Universidad de las Ciencias Informáticas

Facultad de Ciencias y Tecnologías Computacionales

Trabajo de diploma para optar por el título de

Ingeniero en Ciencias Informáticas

Título: Video sensor para la detección y seguimiento de personas.

Autor: Ramiro Rodríguez Rivero

Tutor: MSc. Reinier Pupo Ruiz

La Habana, martes 20 de junio del 2017

"Año 59 de la Revolución"

“Confía en el tiempo, que suele dar dulces salidas a muchas amargas dificultades.”

Miguel de Cervantes

Declaración de Autoría

Declaro ser autor de la presente tesis que tiene por título: Video sensor para la detección y seguimiento de personas y reconozco a la Universidad de las Ciencias Informáticas los derechos patrimoniales de la misma, con carácter exclusivo. Para que así conste firmamos la presente a los ____ días del mes de _____ del año _____.

Autor: _____

Ramiro Rodríguez Rivero

Tutor: _____

MSc. Reinier Pupo Ruiz

Datos del Contacto

Autor: Ramiro Rodríguez Rivero

Correo electrónico: rrodriguezr@estudiantes.uci.cu

Tutor: MSc. Reinier Pupo Ruiz

Síntesis del tutor: Graduado de Ingeniero en Ciencias Informáticas en la UCI en el año 2009. Máster en Informática Aplicada en el 2015. Profesor de Práctica Profesional del Centro de Desarrollo de Geoinformática y Señales Digitales (GEYSED) de la Facultad de Ciencias y Tecnologías Computacionales de la UCI.

Correo electrónico: rpupo@uci.cu

A mi mamá por dedicarme su vida.

A mi padre por transmitirme esa energía de seguir adelante.

A ustedes porque soy el fruto de todo su esfuerzo, y espero que se sientan orgullosos.

Los quiero.

Resumen

Con el avance de las tecnologías de la información, el campo de la video vigilancia ha cobrado fuerza. En Cuba se han desarrollado las empresas que se dedican a la creación de software. En la Universidad de Ciencias Informáticas está el proyecto video vigilancia Xilema Suria 3.0, el cual cuenta con un módulo de análisis que permite procesar en tiempo real los flujos de videos obtenidos de las cámaras IP para detectar diferentes tipos de eventos. La presente investigación tiene como objetivo general desarrollar un video sensor para la detección y seguimiento de personas. Este ha sido implementado utilizando el algoritmo Viola and Jones que utiliza las cascadas Haar para la detección de las personas. Se realiza un seguimiento de la persona utilizando el área y la distancia entre objetos. Para la realización de este trabajo se utiliza la biblioteca de Visión por Computadora OpenCV. Los resultados que arroja el algoritmo implementado muestran una tasa de efectividad de 0.85 y de error de 0.15.

Palabras Clave: Cascadas Haar, detección y seguimiento de personas, OpenCV, Video Sensor, video vigilancia.

Abstract

With the advancement of information technologies, the field of video surveillance has gained impetus. In Cuba have been developed companies that are dedicated to the creation of software. In the University of Computer Science exists the video surveillance project named Xilema Suria 3.0, which has an analysis module that allows real-time processing of video streams obtained from IP cameras to detect different types of events. The present research aims to develop a video sensor for the detection and monitoring of people. This has been implemented using the Viola and Jones algorithm that uses Haar cascades for the detection of people. The person is tracked using the area and the distance between objects. In order to carry out this work, the OpenCV Computer Vision library is used. The results of the implemented algorithm show an effectiveness rate of 0.85 and an error rate of 0.15

Keywords: Haar cascades, detection and tracking of people, OpenCV, video sensor, video surveillance.

Índice

Índice de tablas

Índice de Figuras

Introducción

En los últimos tiempos, la seguridad se está convirtiendo en uno de los ejes fundamentales de la sociedad. La gran preocupación que generan los altos índices de violencia e inseguridad actuales en el mundo, motivan el estudio de nuevas tecnologías que permitan vivir con mayor tranquilidad a los habitantes, sobre todo, de las grandes ciudades, en donde la inseguridad es mayor. Según (Fría, 2016), la tasa de población penitenciaria en América Latina, creció y según las tendencias en los buscadores web, en los últimos 10 años el interés de las personas en términos como “Robo”, “Inseguridad o *Insecurity*” se encuentran en constante aumento.

