donde la imagen es polícroma.

$$I = (I,J,K) = f(I,J,K)$$

$$I=1,2,3,4,\dots,n$$

$$J=1,2,3,4,\dots,m$$

$$K=1,2,3, \text{ donde}$$

$$1 = \text{Rojo.}$$

2=Verde .

3=Azul.

4.1 Diseño de para la extracción del área de interés (llegada del bus)

Por posición de la cámara y estacionamiento del bus se realiza la extracción del lugar de interés del BUS.



Figura 33- Llegada del bus y extracciones del área de interés.
Fuente: Propia.

Donde se cumple que

$$M(U,V,S) = I(I,J,K)$$

Para

I=373.....,450.

J=480.....,570.

K=1,2,3

U=1.....,80

V=1.....,80

S=1,2,3

4.2 Diseño para la detección del faro de color rojo del bus. Valor umbral

Son un grupo de algoritmos cuya finalidad es segmentar gráficos, rasterizados, es decir separar una imagen que nos interesa del resto.

4.2.1 Identificación o reconocimiento por umbrales de color



Figura 34- Imagen a color a procesar.
Fuente: Propia.

34	110	147	144	158	152	150	153	152	152	153	153	153	153	153	153	153
35	115	143	142	152	149	147	151	149	150	151	152	152	152	152	153	153
55	102	149	141	148	146	145	149	148	150	151	152	152	152	152	151	151
59	100	153	142	145	145	146	149	148	151	150	152	152	151	151	151	151
54	107	152	142	143	144	147	148	147	151	149	151	151	150	151	151	151
53	110	152	141	143	145	148	147	147	151	149	150	150	150	150	149	149
58	110	153	141	144	146	147	147	147	150	149	149	149	150	150	149	149
59	111	151	138	147	148	146	147	148	149	149	148	148	148	148	148	148
62	111	147	137	149	148	144	147	148	149	149	148	148	148	148	148	148
63	106	145	136	149	145	147	147	149	145	147	148	148	148	148	148	148
54	117	127	147	141	147	149	141	146	143	149	147	147	147	148	148	147
101	87	158	132	156	137	137	147	147	148	144	143	143	146	147	147	147
78	113	136	136	145	145	147	149	150	148	148	141	142	145	146	146	146
93	102	150	134	144	149	147	143	139	139	141	140	141	144	145	145	146
69	91	143	130	147	139	134	141	138	140	138	140	140	143	143	144	144
23	84	74	121	101	122	127	130	140	140	144	139	139	141	142	143	144
77	30	93	66	70	90	97	107	116	126	131	139	139	138	139	141	143
110	81	95	73	66	66	65	65	72	84	102	122	131	135	135	139	140
115	108	110	103	97	92	84	75	66	62	67	78	90	109	140	140	120
118	124	122	117	114	112	109	99	83	70	59	55	56	83	75	120	164
123	128	131	124	123	119	121	116	106	96	82	76	64	43	61	85	117
131	140	137	134	130	123	121	121	119	116	106	102	92	51	83	64	43
134	146	141	138	136	132	131	128	127	125	117	110	101	108	72	68	68
135	145	147	146	145	149	145	140	136	130	125	120	114	125	93	89	91

Figura 35- Imagen a color en sus 3 matrices.
Fuente: Propia.

4.2.2 Discriminación del componente de la imagen con mayor tonalidad Roja

- Para todo $M(U,V,2)$ Mayor a 80 en tonos equivalente a (Verde Oscuro). Se le asigna valor 0.
- Para todo $M(U,V,3)$ Mayor a 80 en tonos equivalente a (Azul Oscuro). Se le asigna valor 0.
- Donde el número 80 es el Umbral de significancia de color.



Figura 36- Imagen discriminada.
Fuente: Propia.

34	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
35	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
55	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
53	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
58	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
59	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
62	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
63	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
54	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
69	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
23	84	74	0	0	0	0	0	0	0	0	0	0	0	0	0	0
77	30	0	66	70	0	0	0	0	0	0	0	0	0	0	0	0
0	81	95	73	66	66	65	65	72	0	0	0	0	0	0	0	0
0	108	0	103	97	92	84	75	66	62	67	78	0	0	0	0	0
118	124	0	0	0	0	0	99	83	70	59	55	56	0	75	0	0
123	128	131	124	123	119	121	0	106	96	82	76	64	43	61	0	0
131	140	137	134	130	123	121	121	119	0	106	102	92	51	83	64	43
134	146	141	138	136	132	131	128	127	125	117	110	101	0	72	68	68
135	145	147	146	145	149	145	140	136	130	125	120	114	0	93	89	91
138	146	151	147	149	156	152	146	145	142	145	0	0	123	0	106	93

Figura 37- Imagen discriminada de 0 a 255.

Fuente: Propia.

4.2.3 Finalmente tonalidad de color para todo M (U,V,1) Menor que 100 se le asigna valor 0



Figura 38- Imagen solo del color rojo.

Fuente: Propia.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	108	0	103	0	0	0	0	0	0	0	0	0	0	0	0	0
118	124	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
123	128	131	124	123	119	121	0	106	0	0	0	0	0	0	0	0
131	140	137	134	130	123	121	121	119	0	106	102	0	0	0	0	0
134	146	141	138	136	132	131	128	127	125	117	110	101	0	0	0	0
135	145	147	146	145	149	145	140	136	130	125	120	114	0	0	0	0

Figura 39- Imagen solo de 0 a 255.

Fuente: Propia.

4.2.4 Binarización de la imagen correspondiente al plano de color Rojo.

$$R(I, J) = \text{round}(M(U, V, 1) / 255)$$

Donde R es imagen binarizada.

0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0	1	1	0	0	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	0	0	0	0	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0	0
1	1	1	1	1	1	1	1	1	1	1	0	0	0	0	0	0

Figura 40- Imagen binarizada del color rojo.
Fuente: Propia.

4.2.5 Cálculo de la Sumatoria de Magnitudes de la Imagen o sumatoria de la masa

Mediante la siguiente formula obtenemos la sumatoria:

$$U = \sum_{i=1}^M \sum_{j=1}^N R(i,j)$$

U = Es un escalar.

Donde

M= 83, **N**=80.

U= 1927

4.2.6 Decisión por Umbral.

- Hay bus si U es mayor de 300.
- No hay bus si U es menor de 300.

4.3 Diseño para la extracción del área de interés para la detección de la persona (embarque de la estación)

4.3.1 La posición de la cámara se realiza la extracción del lugar que nos interesa del embarque

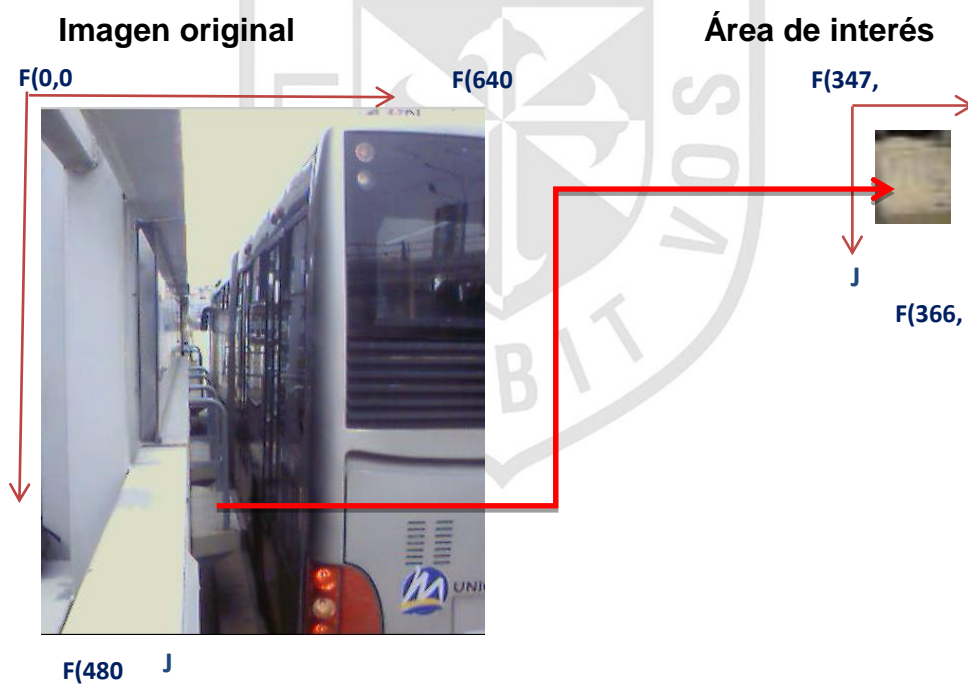


Figura 41- Extracción del área de interés.

Fuente: Propia.

4.3.2 Aplicación de luminancia a área de interés para la detección de persona.

Aplicación de luminancia por fórmula

$$Y = 0.299 * R + 0.587 * G + 0.114 * B$$

Imagen original
grises

imagen en RGB

imagen en escala de
grises

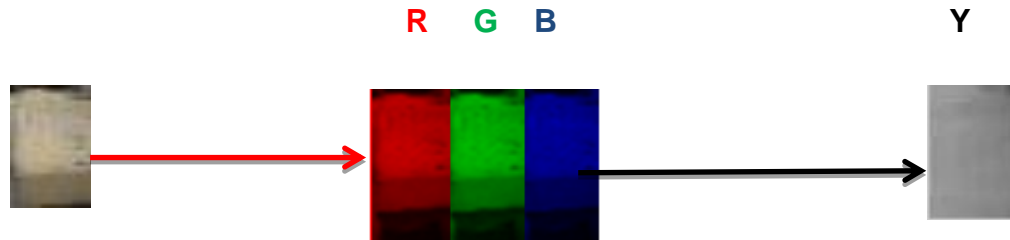


Figura 42- Imagen a escala de grises.

Fuente: Propia.

4.4 Diseño para la detección de personas usando histogramas, varianza y umbral en la zona de interés

4.4.1 Cálculo utilizando método estadístico del histograma

Los métodos basados en el histograma son muy eficientes en comparación con otros métodos de segmentación de la imagen, ya que normalmente requieren sólo una pasada por los píxeles. En esta técnica, un histograma se calcula a partir de todos los píxeles de la imagen, y los picos y valles en el histograma se utilizan para localizar los grupos en la imagen (el color o la intensidad pueden ser usados como medida).

$$his(r) = Nr$$

$$0 \leq r \leq 255$$

Dónde:

r: N° de pixel de la tonalidad (*r*)

Nr: cantidad de pixel de la tonalidad (*r*)

4.4.2 Cálculo utilizando método estadístico del histograma

Histograma con embarque vacío



Figura 43- Embarque vacío.
Fuente: Propia

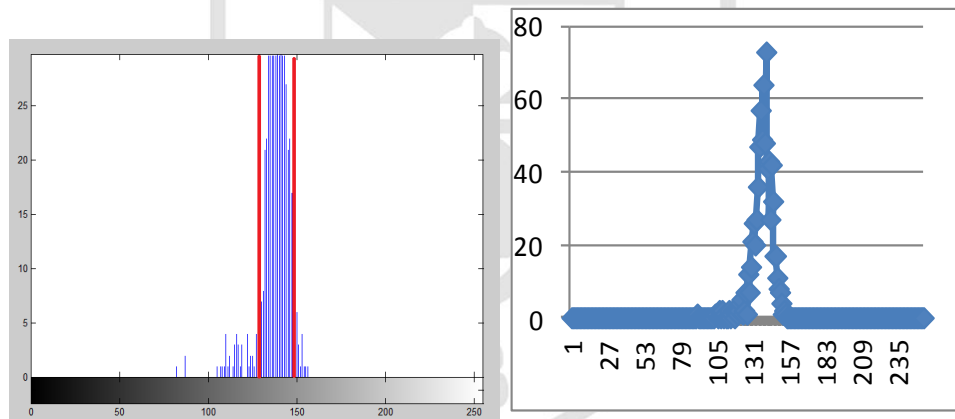


Figura 44- Histograma.
Fuente: Propia.

La imagen muestra la escala de grises de la imagen

145	146	151	148	150	152	151	148	145	145	146	148	141	135	131	121	111	109	114	117	107	91
148	149	154	148	150	152	151	148	145	145	146	145	138	134	133	128	120	119	123	132	124	109
150	151	155	147	149	151	150	147	144	144	145	144	138	135	137	136	130	128	131	141	137	122
149	150	154	145	147	149	148	145	142	142	143	146	139	136	139	139	133	130	132	138	138	126
145	145	149	143	145	147	146	143	140	140	141	143	135	133	138	140	134	130	131	131	136	128
141	141	144	141	144	145	144	141	139	139	140	136	129	128	135	139	136	132	132	126	135	131
145	146	146	141	144	147	147	143	140	139	140	140	137	136	136	138	139	137	135	137	140	130
145	146	146	142	145	148	148	144	141	140	141	141	139	137	137	139	140	138	136	139	141	130
145	146	146	144	147	149	149	146	143	142	142	143	141	139	139	141	141	139	137	139	142	132
145	146	146	145	148	151	150	147	144	143	143	144	142	140	140	141	142	139	137	136	142	134
145	146	146	145	148	151	150	147	144	143	143	145	143	140	140	141	141	139	136	134	141	135
145	146	146	144	147	149	149	146	143	142	142	145	142	140	139	140	140	137	135	132	140	133
145	146	146	142	145	148	148	144	141	140	141	144	141	139	138	139	138	135	133	129	138	132
145	146	146	141	144	147	147	143	140	139	140	143	140	138	137	137	137	134	131	126	136	132
144	139	147	145	141	142	146	145	137	134	137	138	140	137	132	133	139	137	131	127	135	137
144	138	146	144	140	141	145	144	136	133	136	137	139	136	131	133	138	137	131	128	135	136
144	138	146	142	138	139	143	142	135	132	135	136	138	135	131	132	138	137	131	129	134	136
143	138	146	140	136	137	142	141	134	132	134	136	138	135	131	133	139	138	132	131	134	136
142	138	147	139	136	137	142	141	135	132	135	137	139	136	132	134	140	140	134	134	134	136
142	137	146	140	136	138	143	142	136	134	137	138	140	138	134	137	143	143	137	136	134	135
142	136	144	141	137	139	144	144	138	136	139	141	143	141	137	139	146	146	140	137	133	135
142	136	143	142	138	140	145	145	139	137	140	142	144	142	138	141	148	148	142	138	133	135
144	138	145	145	142	141	141	143	143	141	139	143	142	141	141	141	141	142	143	136	136	138
145	139	147	145	142	141	141	143	143	141	139	141	140	139	138	138	139	140	141	135	135	137
147	141	149	145	142	141	141	143	143	141	139	139	139	138	137	137	137	138	139	136	135	136
148	143	152	145	142	141	141	143	143	141	139	139	139	138	137	137	138	139	139	140	138	136
148	144	153	145	142	141	141	143	143	141	139	141	140	139	139	139	139	140	141	142	140	137
147	143	152	145	142	141	141	143	143	141	139	141	141	140	139	139	140	141	141	141	139	137
146	142	150	145	142	141	141	143	143	141	139	140	139	138	137	137	138	139	140	135	134	135
146	141	149	145	142	141	141	143	143	141	139	137	137	136	135	135	136	137	137	130	130	132
134	142	143	142	139	137	138	140	140	138	136	131	134	136	136	134	132	133	134	130	128	128
133	141	141	141	139	137	137	139	140	138	135	130	133	135	135	132	130	131	132	136	132	129
133	140	139	140	138	136	137	138	139	137	135	129	131	133	133	130	128	128	129	133	128	122
132	138	137	140	137	135	136	138	138	136	134	128	130	132	131	128	125	125	126	126	121	117
132	137	134	139	136	135	135	137	137	135	133	129	131	132	130	126	123	123	123	119	116	114
131	135	131	138	136	134	134	136	136	134	132	130	132	133	131	126	123	122	122	107	105	104

Figura 45- A escala de grises.

Fuente: Propia.

Histograma con persona en el embarque



Detecta el lado de interés

Figura 46- Con persona en embarque.
Fuente: Propia.

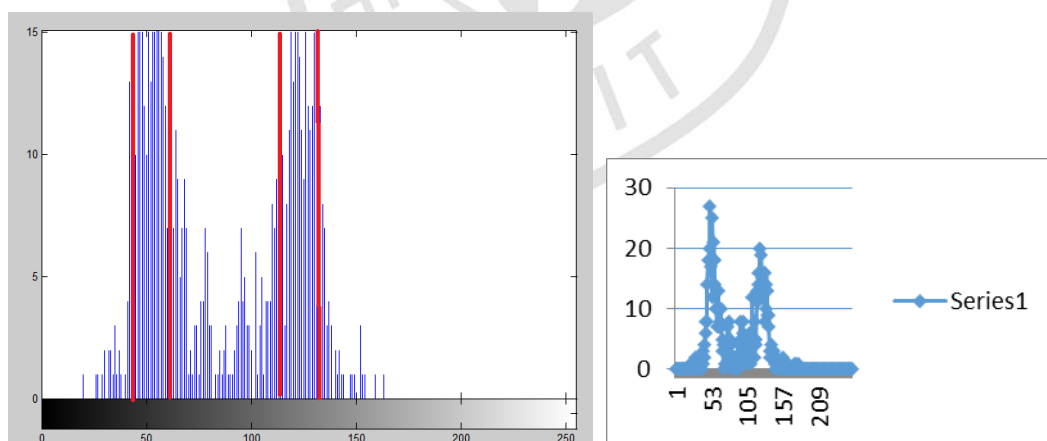


Figura 47- Histograma de área de interés.
Fuente: Propia.

47	48	49	48	47	43	27	70	122	161	141	147	134	126	126	118	104	101	108	105	99	91
48	48	50	44	42	42	21	77	140	135	139	145	136	131	131	124	112	108	112	113	106	98
48	49	50	43	43	44	28	88	155	126	142	139	134	132	132	127	120	117	118	119	113	106
49	49	51	45	50	47	48	98	163	131	142	132	131	130	129	127	124	123	123	122	118	113
49	49	51	45	55	45	64	98	157	135	133	129	129	127	125	124	125	125	125	120	117	115
49	50	51	41	55	39	71	93	146	134	121	128	129	126	122	121	124	125	124	117	116	115
45	56	51	44	50	40	91	91	123	135	125	132	129	127	126	127	126	124	121	123	120	116
45	51	46	51	49	31	82	96	129	142	133	133	130	128	127	128	127	125	122	124	121	117
45	47	43	53	44	23	74	106	133	145	138	135	132	130	129	130	129	127	124	126	122	119
46	46	46	47	42	28	79	123	133	140	138	136	134	131	131	131	131	128	126	127	123	120
48	47	53	42	47	46	97	150	135	135	139	137	134	132	131	132	131	129	126	127	123	120
49	46	56	43	52	58	107	172	138	132	141	136	134	131	130	131	131	128	126	126	122	119
49	41	52	47	48	49	97	177	137	129	139	135	132	130	129	130	130	127	124	124	121	117
49	37	47	48	40	33	79	173	133	124	134	134	132	129	128	129	129	126	124	123	120	116
47	50	49	48	46	51	65	86	104	113	115	125	122	120	129	128	128	122	122	119	120	120
48	49	47	49	47	50	60	74	83	85	83	101	101	109	125	133	132	126	125	120	122	122
49	49	47	48	47	49	54	61	63	59	54	68	70	90	113	134	130	127	124	121	123	124
49	50	50	47	46	46	50	54	55	53	50	47	46	74	98	131	121	124	119	120	122	123
48	51	54	48	46	44	45	48	51	52	53	49	40	71	91	133	116	124	118	118	120	121
49	52	56	51	49	46	44	44	46	48	50	61	44	78	93	143	117	129	122	119	120	119
50	51	55	53	53	53	51	49	48	48	49	63	43	82	95	152	119	133	123	122	122	120
52	50	53	53	56	59	60	58	55	53	53	57	37	80	94	155	118	132	121	125	124	122
56	56	59	61	63	65	66	64	59	53	49	47	54	68	90	111	125	127	125	118	123	117
57	57	59	62	64	66	67	65	60	55	52	51	56	67	80	93	100	101	101	102	111	112
58	57	59	62	64	66	66	65	61	57	54	53	58	65	70	72	74	77	78	80	92	97
57	55	56	60	61	63	63	62	59	56	54	53	58	63	64	61	61	65	70	65	71	73
56	53	54	56	57	58	58	58	56	54	53	52	56	60	58	54	53	57	62	60	56	51
57	54	53	54	54	55	55	55	54	53	53	53	54	55	53	50	48	49	51	59	52	45
61	57	56	54	54	54	55	55	55	55	55	52	51	50	51	53	52	50	47	56	57	55
64	60	58	55	55	55	55	56	56	57	57	51	49	48	53	59	61	57	52	52	62	68
66	71	61	68	68	67	65	64	62	61	61	62	63	65	66	65	63	61	59	46	53	66
77	85	77	80	79	79	77	76	75	74	74	74	75	76	77	77	77	76	76	75	79	88
93	103	97	97	97	96	96	95	94	94	94	93	93	93	93	94	95	97	98	97	98	102
103	115	110	111	111	111	111	111	111	111	111	113	112	110	108	108	110	111	113	106	106	107
106	119	113	118	118	119	119	120	121	122	122	127	125	122	119	117	117	117	118	116	114	112
105	117	111	119	120	121	122	124	125	126	127	132	130	127	124	121	119	118	118	118	114	110

Figura 48- Escala de grises.

Fuente: Propia.

4.4.3 Cálculo varianza por medio del histograma

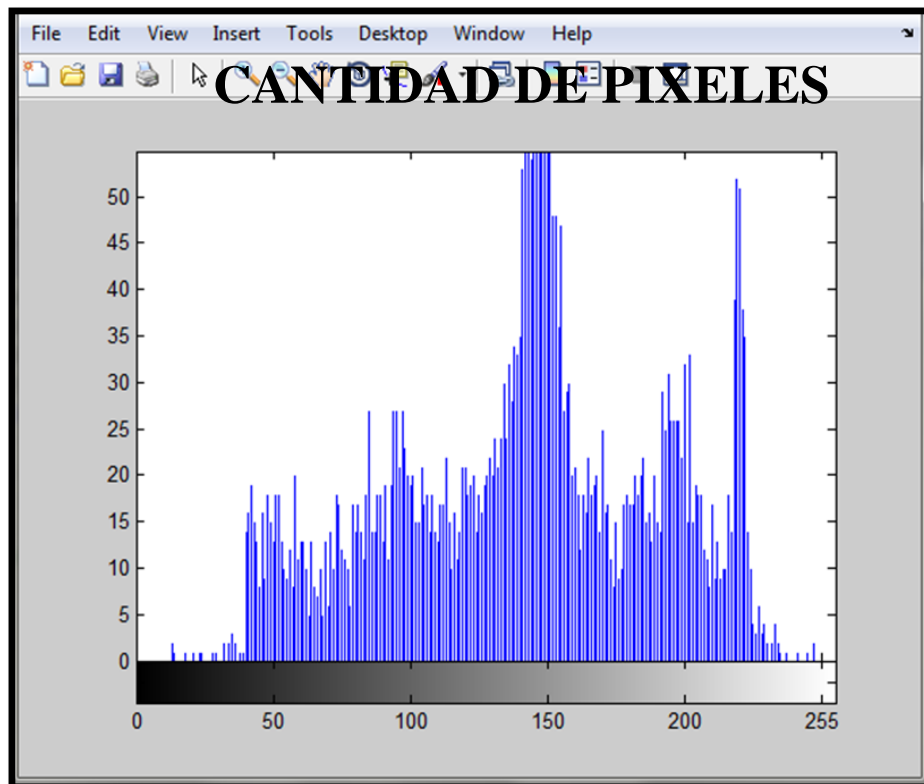


Figura 49- Números de tonalidades en escala de gris.

Fuente: Propia.

La ecuación para calcular el fdp es:

$$\frac{HIS}{M1XN1} = fdp(r)$$

$$HIS = \text{CANTIDAD DE PIXEL}$$

$$M1XN1 = \text{NÚMERO TOTAL DE PÍXELES}$$

$$fdp(r) = \text{función densidad de probabilidad}$$

Calculando la media:

$$r^- = \sum_{r=0}^{255} r \times fdp(r)$$

Calculando la varianza:

$$var = \sum_{r=0}^{255} (r - r^-)^2 f dp(r)$$

Aplicación en embarque vacío



Figura 50- Área de interés vacío.
Fuente: Propia.

El valor de la varianza es **48.3806**

Aplicación en embarque con persona

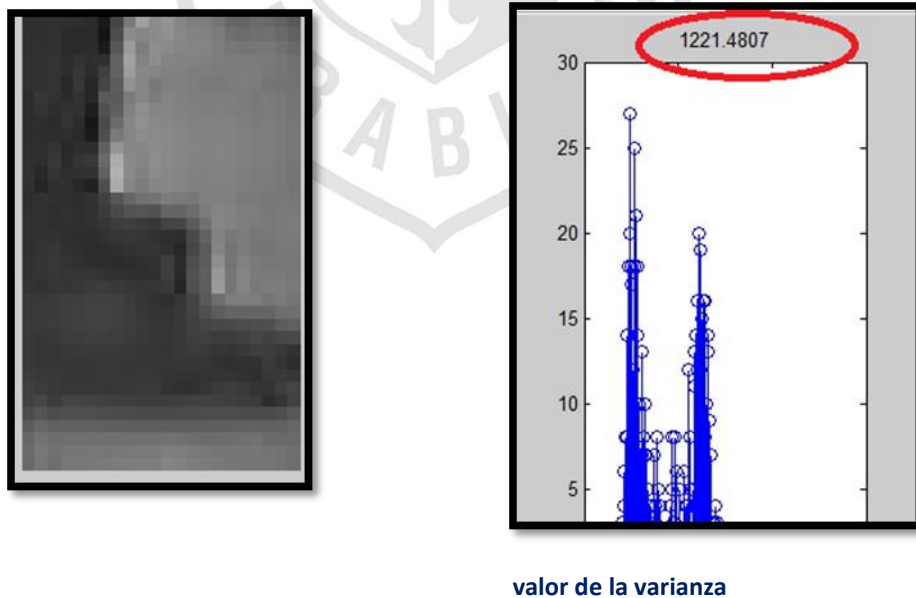


Figura 51- Área de interés con persona.
Fuente: Propia.

El valor de la varianza es **1221.4807**

Valor umbral de la varianza

Si VAR es $<$ que 300 no hay persona en el embarque.

Si VAR es $>$ que 300 hay persona en el embarque.

Pruebas tomadas y procesadas:



Figura 52- Imagen tomada.

Fuente: Propia.

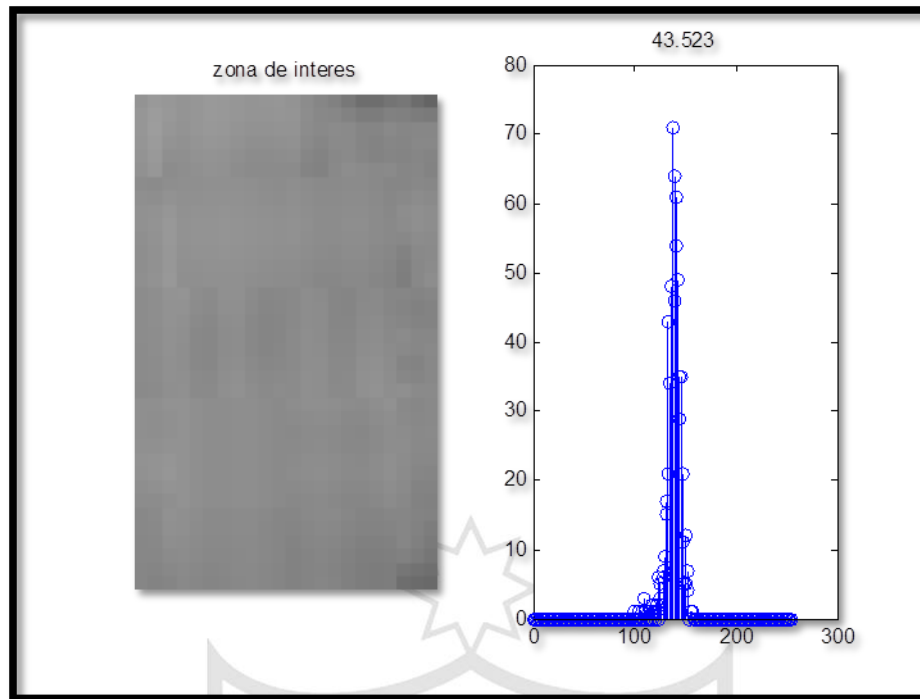


Figura 53- Imagen procesada.
Fuente: Propia.