Por el aumento de situaciones de seguridad, surgió la videovigilancia en 1942 (Reina, 2016), que es el cuidado de un lugar mediante cámaras de video. En sus inicios utilizaba una señal y trasmisión analógica que permitía la grabación del material de video en cintas magnéticas, lo cual limitaba el periodo de grabación que podía archivarse. El análisis de la secuencia de video era realizado por una persona que atendía varios monitores a la vez, lo que provocaba una posible pérdida de información si la persona se distraía.

Con el desarrollo se permitió centrar la atención del operador humano en situaciones de interés, debido al surgimiento de los primeros algoritmos de procesado de video y los métodos de compresión digital. Esta vigilancia basó en métodos de procesamiento y comunicación híbridos analógico-digitales, o completamente digitales.

Los sistemas actuales procesan el flujo de imágenes en tiempo real utilizando cámaras IP. Estos sistemas de vigilancia permiten que cada cámara sea independiente de otra en cuanto al tratamiento de la información captada, así los sistemas son más confiables en estabilidad y en la información que brinda. También cuentan con procesamiento inteligente autónomo, por lo que se incrementa la seguridad, ejemplo de esto son los videos sensores.

Un video sensor no es más que una herramienta de análisis de video digital que ofrece información significativa proveniente de una secuencia de video (Pineda, 2010). Existen diferentes videos sensores, entre los que se pueden mencionar los de detección de movimiento (Hueber, y otros, 2014), conteo de personas (Parody, y otros, 2015) y para extracción de matrículas en vehículos (Álvarez, 2015). En esta rama, la actividad relacionada con la capacidad para diferenciar personas de objetos y poder seguirlas ha cobrado gran importancia. Esta se lleva a cabo a través de un análisis de características del sujeto en la imagen digital o fotograma en cuestión, comparándolas con otras almacenadas previamente.

La Universidad de las Ciencias Informáticas (UCI) tiene al Centro Geoinformática y Señales Digitales (GEYSED), perteneciente a la Facultad de Ciencias y Tecnologías Computacionales de dicha institución, que está desarrollando el sistema de video vigilancia Xilema Suria en su versión 3.0. Este sistema cuenta con 6 módulos: Administración, Visor, Visor Apk, Autonomía, Grabador y Análisis. Este último se encarga de acoplar un conjunto de video sensores, que contienen funcionalidades de procesamiento de video que le aportan al proceso de vigilancia la capacidad de automáticamente y en tiempo real emitir y registrar alarmas.

Los trabajadores de seguridad son incapaces de llevar el control exacto de lo que ocurre en varias cámaras simultáneamente, y a su vez examinar las personas y sus recorridos al salir y entrar en cierto local. Por otra parte, se dificulta considerablemente identificar y seguir las personas que puedan acceder a cierto recurso y llevar el registro de estas. Por razones de seguridad, a veces es necesario llevar el control de una persona que sea de interés. Si esto no se realizara o se ejecutara incorrectamente podría pasarse por alto una situación determinada y provocar escenarios indeseables que amenacen la seguridad del entorno bajo vigilancia.

A partir de la situación descrita anteriormente, surge el siguiente **problema** a resolver: ¿Cómo detectar y monitorear personas en flujos de video provenientes de cámaras IP?, definiéndose como **objeto de estudio:** El procesamiento de video, enmarcado en el **campo de acción:** el proceso de detección y monitoreo de personas en flujos de video provenientes de cámaras IP.

Para darle solución al problema antes planteado, se define como **objetivo general:** Desarrollar un Video Sensor para la detección y monitoreo de personas en flujos de video provenientes de cámaras IP.

Con el propósito de dar cumplimiento al objetivo general definido, se establecen las siguientes **preguntas de la investigación:**

- ✓ ¿Cuáles son los fundamentos teóricos en los que se basa el proceso de detección y seguimiento de personas?
- ✓ ¿Cuáles son las características y capacidades que debe tener un algoritmo que permita la detección y seguimiento de personas?
- ✓ ¿Cómo desarrollar el video sensor para la detección y seguimiento de personas?
- ✓ ¿Cómo validar el correcto funcionamiento del video sensor para la detección y seguimiento de personas?

Con el propósito de dar respuestas a las preguntas de investigación, se establecen las siguientes **tareas de la investigación**:

- ✓ Caracterizar los algoritmos realizados sobre la detección y seguimiento de personas.
- ✓ Analizar y seleccionar las herramientas y tecnologías a utilizar en el proceso de desarrollo.
- ✓ Realizar el análisis y diseño de la solución.
- ✓ Realizar del modelo de implementación.
- ✓ Implementar de los componentes del modelo.
- ✓ Diseñar de los casos de prueba.
- ✓ Ejecutar las pruebas al rendimiento y funcionamiento del algoritmo.

Para la organización y estructuración del trabajo investigativo se utilizarán los **métodos empíricos,** que según (González, 2011) describen y explican las características fenomenológicas del objeto, representan un nivel de la investigación cuyo contenido procede de la experiencia y es sometido a cierta elaboración racional, y también según (González, 2011) los **métodos lógicos,** se basan en el estudio histórico del fenómeno, ponen de manifiesto la lógica interna de su desarrollo, de su teoría y encuentra el conocimiento más profundo de su esencia, entre ellos se encuentran:

Observación: Con su puesta en práctica se podrá determinar los beneficios existentes en la creación de un video sensor permitiendo valorar los beneficios y factibilidad del proceso.

Analítico-Sintético: Facilitará mediante el análisis, la síntesis y la búsqueda de lo fundamental en la bibliografía consultada sobre el proceso de detección y monitoreo de personas en flujos de video para poder efectuar una amplia investigación sobre los elementos que se relacionan con el objeto de estudio.

Histórico-Lógico: Permitirá estudiar la evolución de la detección y seguimiento de personas con el objetivo de reproducir y mejorar en el plano teórico lo más importante de este fenómeno y estudiar sus posibles aplicaciones para el desarrollo del video sensor.

El Trabajo de Diploma está estructurado de la siguiente manera: Introducción, tres capítulos, conclusiones, recomendaciones, referencias bibliográficas y bibliografía.

Capítulo 1: Fundamentación Teórica:

En este capítulo se presentará la definición del marco teórico de la investigación. En el mismo se abordarán los conceptos de pixel, imagen digital o fotograma, video digital y sistema de video vigilancia. Se expondrán características de sistemas informáticos existentes vinculados a la detección y seguimiento de personas. Se abordará un estudio de la metodología, herramientas, tecnologías y lenguajes a utilizar en el desarrollo del video sensor.

Capítulo 2: Análisis y Diseño del sistema:

En el capítulo se describirá la propuesta de solución para el problema de la investigación. Mediante el modelo de dominio se observará la representación visual de clases conceptuales u objetos reales que se representarán en el video sensor a desarrollar. Describiendo los principales conceptos del entorno que serán objeto de análisis para la realización de la fase de análisis y diseño del video sensor. Posteriormente se realizará el levantamiento de requisitos funcionales y no funcionales para darle cumplimiento al objetivo general de la investigación. Además, se mostrarán los diagramas de clases del diseño, por medio de los cuales se representarán la estructura estática del sistema y los patrones de arquitectura y diseño que se utilizarán en el sistema.

Capítulo 3: Implementación y Pruebas:

En el capítulo se abordará sobre las actividades que se llevan a cabo durante la fase de implementación y pruebas. Se analizará el Modelo de Implementación, así como una descripción de cómo los elementos del modelo de diseño se implementarán en términos de componentes. Además, se diseñará y aplicarán las pruebas para comprobar el correcto funcionamiento del video sensor.