Figura 54- Imagen tomada.
Fuente: Propia.

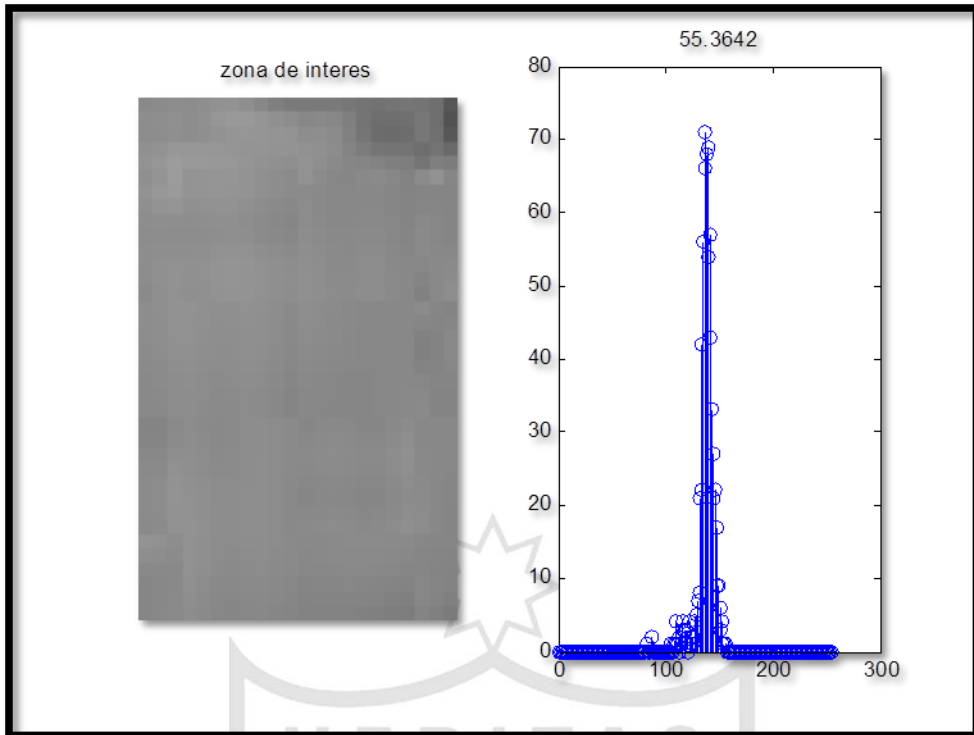


Figura 55- Imagen procesada.

Fuente: Propia.



Figura 56- Imagen tomada.

Fuente: Propia.

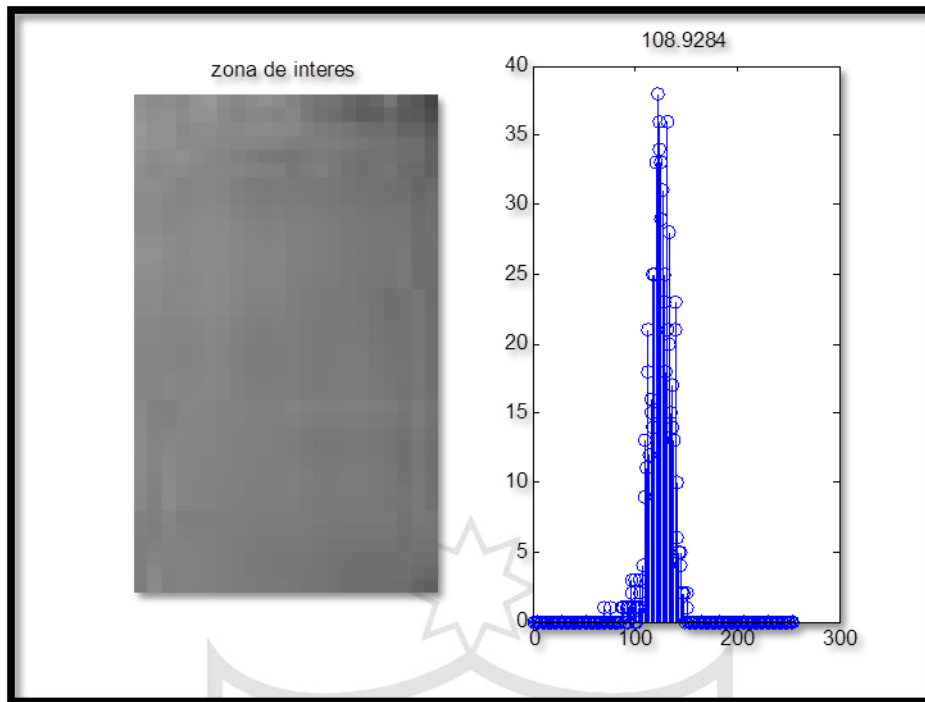


Figura 57- Imagen procesada.

Fuente: Propia.



Figura 58- Imagen tomada.

Fuente: Propia.

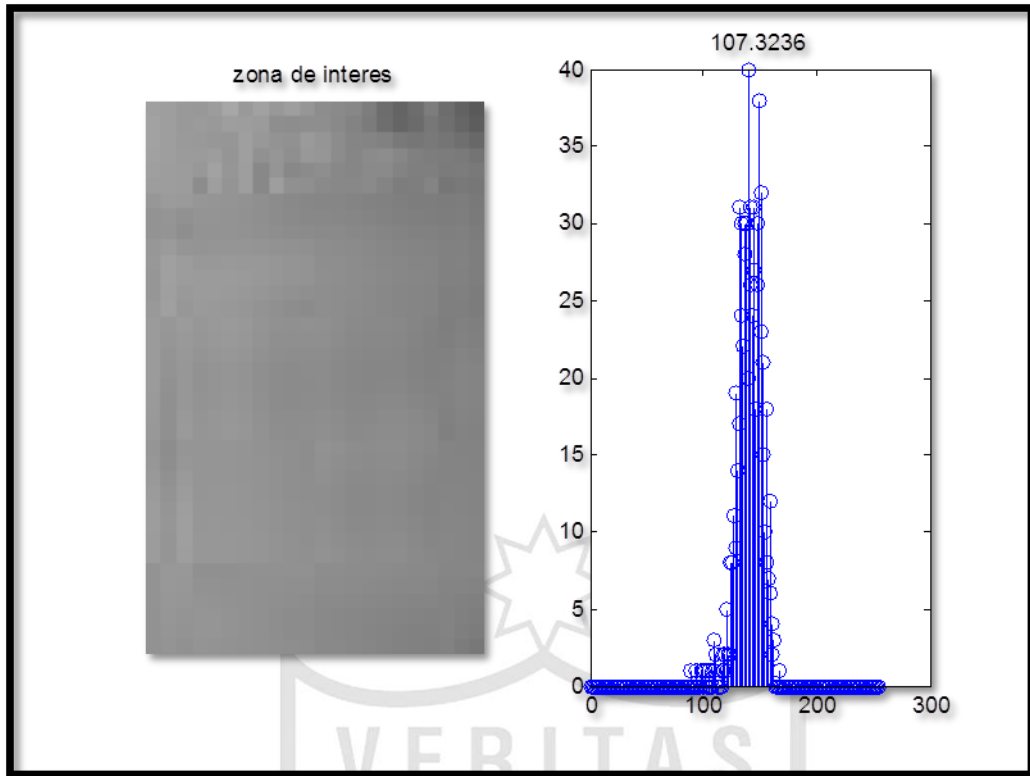


Figura 59- Imagen procesada.

Fuente: Propia.



Figura 60- Imagen tomada.

Fuente: Propia.

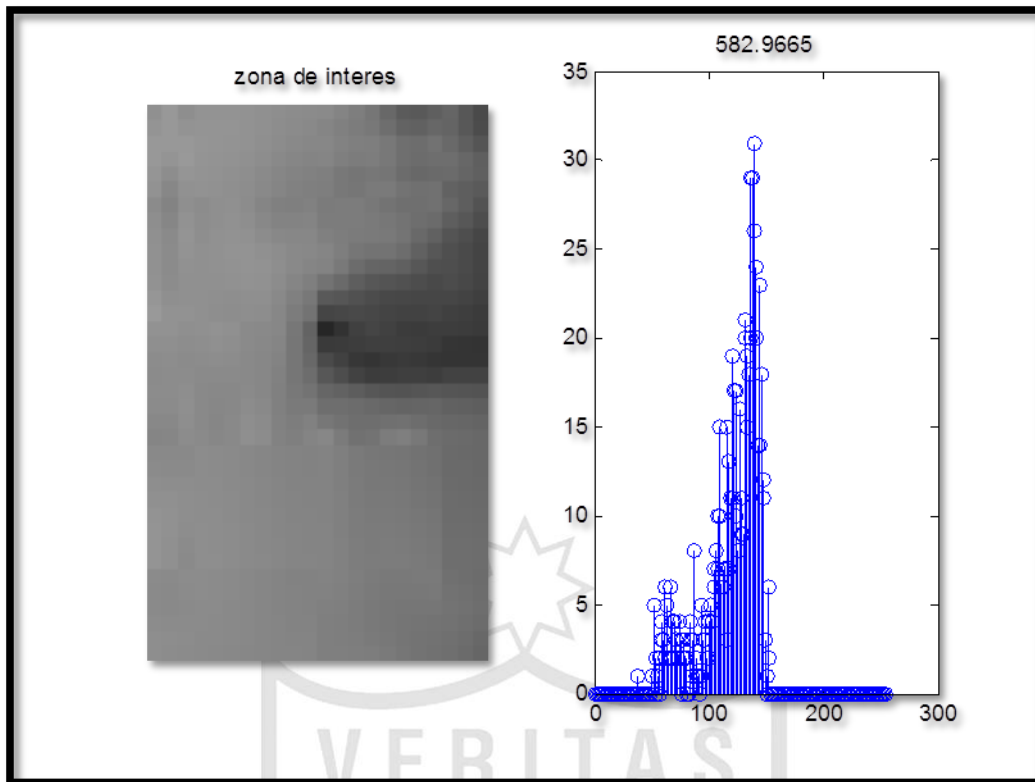


Figura 61- Imagen procesada.

Fuente: Propia.



Figura 62- Imagen procesada.

Fuente: Propia.

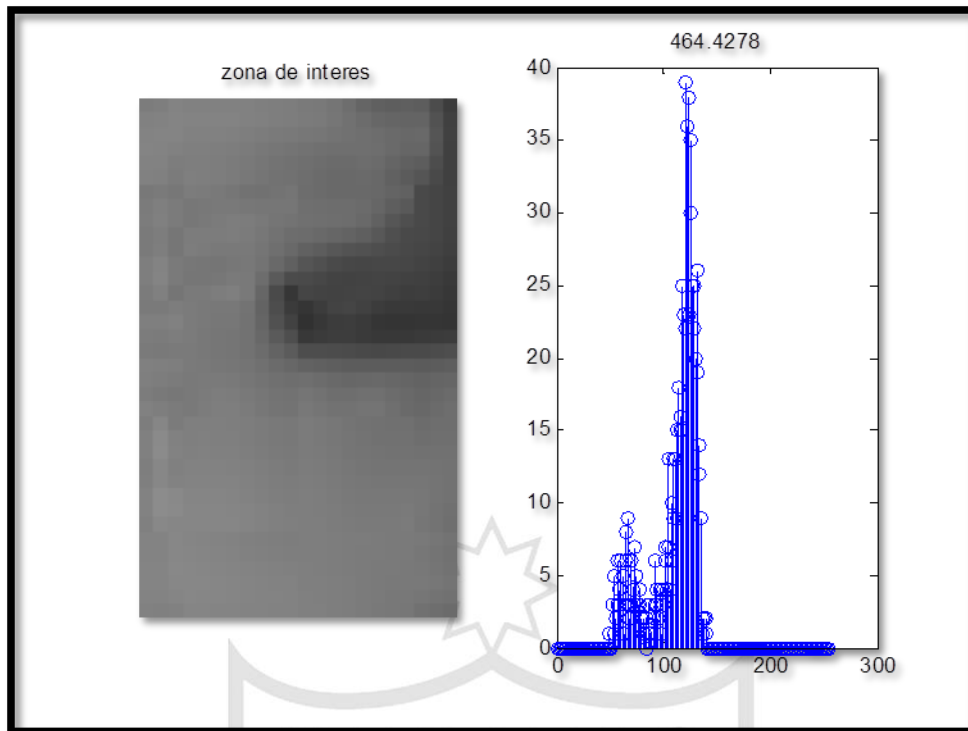


Figura 63- Imagen procesada.

Fuente: Propia.



Figura 64- Imagen tomada.

Fuente: Propia.