Capítulo 1: Fundamentación Teórica

1.1. Introducción

En el presente capítulo se hace referencia a las principales definiciones que servirán como punto de partida para un mayor entendimiento de los principales conceptos y definiciones que se manejan en esta investigación. Se realiza un estudio de sistemas y algoritmos que existen en el mundo que abordan la detección y seguimiento de personas. Se describen las tendencias y tecnologías actuales, así como la metodología utilizada para el modelado de los procesos definidos. Se explica en detalle el objeto de estudio propuesto y caracterizan las herramientas y tecnologías seleccionadas para el desarrollo de la aplicación.

1.2. Definiciones Generales

A continuación, se muestran una serie de conceptos asociados al dominio de la investigación para un mayor entendimiento de la misma.

Pixeles: En (García, 2005) se plantea que un pixel es una contracción de imagen ya que es la unidad mínima que forma una imagen digital en lenguaje binario de 0 y 1. En (Alegsa, 2016) se describe al píxel como la menor unidad posible con la que se compone cualquier imagen digital en una computadora. Por lo que se puede concluye que el pixel constituye la menor unidad de la que se compone una imagen, y contiene la intensidad de color de ese punto.

Imagen Digital o Fotograma: En (Chávez, 2010) se plantea que una imagen digital son varias colecciones de pixel cuyos valores distintos forman figuras. Es decir, es la representación visual de un elemento que se logra a partir de técnicas enmarcadas en la fotografía. En el libro Ilustración Digital (García, 2005) se describen como la traducción de los valores de color y luminosidad a un lenguaje binario de 0 y 1. En (Gonzalez, 2011) se explica que una imagen digital puede ser considerada una matriz cuya fila y los índices de la columna identifican un punto en la imagen, y el valor matricial correspondiente del elemento identifica el nivel de gris en ese momento. Por lo que se puede concluir que una imagen digital es una matriz de pixeles donde cada uno contiene la intensidad de color que forman y delimitan los objetos que componen.

Video Digital: En (Alegsa, 2016) se detalla que un video digital es una secuencia de imágenes digitales o fotogramas que, ejecutadas en secuencia, simulan movimiento y se almacenan en un determinado formato digital de video como, AVI, MPG, RealVideo y WMV. En (Fischer, 2010) se define como un tipo de sistema de grabación de video, que funciona usando una representación digital acompañada de audio que es almacenada en discos duros o memorias flash. En otras palabras, para la presente investigación un video

digital será una secuencia de imágenes digitales o fotogramas que no necesariamente están acompañadas de audio y ejecutadas en secuencia simulan movimiento.

Video Sensor: En (Bhanu, y otros, 2011) se define como un algoritmo que permite procesamiento de algún tipo sobre un flujo de video para identificar variaciones significativas en la secuencia de imágenes. Entonces se puede decir que un video sensor es un algoritmo que procesa el video para detectar automáticamente alguna situación de interés que ocurra en la secuencia de video.

Sistema de Video Vigilancia IP: En (Mata, 2010) se describe como un sistema de video IP que permite supervisar video y grabarlo desde cualquier lugar de la red ya que permite una monitorización remota en tiempo real, centralizando las labores de monitorización, almacenamiento y gestión en una central de alarmas ubicada en un emplazamiento diferente al del espacio monitorizado. En (Lucena, 2004) se plantea que la video vigilancia consiste en instalar cámaras de vídeo para vigilar determinadas áreas, y podrán incorporar equipos que posibiliten almacenar la imágenes, transmitirlas o utilizarlas. Por lo tanto, se puede definir como una tecnología de vigilancia IP que consiste en instalar cámaras de video para ser vistas y usadas para la monitorización, almacenamiento y gestión en una central de alarmas de forma remota en tiempo real.

1.3. Objeto de Estudio

A continuación, se describen los procesos involucrados en el objeto de estudio, en los que se expone lo fundamental de cada paso en el procesamiento de imágenes digitales para el análisis de un flujo de video.

1.3.1. Descripción General

El procesamiento de imágenes digitales constituye el punto de partida para el análisis de flujos de video digitales, este se define como un conjunto de técnicas y procesos para descubrir o hacer resaltar información contenida en una imagen usando como herramienta principal una computadora. Al examinar una imagen digital o fotograma, esta puede ser modificada, y a su vez se le pueden extraer características o trabajar simplemente con alguna región de dicha imagen. En (Torres, 1996), se plantea*: "Al conjunto de técnicas y procesos para descubrir o hacer resaltar información contenida en una imagen usando como herramienta principal una computadora se le conoce como Procesamiento de Imágenes Digitales (PID)."*

El primer paso consiste en adquirir el fotograma, para ello se necesita una cámara IP o un video almacenado que aporte un flujo de video del cual se puedan extraer los fotogramas que lo componen.

Una vez que se obtiene el fotograma, este es preprocesado con el objetivo de ser mejorado y si es necesario, restaurarlo para así garantizar que la fase final del procesamiento tenga mayores posibilidades de éxito. El preprocesamiento consiste en aplicar algoritmos para, por ejemplo: mejorar el contraste,

suprimir ruido y aislar regiones que por sus características indiquen la existencia real de algún tipo de objeto.

El paso siguiente es la segmentación, su objetivo es dividir el fotograma en regiones de objetos que la forman y retorna un conjunto de puntos por cada región encontrada. Aplicar algoritmos de segmentación en la solución propuesta a desarrollar tiene como objetivo obtener las áreas que definen al objeto persona dentro de la imagen.

Luego de realizar una correcta segmentación se procede a representar las regiones de interés, esta representación se puede realizar de dos maneras, por contorno o por región interna.

Se realiza la descripción después de haber representado las regiones de interés, con el objetivo de extraer los datos que las caracterizan, se recibe un objeto y se retornan los rasgos asociados a él.

Posteriormente se lleva a cabo el reconocimiento, el cual se encarga de obtener los rasgos característicos de cada uno de los objetos y les asocia una categoría predefinida, y para finalizar se pone en práctica la interpretación cuya misión es asociar un significado o acción al conjunto de objetos reconocidos, este último paso no se realiza en la presente investigación.

1.4. Situación actual del dominio del problema

El sistema de Video Vigilancia Xilema Suria en su versión 3.0 tiene su enfoque fundamental en la seguridad y protección de las instalaciones que lo utilizan. El análisis es pieza clave en este proceso pues está compuesto por video sensores que garantizan el procesado del flujo de video, en función de detectar la ocurrencia de alguna anomalía y notificar al módulo de análisis. Dichos video sensores están relacionados con: la detección de tapado de las cámaras, reconocimiento facial, detección de pérdida de foco de las cámaras, entre otros. Todos estos procesos contribuyen a la creación de un sistema robusto. Si esto no se realizara o se ejecutara incorrectamente podría pasarse por alto una situación determinada y provocar pérdidas de recursos que no serían fáciles de recuperar.

La implantación de un video sensor para el reconocimiento y seguimiento de personas posibilitaría que sea capaz de un mayor control, brindando apoyo a la vigilancia humana. Por otra parte, permitiría que los hechos delictivos fueran detectados con mayor facilidad.

1.5. Soluciones existentes.

Como resultado del estudio de distintas fuentes bibliográficas se han encontrado varios trabajos que se han realizado sobre la detección y seguimiento de personas.

En el año 1999, David Lowe propuso el método *Scale Invariant Feature Transform* (SIFT) (Lowe, 1999), que se centra en buscar puntos característicos que cumplen criterios espacio-escalares. Los descriptores

se calculan a través de la orientación de los gradientes de cada punto. Así se extraen puntos característicos invariantes y distintivos de una imagen que pueden ser usados para mejorar la correspondencia entre dos vistas diferentes de un objeto o una escena. Esto podría ser aplicado después de la detección de una persona para después hacer un seguimiento de la misma.