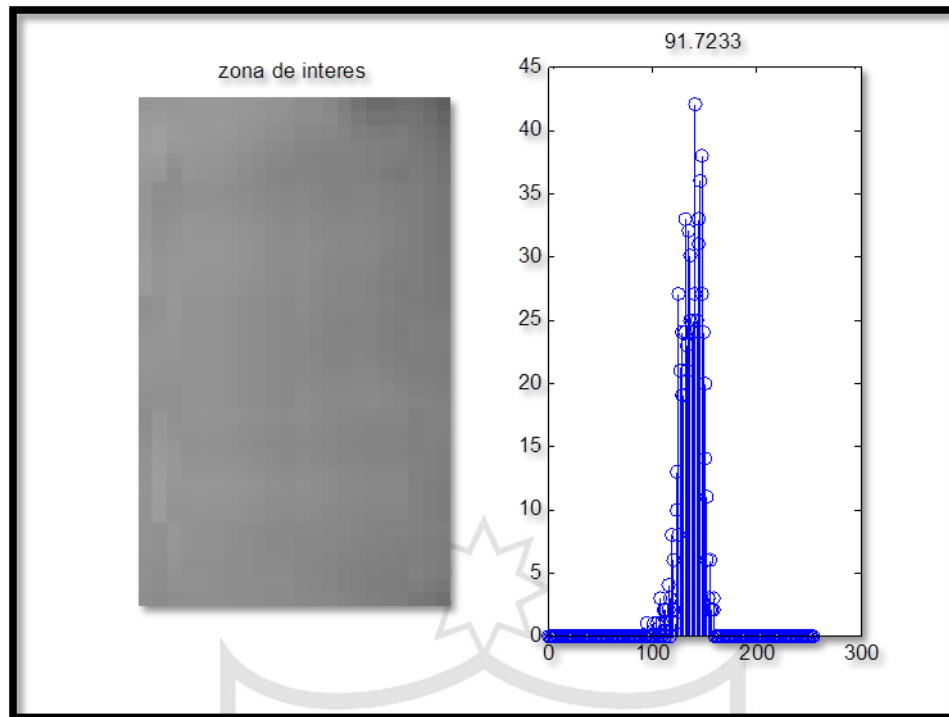


Figura 65- Imagen procesada.

Fuente: Propia.



Figura 66- Imagen tomada.

Fuente: Propia.

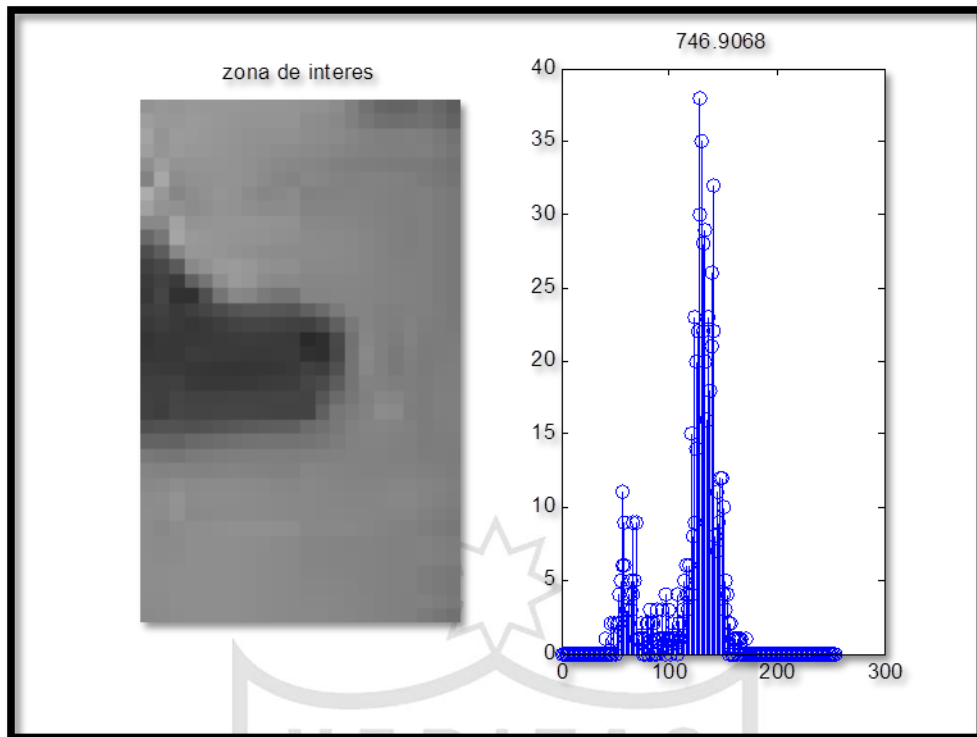


Figura 67- Imagen procesada.

Fuente: Propia.



Figura 68- Imagen tomada.

Fuente: Propia.

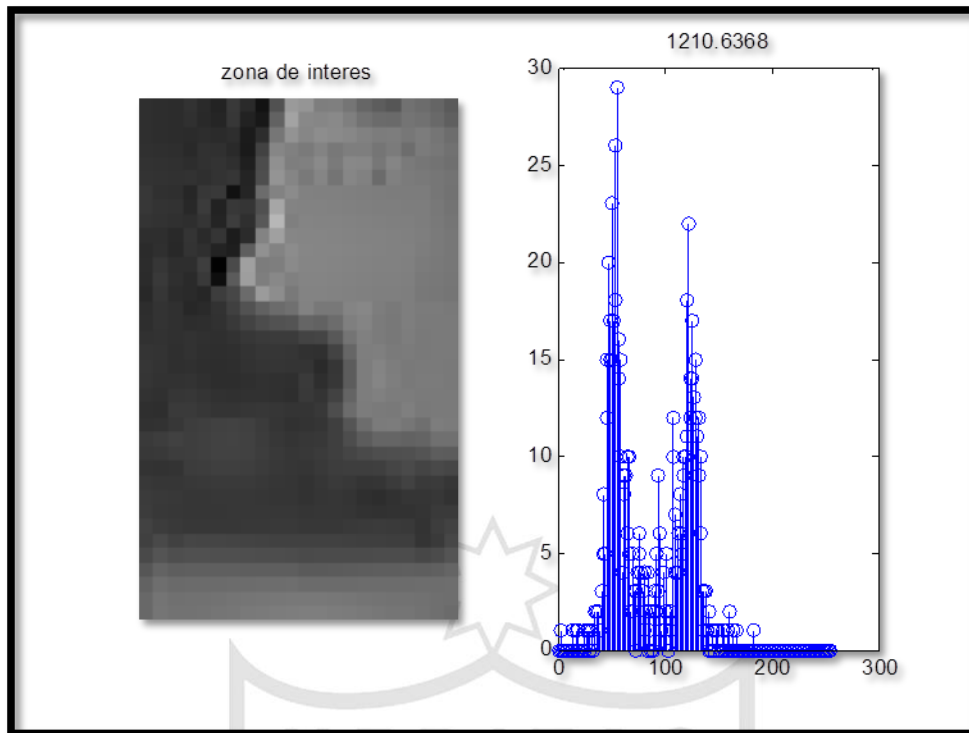


Figura 69- Imagen procesada.

Fuente: Propia.



Figura 70- Imagen tomada.

Fuente: Propia

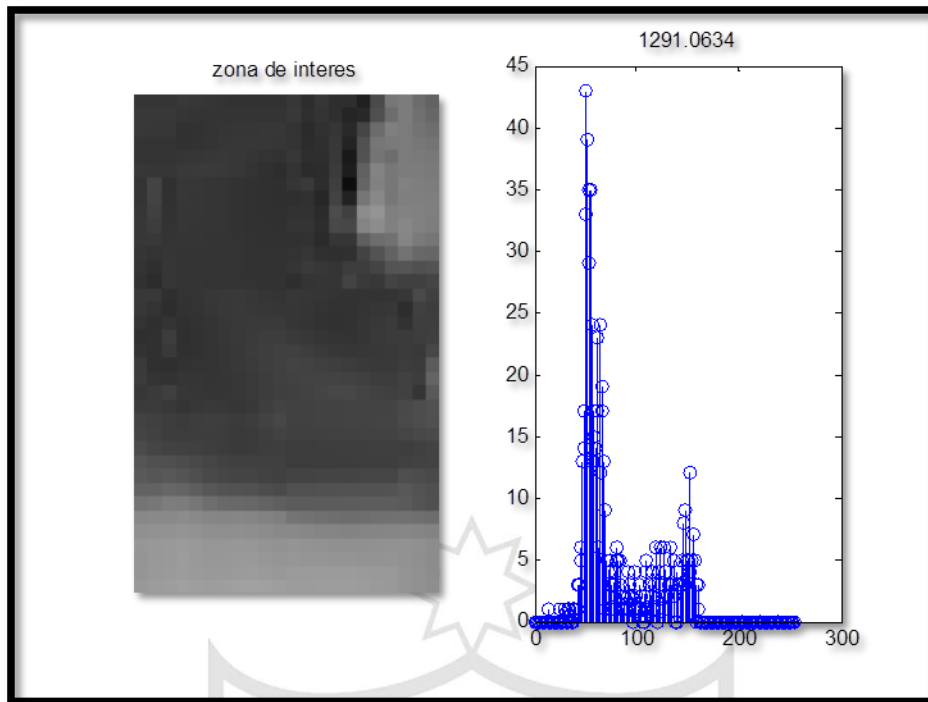


Figura 71- Imagen procesada.

Fuente: Propia.



Figura 72- Imagen tomada.

Fuente: Propia

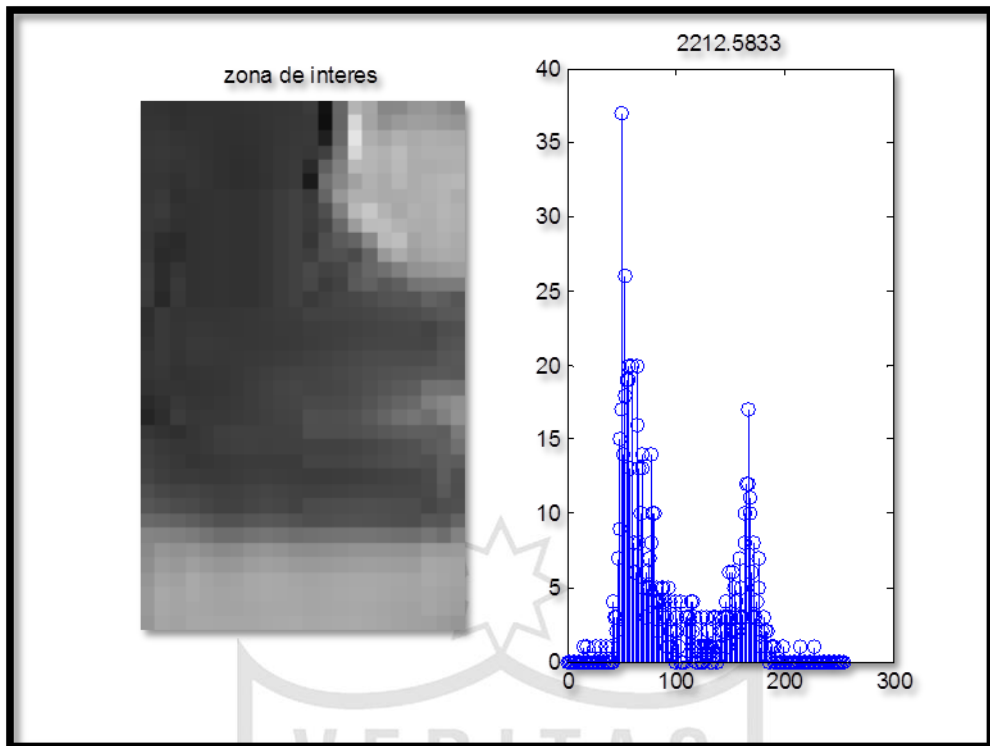


Figura 73- Imagen procesada.

Fuente: Propia.

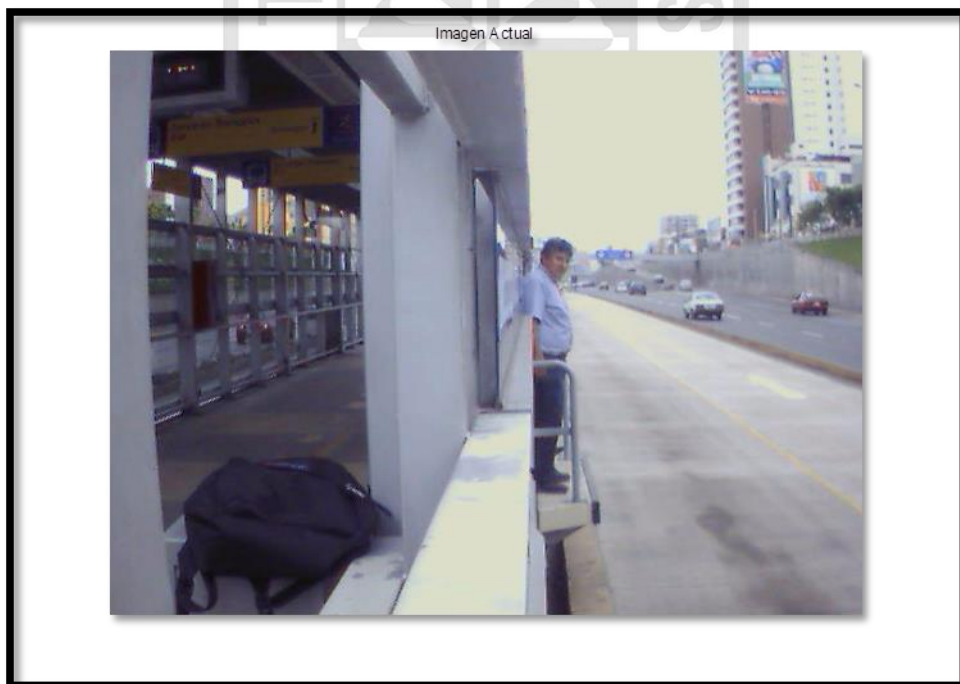


Figura 74- Imagen tomada.