En (Jones, 2004) se propusieron a los "Descriptores Wavelet Haar" que mediante el algoritmo de Viola and Jones permiten definir de manera robusta clases de objetos complejos, siendo invariantes a cambios de color y de textura. Estos se emplean habitualmente para la descripción de personas. Presentan la capacidad de codificar rasgos tales como cambios de intensidades a diferentes escalas. La base de wavelet más sencilla es la de Haar que consiste en que se recorre la imagen con una ventana a la que se le aplican varios clasificadores en serie, cada uno más complejo que el anterior, los cuales usan las características para confirmar o descartar la hipótesis de que se trata del objeto buscado. Si la hipótesis se rechaza en cualquier nivel, el proceso no continúa, pero si se confirma todos los filtros significará que se ha detectado el objeto deseado. Los patrones se consideran girados en varios posibles ángulos. Además, el algoritmo puede ejecutarse a varias escalas para obtener objetos de diferentes tamaños o de tamaño desconocido. (Bhatia, 2007)

También en el año 2006 se desarrolló el *Speeded Up Robust Features* (SURF) (Herbert Bay, 2006) en Austria. Este está parcialmente inspirado en SIFT y se ha demostrado que en la práctica en la totalidad de los casos se consigue mejorar el rendimiento de este algoritmo. Se basa en el cálculo del determinante de la matriz Hessiana (*DoH: Determinant of Hessian*) para la detección de puntos de interés y en las *wavelets* de Haar para la descripción de dichos puntos. A pesar de haberse desarrollado en el año 2006 hoy en día es muy utilizado en el reconocimiento, clasificación o reconstrucción de objetos 3D. (Herbert, 2016)

En (Txarterina, 2013), se desarrolló un detector para peatones, probado sobre secuencias de vídeos, usando la técnica de histogramas orientados de flujo óptico diferencial combinado con los descriptores de apariencia de Histogramas Orientados a Gradientes, estos detectan las regiones en movimiento y las segmentan del fondo de la escena. Con base a las proporciones se clasifican y segmentan humanos de otros objetos móviles.

En (Omid, y otros, 2014) presentan un sistema de detección y seguimiento multipersonal basado en RGB-D en tiempo real, adecuado para robots móviles y cámaras portátiles. El enfoque combina la estimación de odometría visual RGB-D, procesamiento de región de interés, estimación de plano de tierra, detección de peatones y componentes de seguimiento de múltiples hipótesis en un sistema de visión robusto que funciona a más de 20fps en un portátil. Dado que la detección de objetos es el componente más caro de cualquier integración, invirtieron un esfuerzo significativo en aprovechar al máximo la información de profundidad

disponible. En particular, propusieron utilizar dos detectores diferentes para diferentes rangos de distancia. Para el rango cercano (hasta 5-7m), presentaron un detector de cuerpo superior de profundidad extremadamente rápido que permite el rendimiento del sistema de velocidad de video en un solo núcleo de CPU cuando se aplica a sensores Kinect. Para cubrir también rangos de distancia más lejanos, opcionalmente agregaron un detector HOG de cuerpo completo basado en apariencia (que se ejecuta en la GPU) que explora la geometría de la escena para restringir el espacio de búsqueda. El enfoque puede funcionar tanto con entrada Kinect RGB-D para configuraciones de interior como con entrada de profundidad estéreo para escenarios al aire libre.

En (Russell Stewart, 2016) se explica que los detectores de personas actuales operan escaneando una imagen de una ventana deslizante o clasificando un conjunto discreto de propuestas y proponen un modelo que se basa en la descodificación de una imagen en un conjunto de detecciones de personas. El sistema toma una imagen como entrada y emite directamente un conjunto de distintas hipótesis de detección. Debido a que generan predicciones conjuntamente, son innecesarias las etapas comunes de postprocesamiento como la supresión no máxima. Utilizaron una capa *Long Short Term Memory* (LSTM) recurrente para la generación de secuencias y capacitaron a un modelo de extremo a extremo con una nueva función de pérdida que opera en conjuntos de detecciones basados en las predicciones de los objetos.

Aporte de las soluciones estudiadas

Al realizar un estudio del estado del arte de los principales algoritmos, herramientas y tecnologías más utilizadas en los sistemas de video vigilancia actuales de detección y seguimiento de personas se logró una mayor comprensión de la problemática permitiendo identificar los algoritmos, y sistemas que se utilizan en el mundo en la detección y seguimiento de personas. Los algoritmos identificados fueron el SIFT, el modelo de predicción de personas, SURF y Viola and Jones. De los algoritmos identificados se escogió Viola and jones ya que este permite crear un clasificador en cascada que puede ser adiestrado para la detección de cualquier objeto y el tiempo de ejecución de este es muy bueno para un sistema de visión por computadora en tiempo real logrando un gran rendimiento en cuanto a procesamiento. Se identificó un sistema llamado “Sistema de detección y seguimiento multipersonal basado en RGB-D en tiempo real,” adecuado para robots móviles y cámaras portátiles que hubiera podido ser usado en la presente investigación, pero debido a que es un sistema propietario no se pudo utilizar.