Fuente: Propia

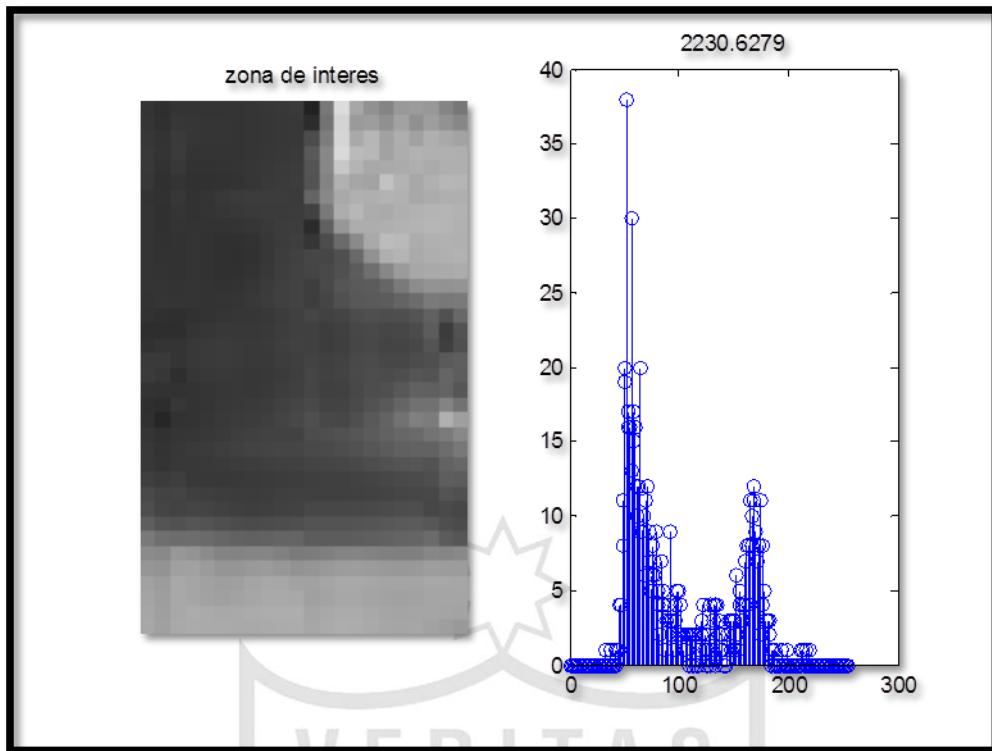


Figura 75- Imagen procesada.

Fuente: Propia.



Figura 76- Imagen tomada.

Fuente: Propia

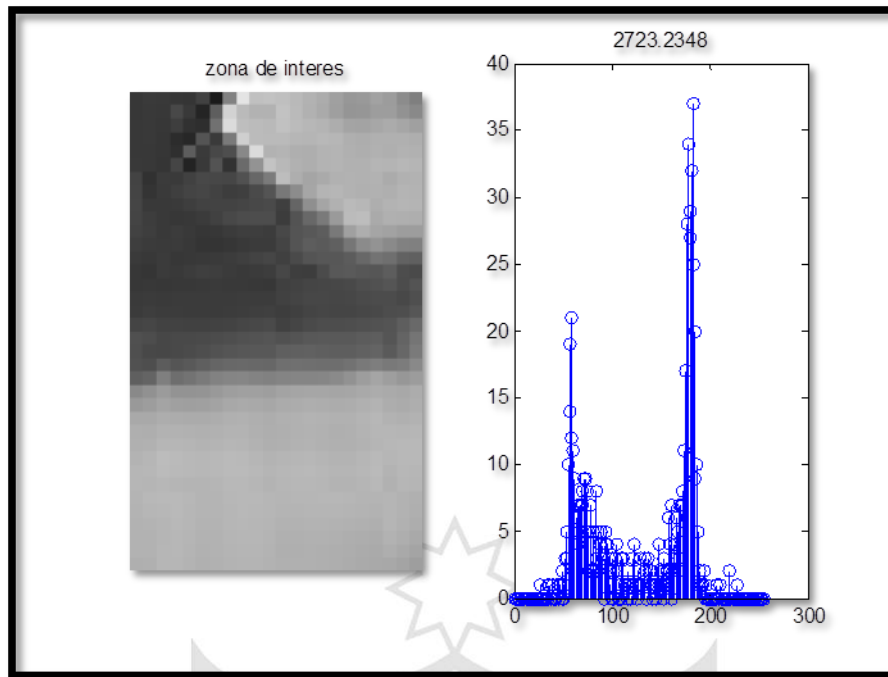


Figura 77- Imagen procesada.
Fuente: Propia.



Figura 78- Imagen tomada.
Fuente: Propia

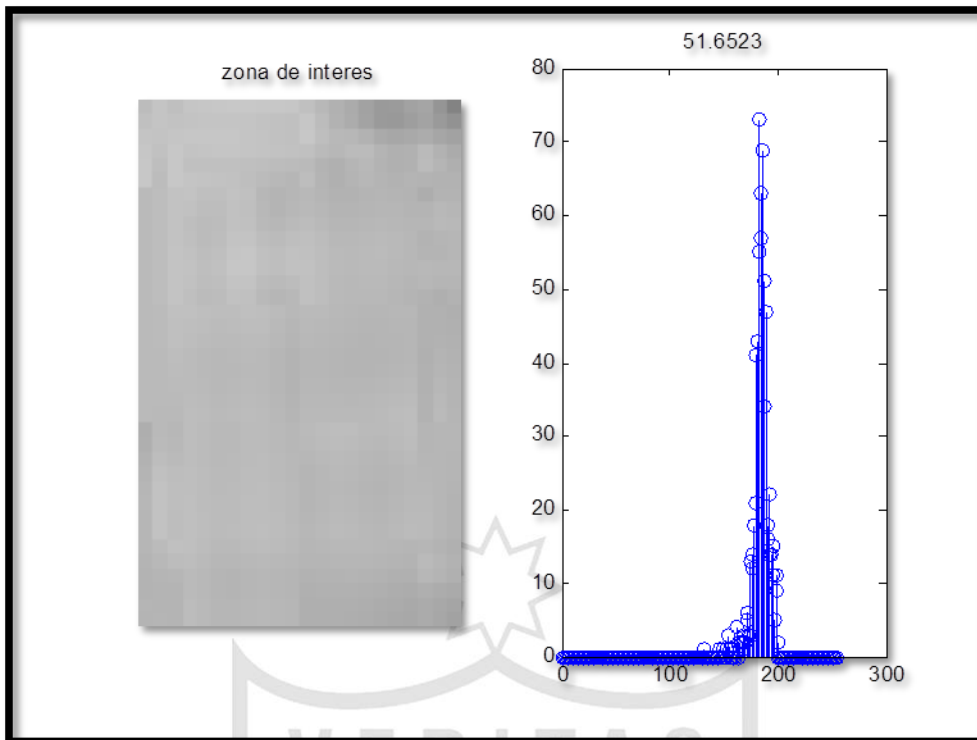


Figura 79- Imagen procesada.

Fuente: Propia.



Figura 80- Imagen tomada.

Fuente: Propia

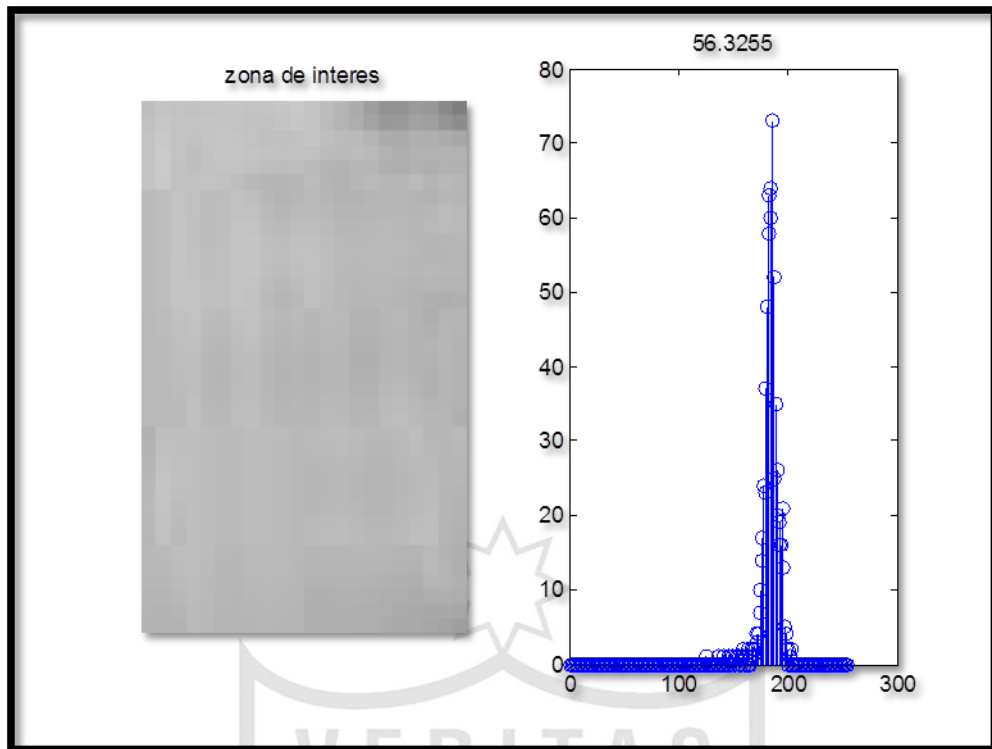


Figura 81- Imagen procesada.

Fuente: Propia.

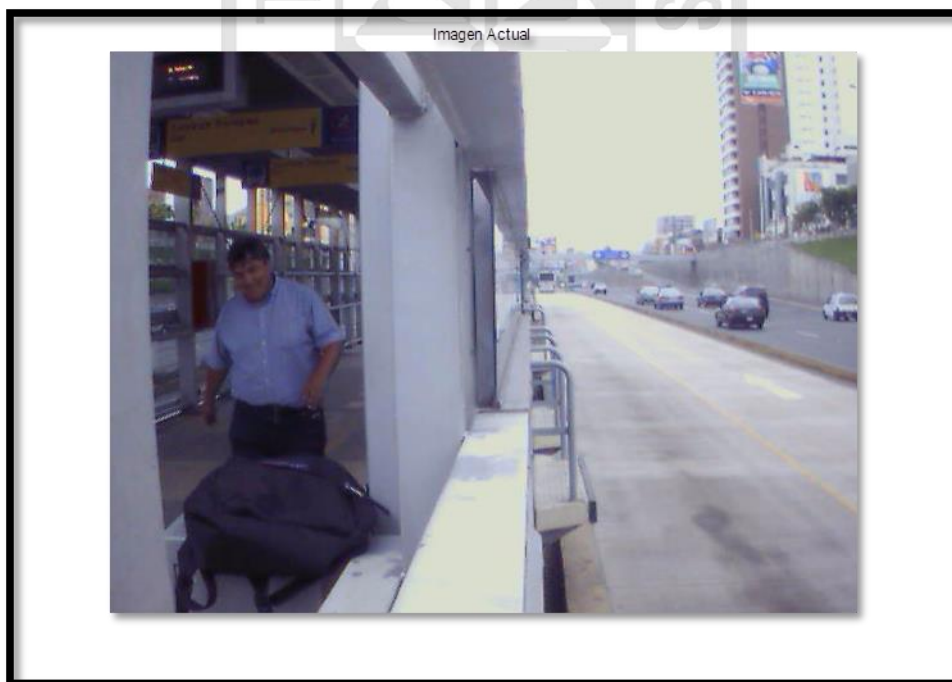


Figura 82- Imagen tomada.

Fuente: Propia

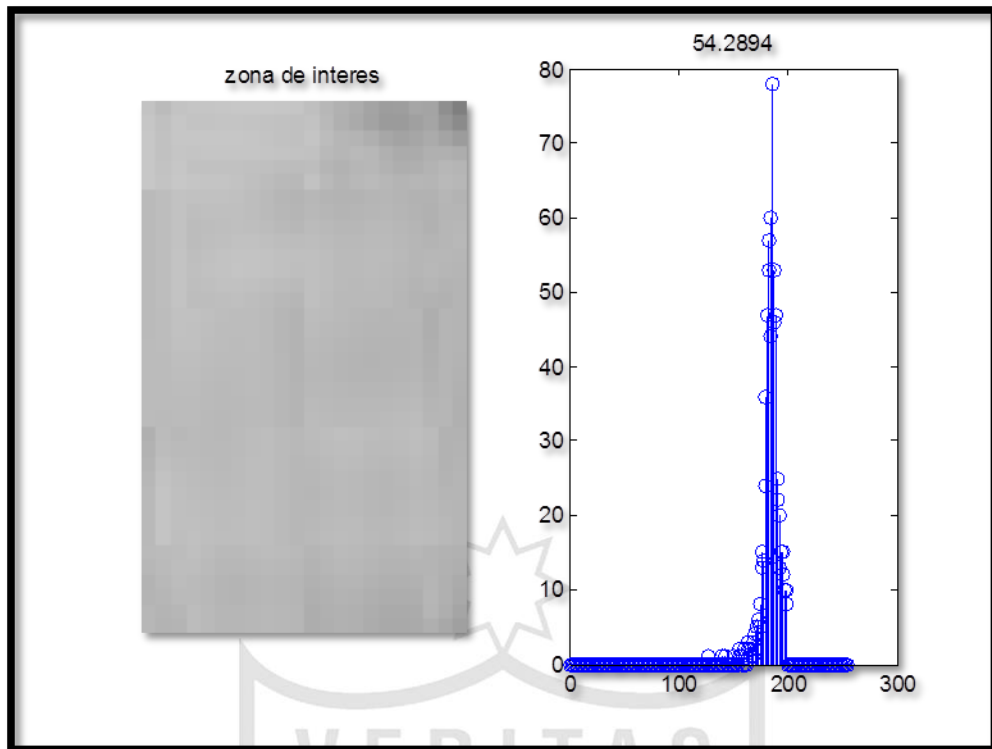


Figura 83- Imagen procesada.

Fuente: Propia.

Costos aproximados de la solución implementada:

Tabla N° 07: Listado de costos.

Concepto	Cantidad	Costo Unitario	Sub-costo total
Desarrollo del Sistema	640	\$20.00	\$12,800.00
Costos Varios	1	\$3,000.00	\$3,000.00
Sub total producto			\$15,800.00
Suministro de camaras, Instalacion y Configuracion de camaras	167	\$1,500.00	\$250,500.00
Servidor IBM System x3850 X5	2	\$35,000.00	\$70,000.00
Licencias de Milleston	167	\$900.00	\$150,300.00
Switch allied telesis at-8000s	2	\$3,195.00	\$6,390.00
PC ThinkCentre M93p SFF Desktop Core i5	10	\$656.00	\$6,560.00
Supervision durante implementacion	376	\$17.00	\$6,392.00
Sub total implementacion			\$490,142.00
Costo total de la solucion			\$505,942.00

Fuente: Propia.

Para un total de 38 estaciones. Se muestran en el anexo 11.

La cantidad de cámara por estación se muestra en el anexo 12.

Las características de la cámara IP se muestra en el anexo 13.

CAPÍTULO V

DISCUSIÓN Y APLICACIÓN

Luego de una lluvia de ideas grupal, y ver qué le faltaría al transporte para poder ayudar a mejorarlo, decidimos hacer la tesis en mención, y así ayudar a las personas.

Esta forma de sistema se puede aplicar a muchos otros sistemas, para detección de objetos, conteo de personas, etc.

Inicialmente se pensó usar para el reconocimiento de persona en el embarque, bordes de sobel o canny, pero realizando pruebas vimos que no necesitábamos esos algoritmos.

Pudimos hacer pruebas con restas de imágenes, y en algunos casos nos daban resultados positivos, no lo era para los demás, razón por la cual se busco otra solución, la varianza.

CONCLUSIONES

Primera: Mediante las simulaciones, se llegó al objetivo principal satisfactoriamente, si bien es cierto parte del alcance fue la corrida en horario diurno, en lo sucesivo se va a incrementar en otros horarios como de tarde, noche, o quizás con lluvia u otro cambio ambiental que pueda afectar el barrido

Segunda: El diseño matemático usado para poder llegar a la simulación, nos dio resultados positivos, en pocos casos fueron variados, pero se pudo ajustar de tal forma que se tuvo un sistema confiable.

Tercera: La cámara con la que se realizaron las pruebas, si bien fue limitante para tener una mejor resolución, fue suficiente para el desarrollo y pruebas del sistema.

RECOMENDACIONES

- Primera:** Como mejora futura debemos adecuar el sistema, para casos de tarde, noche o con otros fenómenos que pudiesen causar lecturas no válidas, esto debido al cambio de colores e intensidades en las imágenes.
- Segunda:** Hacer la implementación, en fases, ya que tiene un costo considerable, de tal forma que se puede romper la inercia con las estaciones principales, e ir creciendo con las medianas y bajas.
- Tercera:** Usar cámaras para la implementación de buena resolución, y que sean outdoor, de tal forma que tendremos un procesamiento más fino sobre las imágenes.
- Cuarta:** Usar el sistema propuesto en paralelo, y como mejora sacar provecho a la visión de la cámara, como por ejemplo detectar las placas de los buses, y así poder llevar el control de llegada y salida.
- Quinta:** Como peruanos se debería cambiar nuestra cultura, ya que por esta, corremos muchos riesgos de accidentes, en otros países desarrollados, parte de su cultura es el orden y la limpieza. Ejemplos que debemos seguir y así seguir creciendo como país.

GLOSARIO

BINARIZACION:	<i>Pasar colores a dos estados</i>
BIT:	<i>Mínima unidad digital, o cero o uno</i>
CCTV:	<i>Circuito cerrado de televisión</i>
DSP:	<i>Procesador Digital de Señales</i>
ESTEREO:	<i>Reproducción de dos señales</i>
HISTOGRAMA:	<i>Representación grafica de una variable</i>
IP:	<i>Protocolo de Internet</i>
MONOCULAR:	<i>Cyclic Prefix OFDM</i>
PIXEL:	<i>Mínima unidad de color en una pantalla.</i>
RGB:	<i>Red, Green, Blue.</i>
SENSOR:	<i>Hardware con entrada de diferentes tipos.</i>
TRAKING:	<i>Barrido de video con una cámara</i>
VARIANZA:	<i>La esperanza del cuadrado de la desviación de dicha variable respecto a su media</i>
VX:	<i>Vector en coordenada X.</i>
VY:	<i>Vector en coordenada Y.</i>
X:	<i>Coordenada X del plano cartesiano.</i>
Y:	<i>Coordenada Y del plano cartesiano.</i>

FUENTES DE INFORMACIÓN

Área de Desarrollo. (2013). El metropolitano.

Erwin Kreyszing. (1981). *Introducción a la estadística matemática*.

<http://www.metropolitano.com.pe/>

José E. Briceño. (2012). *Principios de las comunicaciones*.

Juan Vignolo Barchiesi. (2008). *Introducción al procesamiento digital de señales*. Valparaíso. Chile

Martine Joly. (2005). *Introducción al análisis de la imagen*.

Rafael C. Gonzales. (2007). *Digital Image Processing Third Edition*.

Steven L. Eddins. (2004). *Digital Image Processing using Matlab*

William Mendehall. (1989). *Introducción a la probabilidad y la estadística*.

W. D. Stanley. (1975). *Digital signal processing*.



ANEXOS

ANEXO 01

PROGRAMA PARA LA TOMA DE IMÁGENES UTILIZANDO CÁMARA IP Y MATLAB

(Pruebas)

```
url = 'http://192.168.1.42/snapshot.htm';
```

```
ss = imread(url);
```

```
FileNum = 1;
```

```
fh = image(ss);
```

```
while(1)
```

```
pause(1)
```

```
ss = imread(url);
```

```
set(fh,'CData',ss);
```

```
drawnow;
```

```
FileNum = FileNum + 1;
```

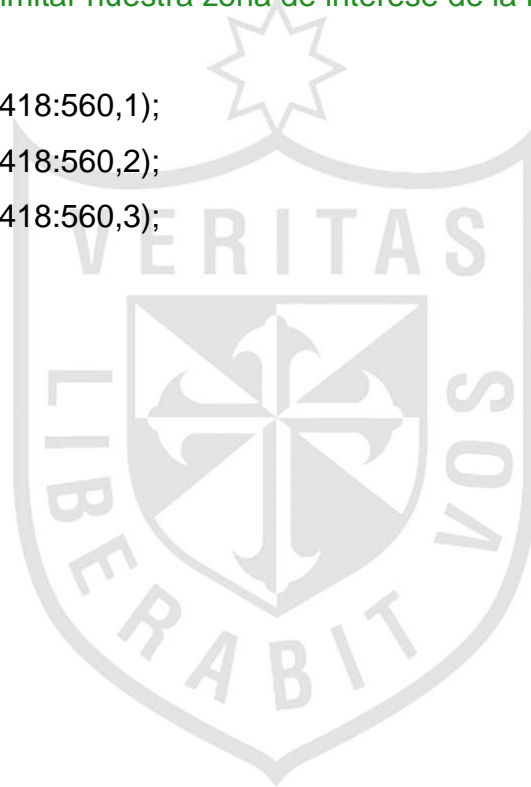
```
End
```



ANEXO 02

PROGRAMA PARA ELEGIR LA PRIMERA ÁREA DE INTERÉS

```
%segmenta zona de interes
clear all;
close all;
clc;
I=imread('701.JPG');
figure(1);imshow(I)
%Necesitamos delimitar nuestra zona de interese de la imagen.
M=[];
M(:, :, 1)=I(392:476,418:560,1);
M(:, :, 2)=I(392:476,418:560,2);
M(:, :, 3)=I(392:476,418:560,3);
M=uint8(M);
figure(2);
imshow(M)
impixelinfo
```



ANEXO 03

PRIMER PROCESAMIENTO DE ÁREA DE INTERÉS

```
[f,c,d]=size(M);  
M1=M;  
for m=1:f  
    for n=1:c  
        if M(m,n,2)>80 && M(m,n,3)>80  
            M1(m,n,1)=0;  
            M1(m,n,2)=0;  
            M1(m,n,3)=0;  
        end  
    end  
end  
M1=uint8(M1);  
figure(3);  
imshow(M1)  
impxelinfo
```



ANEXO 04

SEGUNDO PROCESAMIENTO DE ÁREA DE INTERÉS

```
M2=M1;
for m=1:f
    for n=1:c
        if M1(m,n,1)<100
            M2(m,n,1)=0;
            M2(m,n,2)=0;
            M2(m,n,3)=0;
        end
    end
end
M2=uint8(M2);
figure(4);
imshow(M2)
impixelinfo
```



ANEXO 05

DETECCIÓN DEL BUS

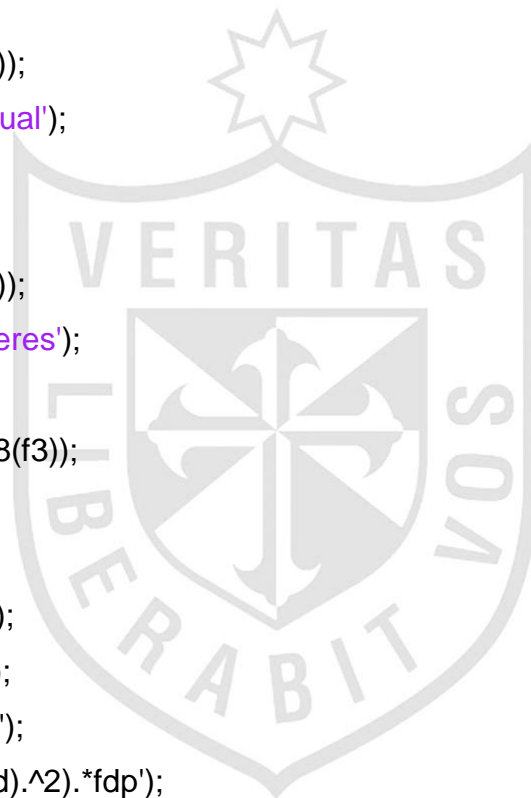
```
M2=uint8(M2);  
figure(4);  
imshow(M2)  
impixelinfo  
M3=round(M2(:,:,1)/255);  
R1=sum(sum(M3))  
if R1>300  
    disp('Llego bus metropolitano');  
end  
if R1<300  
    disp('No hay bus metropolitano');  
end
```



ANEXO 06

COMANDOS PARA HALLAR LA VARIANZA DEL ÁREA DE INTERÉS

```
r=0:255;
for i=1:50:Nf
    s1=VideoReader(stre1);
    f1=read(s1,i);
    f2=rgb2gray(f1);
    f3=f2(347:382,366:387);
    figure(1)
    imshow(uint8(f1));
    title('Imagen Actual');
    figure(2)
    subplot(1,2,1)
    imshow(uint8(f3));
    title('zona de interes');
    subplot(1,2,2)
    [his]=imhist(uint8(f3));
    stem(r,his)
    his=double(his);
    [M1,N1]=size(f3);
    fdp=his/(M1*N1);
    med=sum(r.*fdp');
    var=sum(((r-med).^2).*fdp');
    title(num2str(var));
    %pause
    if var>450;
        disp('Persona Presente');
        [y, Fs]=wavread('alarma.wav')
        sound(y,Fs)
    end
end
```



ANEXO 07

PARA DETECCIÓN DEL BUS

```
function [llego]=deteccionbus(l)
llego=0;
% %segmenta zona de interes y reconoce el rojo, está o no el metropolitano.
%
% clear all;
% close all;
% clc;
%Necesitamos delimitar nuestra zona de interese de la imagen.
M=[];
M(:,1)=l(373:450,480:570,1);
M(:,2)=l(373:450,480:570,2);
M(:,3)=l(373:450,480:570,3);
M=uint8(M);
% figure(2);
% imshow(M)
% impixelinfo
[f,c,d]=size(M);
M1=M;
for m=1:f
    for n=1:c
        if M(m,n,2)>80 && M(m,n,3)>80
            M1(m,n,1)=0;
            M1(m,n,2)=0;
            M1(m,n,3)=0;
        end
    end
end
M1=uint8(M1);
% figure(3);
% imshow(M1)
% impixelinfo
```

```

M2=M1;
for m=1:f
    for n=1:c
        if M1(m,n,1)<100
            M2(m,n,1)=0;
            M2(m,n,2)=0;
            M2(m,n,3)=0;
        end
    end
end
end
M2=uint8(M2);
% figure(4);
% imshow(M2)
% impixelinfo
M3=round(M2(:,:,1)/255);
R1=sum(sum(M3))
if R1>300
    llego=1;
    disp('Llego bus metropolitano');
end
if R1<300
    llego=0;
    % disp('No hay bus metropolitano');
end

```



ANEXO 08

PARA DETECCIÓN DE BUS

```
close all;
clear all;
clc;

%% Lee video de archivo
videoFReader = vision.VideoFileReader('video5.avi');
videoPlayer = vision.VideoPlayer; % Abre la ventana de visualización de
video
cont=0;
tic
Plantilla=[];
tag1=1;
tag2=0;
E1=[];
while ~isDone(videoFReader)
    videoFrame = step(videoFReader);
    A=uint8(videoFrame*255);
    pause(0.02);
    step(videoPlayer, videoFrame);
    % disp(cont)
    cont=cont+1;
    if mod(cont,25)==0
%     disp('un segundo de video');
%     break
    if deteccionbus(A) && tag1
        Plantilla = step(videoFReader);
        B1=Plantilla(213:387,355:377,1);
        B2=Plantilla(213:387,355:377,2);
        B3=Plantilla(213:387,355:377,3);
        %% Extraccion zona interes
        B=0.299*B1+0.587*B2+0.114*B3; % Luminancia plantilla
```

```

B4=B;
B4=uint8(B4*255);
figure;
imshow(uint8(Plantilla*255));
impixelinfo;
%% Histograma
figure;
imshow(B4);
impixelinfo;
figure;
imhist(B4)
tag1=0;
tag2=1;
end
if tag2
lactual = step(videoFReader);
C1=lactual(213:387,355:377,1);
C2=lactual(213:387,355:377,2);
C3=lactual(213:387,355:377,3);
C=0.299*C1+0.587*C2+0.114*C3; % Luminancia plantilla
%% Histograma de detección
C4=C;
C4=uint8(C4*255);
figure;
imshow(uint8(lactual*255));
impixelinfo;
figure;
imhist(C4);
[histograma]=histograma2(C4);
figure;
plot(histograma);
grid on;

%%%%%%%%%

```

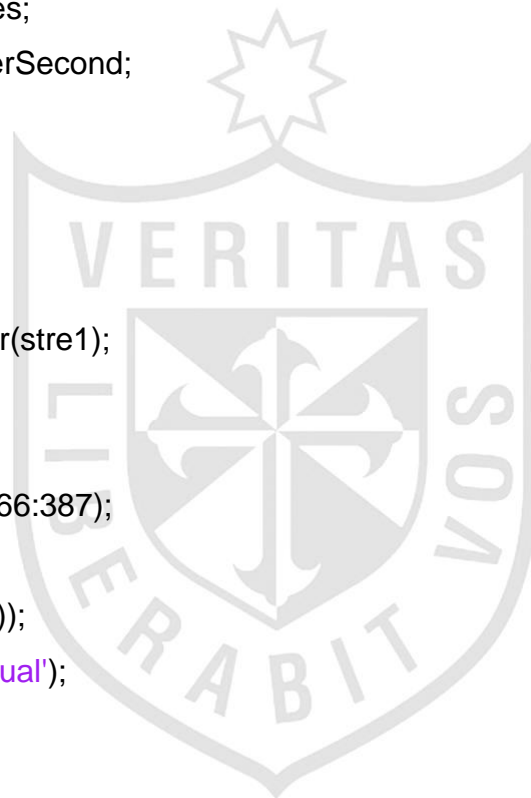
```
E=B-C;  
e=sum(sum(E.^2))/4025  
E1=[E1 e];  
end  
end  
end  
toc  
release(videoPlayer);  
release(videoFReader);
```



ANEXO 09

PARA DETECCIÓN DE PERSONA EN EMBARQUE

```
close all;
clear all;
clc;
archivo1='video5';
stre1=strcat(archivo1, '.avi');
info=aviinfo(stre1);
Nf=info.NumFrames;
fps=info.FramesPerSecond;
M=info.Height;
N=info.Width;
r=0:255;
for i=1:50:Nf
    s1=VideoReader(stre1);
    f1=read(s1,i);
    f2=rgb2gray(f1);
    f3=f2(347:382,366:387);
    figure(1)
    imshow(uint8(f1));
    title('Imagen Actual');
    figure(2)
    subplot(1,2,1)
    imshow(uint8(f3));
    title('zona de interes');
    subplot(1,2,2)
    [his]=imhist(uint8(f3));
    stem(r,his)
    his=double(his);
    [M1,N1]=size(f3);
    fdp=his/(M1*N1);
    med=sum(r.*fdp');
    var=sum(((r-med).^2).*fdp');
```

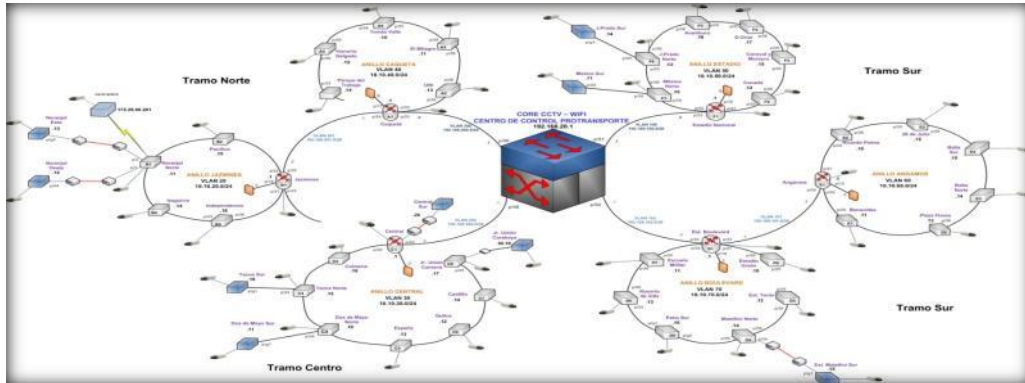


```
title(num2str(var));  
%pause  
if var>600;  
    disp('Persona Presente');  
    [y, Fs]=wavread('alarma.wav')  
    sound(y,Fs)  
end  
end;
```



ANEXO 10

RED DE FIBRA ÓPTICA METROPOLITANO



ANEXO 11

ESTACIONES DEL METROPOLITANO

ITEM	ESTACION METROPOLITANO
1	Terminal Naranjal
2	Carlos Izaguirre
3	Pacífico
4	Independencia
5	Los Jazmines
6	Tomas Valle
7	El Milagro
8	Honorio Delgado
9	UNI
10	Parque del Trabajo
11	Caquetá
12	Ramón Castilla
13	Tacna
14	Jirón de la Unión
15	Colmena
16	Estación Central
17	España
18	Quilca

19	2 de Mayo
20	Estadio Nacional
21	México
22	Canadá
23	Javier Prado
24	Canaval y Moreyra
25	Aramburú
26	Domingo Orué
27	Angamos
28	Ricardo Palma
29	Benavides
30	28 de Julio
31	Plaza de Flores
32	Balta
33	Boulevard
34	Estadio Unión
35	Escuela Militar
36	Fernando Terán
37	Rosario de Villa
38	Terminal Matellini

ANEXO 12

INSTALACIÓN DE CÁMARAS VS EMBARQUE

N°	Estación	TIPO	MODELO	UBICACIÓN	EMBARQUE	SENTIDO	COSTO
1	Terminal Naranjal	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	NORTE A SUR	\$1,500.00
2	Terminal Naranjal	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	NORTE A SUR	\$1,500.00
3	Terminal Naranjal	Cámara Fija	AXIS P1353	EXTERIOR	EMB3	NORTE A SUR	\$1,500.00
4	Terminal Naranjal	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	SUR A NORTE	\$1,500.00
5	Terminal Naranjal	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	SUR A NORTE	\$1,500.00
6	Terminal Naranjal	Cámara Fija	AXIS P1353	EXTERIOR	EMB3	SUR A NORTE	\$1,500.00
7	Carlos Izaquirre	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	NORTE A SUR	\$1,500.00
8	Carlos Izaquirre	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	NORTE A SUR	\$1,500.00
9	Carlos Izaquirre	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	SUR A NORTE	\$1,500.00
10	Carlos Izaquirre	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	SUR A NORTE	\$1,500.00
11	Pacífico	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	NORTE A SUR	\$1,500.00
12	Pacífico	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	NORTE A SUR	\$1,500.00
13	Pacífico	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	SUR A NORTE	\$1,500.00
14	Pacífico	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	SUR A NORTE	\$1,500.00
15	Independencia	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	NORTE A SUR	\$1,500.00
16	Independencia	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	NORTE A SUR	\$1,500.00
17	Independencia	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	SUR A NORTE	\$1,500.00
18	Independencia	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	SUR A NORTE	\$1,500.00
19	Los Jazmines	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	NORTE A SUR	\$1,500.00
20	Los Jazmines	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	NORTE A SUR	\$1,500.00
21	Los Jazmines	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	SUR A NORTE	\$1,500.00
22	Los Jazmines	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	SUR A NORTE	\$1,500.00
23	Tomas Valle	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	NORTE A SUR	\$1,500.00
24	Tomas Valle	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	NORTE A SUR	\$1,500.00
25	Tomas Valle	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	SUR A NORTE	\$1,500.00
26	Tomas Valle	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	SUR A NORTE	\$1,500.00
27	El Milagro	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	NORTE A SUR	\$1,500.00
28	El Milagro	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	NORTE A SUR	\$1,500.00
29	El Milagro	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	SUR A NORTE	\$1,500.00
30	El Milagro	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	SUR A NORTE	\$1,500.00
31	Honorio Delgado	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	NORTE A SUR	\$1,500.00
32	Honorio Delgado	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	NORTE A SUR	\$1,500.00
33	Honorio Delgado	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	SUR A NORTE	\$1,500.00
34	Honorio Delgado	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	SUR A NORTE	\$1,500.00
35	UNI	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	NORTE A SUR	\$1,500.00
36	UNI	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	NORTE A SUR	\$1,500.00
37	UNI	Cámara Fija	AXIS P1353	EXTERIOR	EMB3	NORTE A SUR	\$1,500.00
38	UNI	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	SUR A NORTE	\$1,500.00
39	UNI	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	SUR A NORTE	\$1,500.00
40	UNI	Cámara Fija	AXIS P1353	EXTERIOR	EMB3	SUR A NORTE	\$1,500.00
41	Parque del Trabajo	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	NORTE A SUR	\$1,500.00
42	Parque del Trabajo	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	NORTE A SUR	\$1,500.00
43	Parque del Trabajo	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	SUR A NORTE	\$1,500.00
44	Parque del Trabajo	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	SUR A NORTE	\$1,500.00
45	Caquetá	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	NORTE A SUR	\$1,500.00
46	Caquetá	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	NORTE A SUR	\$1,500.00
47	Caquetá	Cámara Fija	AXIS P1353	EXTERIOR	EMB3	NORTE A SUR	\$1,500.00
48	Caquetá	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	SUR A NORTE	\$1,500.00
49	Caquetá	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	SUR A NORTE	\$1,500.00
50	Caquetá	Cámara Fija	AXIS P1353	EXTERIOR	EMB3	SUR A NORTE	\$1,500.00
51	Ramon Castilla	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	NORTE A SUR	\$1,500.00
52	Ramon Castilla	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	NORTE A SUR	\$1,500.00
53	Ramon Castilla	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	SUR A NORTE	\$1,500.00
54	Ramon Castilla	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	SUR A NORTE	\$1,500.00
55	Tacna	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	NORTE A SUR	\$1,500.00
56	Tacna	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	NORTE A SUR	\$1,500.00
57	Tacna	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	SUR A NORTE	\$1,500.00
58	Tacna	Cámara Fija	AXIS P1353	EXTERIOR	EMB2	SUR A NORTE	\$1,500.00
59	Jiron de la Unión	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	NORTE A SUR	\$1,500.00
60	Jiron de la Unión	Cámara Fija	AXIS P1353	EXTERIOR	EMB1	SUR A NORTE	\$1,500.00

ANEXO 13

CARACTERÍSTICAS DE LA CÁMARA IP SELECCIONADA PARA LA IMPLEMENTACIÓN

Excelente calidad de imagen con resolución SVGA

- Tecnología Lightfinder
- Múltiples secuencias H.264
- PTZ digital
- Almacenamiento local

Resistente cámara fija de visión diurna y nocturna

La cámara de red AXIS P1353 es una cámara de red fija para visión diurna y nocturna con una calidad de imagen extraordinaria, que ofrece un excelente rendimiento H.264 y un diseño resistente. Ofrece una calidad de vídeo superior con exploración progresiva en varias secuencias H.264 individuales, además de secuencias Motion JPEG. Sus características de asistente de enfoque, enfoque posterior remoto y contador de píxeles simplifican la instalación.

Aplicaciones de video vigilancia exigentes

AXIS P1353 ofrece video vigilancia de alto rendimiento para aplicaciones tanto en interiores como en exteriores. Es perfecta para la seguridad en lugares como edificios gubernamentales e industriales, tiendas minoristas, aeropuertos, estaciones de ferrocarril, colegios y campus universitarios.

Tecnología Lightfinder.

La cámara AXIS P1353 incluye la tecnología Lightfinder de Axis. Esta excepcional sensibilidad a la luz, que mantiene los colores incluso en condiciones de escasa iluminación, se obtiene mediante una combinación de la experiencia de Axis en el procesamiento de imágenes, el desarrollo de sistemas en chip y la selección de los mejores componentes ópticos.

ANEXO 14

SEGUNDA ZONA DE INTERÉS

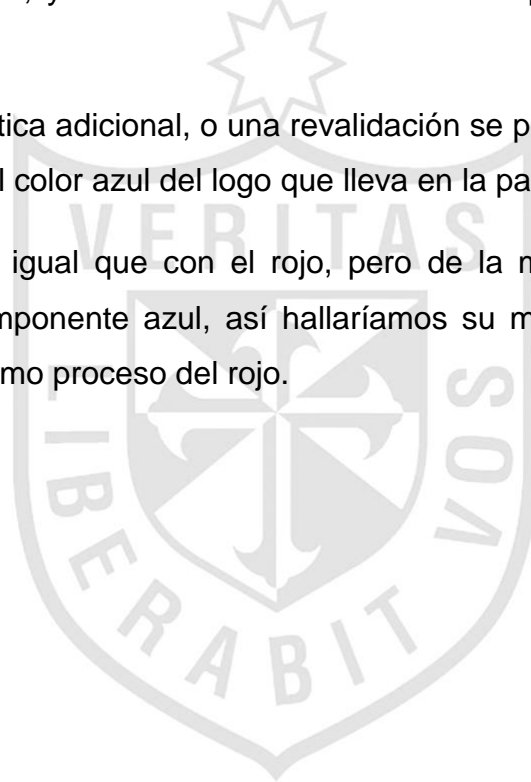
Se adicionó en los cálculos para la detección, una segunda zona de interés, para la detección de la persona.

La alerta se valida, realizando la detección de una persona en el embarque, luego que el bus haya partido.

Ya que si existe una persona en el embarque, y el bus a partido, puede ser que esté en peligro, ya sea de caída o inclusive atropello por el bus que viene.

Por una característica adicional, o una revalidación se puede detectar al bus, sería detectando el color azul del logo que lleva en la parte trasera.

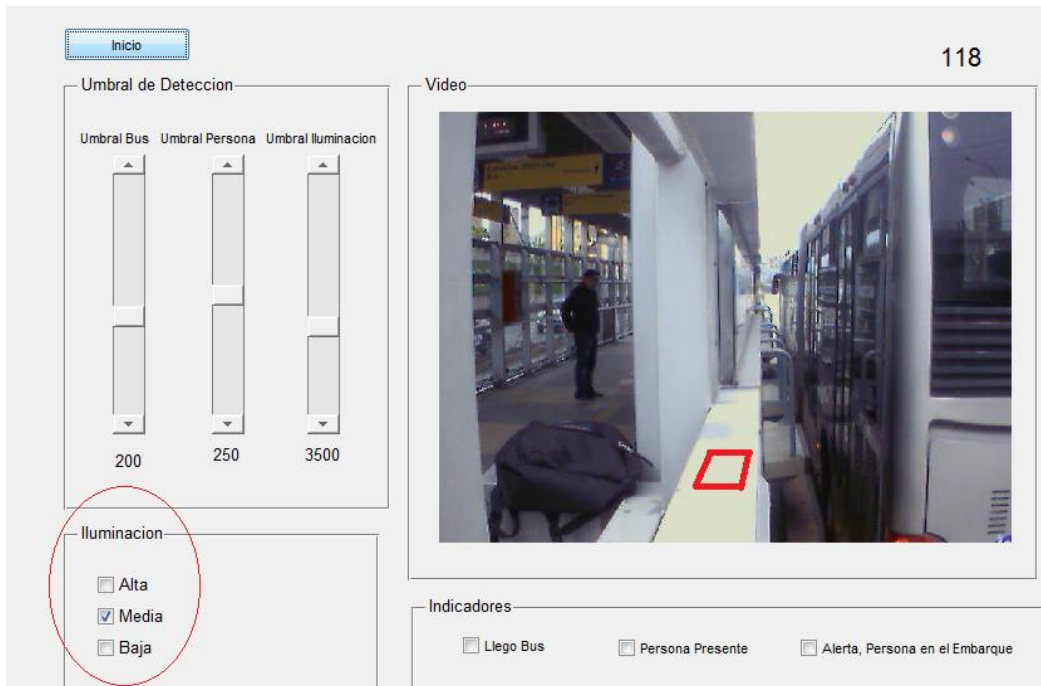
El algoritmo sería igual que con el rojo, pero de la matriz RGB esta vez sacaríamos el componente azul, así hallaríamos su masa, siendo este un escalar, con el mismo proceso del rojo.





Se implementó un algoritmo que detecte el nivel de iluminación, se creó un área de interés y se conto cuanta luz o color claro hay en este segmento.

De tal forma que determine según rango si es iluminación baja, media o alta.



Se implemento interfaz visual Guide.



Detección de incidencia en tiempo real.

Aviso vía celular desde centro de control a estación:

1 minuto.

Llegada de personal de estación a embarque:

2 minutos.

Tiempo total desde ocurre incidente y llega ayuda: 3 minutos.

Programación según observaciones detectadas.

```
function varargout = proyecto01(varargin)
% PROYECTO01 MATLAB code for proyecto01.fig
%   PROYECTO01, by itself, creates a new PROYECTO01 or raises the existing
%   singleton*.
%
%   H = PROYECTO01 returns the handle to a new PROYECTO01 or the handle to
%   the existing singleton*.
%
%   PROYECTO01('CALLBACK',hObject,eventData,handles,...) calls the local
%   function named CALLBACK in PROYECTO01.M with the given input arguments.
%
%   PROYECTO01('Property','Value',...) creates a new PROYECTO01 or raises the
%   existing singleton*. Starting from the left, property value pairs are
%   applied to the GUI before proyecto01_OpeningFcn gets called. An
%   unrecognized property name or invalid value makes property application
%   stop. All inputs are passed to proyecto01_OpeningFcn via varargin.
%
%   *See GUI Options on GUIDE's Tools menu. Choose "GUI allows only one
%   instance to run (singleton)".
%
% See also: GUIDE, GUIDATA, GUIHANDLES

% Edit the above text to modify the response to help proyecto01

% Last Modified by GUIDE v2.5 11-Feb-2015 20:34:21

% Begin initialization code - DO NOT EDIT
gui_Singleton = 1;
gui_State = struct('gui_Name',       mfilename, ...
                  'gui_Singleton',   gui_Singleton, ...
                  'gui_OpeningFcn', @proyecto01_OpeningFcn, ...
                  'gui_OutputFcn',  @proyecto01_OutputFcn, ...
                  'gui_LayoutFcn',   [], ...
                  'gui_Callback',    []);
if nargin && ischar(varargin{1})
    gui_State.gui_Callback = str2func(varargin{1});
end

if nargout
    [varargout{1:nargout}] = gui_mainfcn(gui_State, varargin{:});
else
    gui_mainfcn(gui_State, varargin{:});
end
% End initialization code - DO NOT EDIT

% --- Executes just before proyecto01 is made visible.
function proyecto01_OpeningFcn(hObject, eventdata, handles, varargin)
% This function has no output args, see OutputFcn.
% hObject    handle to figure
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)
% varargin   command line arguments to proyecto01 (see VARARGIN)

% Choose default command line output for proyecto01
handles.output = hObject;

% Update handles structure
guidata(hObject, handles);

% UIWAIT makes proyecto01 wait for user response (see UIRESUME)
% uiwait(handles.figure1);
```



```

% --- Outputs from this function are returned to the command line.
function varargout = proyecto01_OutputFcn(hObject, eventdata, handles)
% varargout cell array for returning output args (see VARARGOUT);
% hObject handle to figure
% eventdata reserved - to be defined in a future version of MATLAB
% handles structure with handles and user data (see GUIDATA)

% Get default command line output from handles structure
varargout{1} = handles.output;

%inicio
% --- Executes on button press in btn01.
function btn01_Callback(hObject, eventdata, handles)
%-----
% Seleccion del video con extension avi o wmv
[FileName Path]=uigetfile({'*.avi;*.wmv'}, 'Abrir Imagen');
if isequal(FileName,0)
    return
else
    video = VideoReader(FileName);

end
%-----
nFrame = 0;
isGood = true ;
while isGood
    set(handles.text1, 'String', nFrame);
    nFrame = nFrame + 1 ;
    try
        thisFrame = read(video, nFrame); % Seleccion de un (01) cuadro del video
%-----
% Algoritmo de Deteccion del Bus
A=uint8(thisFrame*2);
valor1=get(handles.slider5, 'Value'); % valor del slider5
d=deteccionbus(A,fix(valor1)); % d es binario
if d
    set(handles.alarma1, 'Value', 1);
else
    set(handles.alarma1, 'Value', 0);
end
set(handles.ubus, 'String', fix(valor1));
%-----
% Algoritmo de deteccion de las personas
valor2=get(handles.slider6, 'Value'); % valor del slider6
p1=personal(thisFrame,fix(valor2)); % p1 es binario
p2=persona2(thisFrame,fix(valor2)); % p1 es binario
if (p1|p2)
    set(handles.alarma2, 'Value', 1);
else
    set(handles.alarma2, 'Value', 0);

end
set(handles.upersona, 'String', fix(valor2));
%-----
% Algoritmo de deteccion de la iluminacion
valor3=get(handles.slider7, 'Value'); % valor del slider7
iL=iluminacion(thisFrame,fix(valor3)); % iL es binario
if iL==0 % iluminacion baja
    set(handles.baja, 'Value', 1);
    set(handles.media, 'Value', 0);
    set(handles.alta, 'Value', 0);
elseif iL==1 % iluminacion media
    set(handles.baja, 'Value', 0);
    set(handles.media, 'Value', 1);
    set(handles.alta, 'Value', 0);
elseif iL==2 % iluminacion alta
    set(handles.baja, 'Value', 0);
    set(handles.media, 'Value', 0);
    set(handles.alta, 'Value', 1);
end
set(handles.uilum, 'String', fix(valor3));
%-----
    imshow(thisFrame); % Muestra la imagen que se esta procesando en el GUI
%-----
% Deteccion de la persona en el Embarque.
if (~p1&p2&~d) % si p1=0,p2=1,d=0, entonces persona en el embarque

```

```

        set(handles.alarma3,'Value',1);
    elseif (p1&~p2&~d) % si p1=1,p2=0,d=0, entonces persona en el embarque
        set(handles.alarma3,'Value',1);
    elseif (p1&p2&~d) % si p1=1,p2=1,d=0, entonces persona en el embarque
        set(handles.alarma3,'Value',1);
    else
        set(handles.alarma3,'Value',0);
    end
end
%-----
catch
    isGood = false;
    nFrame = nFrame - 1 ;
end
end

function [llego]=deteccionbus(I,parametroR1) % Funcion Desarrollada por el equipo
de Investigacion
    llego=0;
    M=[];
    M(:,:,1)=I(373:450,480:570,1);
    M(:,:,2)=I(373:450,480:570,2);
    M(:,:,3)=I(373:450,480:570,3);

    M=uint8(M);
    [f,c,d]=size(M);
    M1=M;
    for m=1:f
        for n=1:c
            if M(m,n,2)>80 && M(m,n,3)>80
                M1(m,n,1)=0;
                M1(m,n,2)=0;
                M1(m,n,3)=0;
            end
        end
    end

    M1=uint8(M1);
    M2=M1;
    for m=1:f
        for n=1:c
            if M1(m,n,1)<100
                M2(m,n,1)=0;
                M2(m,n,2)=0;
                M2(m,n,3)=0;
            end
        end
    end

    M2=uint8(M2);
    M3=round(M2(:,:,1)/255);
    R1=sum(sum(M3));

    if R1>parametroR1
        llego=1;
        %disp('Llego bus metropolitano');
        %[y, Fs]=wavread('alarma.wav');
        %sound(y,Fs)
    elseif R1<parametroR1
        llego=0;
        %disp('No hay bus metropolitano');
    end

function [per1]=personal(f2,parametroR2) % funcion de deteccion de una persona P1
per1=0;
f3=f2(350:380,365:390); % ROI
BW = roicolor(f3,30,100); % Escala de grises rango de 30 a 100
R2=sum(sum(BW)); % perso de la imagen
if R2>parametroR2
    per1=1;
    %disp('Persona');
    %[y, Fs]=wavread('alarma.wav');
    %sound(y,Fs)
elseif R2<parametroR2
    per1=0;
    %disp('No hay Persona');
end
end

```

```

function [per2]=persona2(f2,parametroR3) % funcion de deteccion de una persona P2
per2=0;
f3=f2(300:325,360:380);
BW = roicolor(f3,30,100);
R3=sum(sum(BW));
if R3>parametroR3
    per2=1;
    %disp('Persona');
    %[y, Fs]=wavread('alarma.wav');
    %sound(y,Fs)
elseif R3<parametroR3
    per2=0;
    %disp('No hay Persona');
end

function [ilu]=iluminacion(f2,parametroR4)% funcion de deteccion de la
iluminacion
%ilu=0;
d=2500;
f3=f2(390:400,310:330);%           3500           6000
R4=sum(sum(f3))/10           % *-----*
if R4<parametroR4           %-----Para4-----Para4 + d-----
    ilu=0;                   % bajo           medio           alto
    %disp('Bajo');
elseif (R4>parametroR4 & R4<(parametroR4+d))
    ilu=1;
    %disp('Medio');
elseif R4>(parametroR4 + d)
    ilu=2;
    %disp('Alto');
end

%guide
% --- Executes during object creation, after setting all properties.
function winvideo_CreateFcn(hObject, eventdata, handles)
axis off;

% Hint: place code in OpeningFcn to populate winvideo

% --- Executes on slider movement.
function slider1_Callback(hObject, eventdata, handles)
handles.slider1=get(hObject,'Value'); %Carga en handles.slider1
handles.slider1=2000*handles.slider1;
set(handles.ubus,'String',handles.slider1); %Escribe el valor de

% --- Executes on slider movement.
function slider3_Callback(hObject, eventdata, handles)
handles.slider3=get(hObject,'Value'); %Carga en handles.slider1
handles.slider3=2000*handles.slider3;
set(handles.upersona,'String',handles.slider3); %Escribe el valor de

% --- Executes on slider movement.
function slider4_Callback(hObject, eventdata, handles)
% hObject    handle to slider4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hints: get(hObject,'Value') returns position of slider
%        get(hObject,'Min') and get(hObject,'Max') to determine range of slider

% --- Executes during object creation, after setting all properties.
function slider4_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider4 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

```

```

% --- Executes on slider movement.
function slider5_Callback(hObject, eventdata, handles)
handles.slider5=get(hObject,'Value'); %Carga en handles.slider5
set(handles.ubus,'String',fix(handles.slider5)); %Escribe el valor de
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function slider5_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider5 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on slider movement.
function slider6_Callback(hObject, eventdata, handles)
handles.slider6=get(hObject,'Value'); %Carga en handles.slider5
set(handles.upersona,'String',fix(handles.slider6)); %Escribe el valor de
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function slider6_CreateFcn(hObject, eventdata, handles)
% hObject    handle to slider6 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes during object creation, after setting all properties.
function ubus_CreateFcn(hObject, eventdata, handles)
% hObject    handle to ubus (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% --- Executes during object creation, after setting all properties.
function uipanel1_CreateFcn(hObject, eventdata, handles)

% --- Executes during object creation, after setting all properties.
function rb01_CreateFcn(hObject, eventdata, handles)
% hObject    handle to rb01 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% --- Executes during object creation, after setting all properties.
function rb02_CreateFcn(hObject, eventdata, handles)
% hObject    handle to rb02 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% --- Executes when selected object is changed in uipanel1.
function uipanel1_SelectionChangeFcn(hObject, eventdata, handles)
handles.umbral=0;
if (hObject==handles.rb01)
    handles.umbral=1;
elseif(hObject==handles.rb02)
    handles.umbral=2;
end
guidata(hObject, handles);

% --- Executes on button press in alarm1.
function alarm1_Callback(hObject, eventdata, handles)
% hObject    handle to alarm1 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB

```

```

% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of alarma1

% --- Executes on button press in alarma3.
function alarma3_Callback(hObject, eventdata, handles)
% hObject    handle to alarma3 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of alarma3

% --- Executes on button press in alta.
function alta_Callback(hObject, eventdata, handles)
% hObject    handle to alta (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of alta

% --- Executes on button press in media.
function media_Callback(hObject, eventdata, handles)
% hObject    handle to media (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of media

% --- Executes on button press in baja.
function baja_Callback(hObject, eventdata, handles)
% hObject    handle to baja (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of baja

% --- Executes on slider movement.
function slider7_Callback(hObject, eventdata, handles)
handles.slider7=get(hObject,'Value'); %Carga en handles.slider5
set(handles.uilum,'String',fix(handles.slider7)); %Escribe el valor de
guidata(hObject, handles);

% --- Executes during object creation, after setting all properties.
function slider7_CreateFcn(hObject, ~, handles)
% hObject    handle to slider7 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    empty - handles not created until after all CreateFcns called

% Hint: slider controls usually have a light gray background.
if isequal(get(hObject,'BackgroundColor'), get(0,'defaultUicontrolBackgroundColor'))
    set(hObject,'BackgroundColor',[.9 .9 .9]);
end

% --- Executes on button press in alarma2.
function alarma2_Callback(hObject, eventdata, handles)
% hObject    handle to alarma2 (see GCBO)
% eventdata  reserved - to be defined in a future version of MATLAB
% handles    structure with handles and user data (see GUIDATA)

% Hint: get(hObject,'Value') returns toggle state of alarma2

```