1.6. Herramientas y Tecnologías

El proceso de desarrollo de software definido por (Pressman, 2010) es una estructura para las actividades, acciones y tareas a fin de construir un software de alta calidad. Un proceso de software define el enfoque adoptado mientras se hace ingeniería sobre el software. Pero la ingeniería de software también incluye

tecnologías que pueblan el proceso: métodos técnicos y herramientas automatizadas. Más importante aún, la ingeniería de software es llevada a cabo por personas creativas y preparadas que deben adaptar un proceso maduro de software a fin que resulte apropiado para los productos que construyen y para las demandas de su mercado. O sea, imponen un proceso disciplinado sobre el desarrollo de software en aras de lograr una mayor eficiencia y predictibilidad.

Debido a que los requisitos y complejidad de un software pueden ser tan variados con respecto a otros, se han creado diferentes tipos de metodologías, estas pueden agruparse en dos grandes grupos:

1. **Metodologías ágiles/ligeras:** Se caracterizan por ser procesos de desarrollo adaptables incrementalmente. Para lograr la adaptación incremental, un equipo ágil requiere retroalimentación con el cliente en intervalos cortos, para que este pueda ir sugiriendo cambios y ser corregidos con prontitud. (Pressman, 2010)

2. **Metodologías prescriptivas:** Estas establecen las actividades a desarrollar, herramientas a utilizar y notaciones que se emplearán. Requieren una extensa documentación, ya que pretende prever todo de antemano. Este tipo de metodología es más eficaz y necesaria cuanto mayor es el proyecto que se pretende realizar respecto a tiempo y recursos que son necesarios emplear. Son orientadas al control de los procesos. (Jacobson, 2000)

Proceso Unificado Ágil UCI (*Agile Unified Process, AUP*)

El Proceso Unificado Ágil de Scott Ambler o *Agile Unified Process* (AUP) es una versión simplificada del Proceso Unificado de Racional (RUP). Este describe de una manera simple y fácil de entender la forma de desarrollar aplicaciones de software de negocio usando técnicas ágiles y conceptos que aún se mantienen válidos en RUP. El AUP aplica técnicas ágiles incluyendo:

- Desarrollo Dirigido por Pruebas (TDD en inglés).
- Modelado ágil.
- Gestión de cambios ágil.
- Refactorización de Base de Datos para mejorar la productividad.

Al igual que en RUP, en AUP se establecen cuatro fases que transcurren de manera consecutiva que son inicio, elaboración, construcción y transición, pero al ser adaptada a las características de cada proyecto. Se decide hacer una variación de la metodología AUP, de forma tal que se adapte al ciclo de vida definido para la actividad productiva en la UCI.

De las 4 fases que propone AUP (Inicio, Elaboración, Construcción, Transición) se decide para el ciclo de vida de los proyectos de la UCI mantener la fase de Inicio, pero modificando el objetivo de la misma, se unifican las restantes 3 fases de AUP en una sola, a la que llamara Ejecución y se agrega la fase de Cierre.

AUP propone siete disciplinas (Modelo, Implementación, Prueba, Despliegue, Gestión de configuración, Gestión de proyecto y Entorno), se decide para el ciclo de vida de los proyectos de la UCI tener 7 disciplinas también, pero a un nivel más atómico que el definido en AUP. Los flujos de trabajos: Modelado de negocio, Requisitos y Análisis y diseño en AUP están unidos en la disciplina "Modelo", en la variación para la UCI se consideran a cada uno de ellos disciplinas. Se mantiene la disciplina "Implementación", en el caso de Prueba se desagrega en 3 disciplinas: "Pruebas Internas, de Liberación y Aceptación". Las restantes 3 disciplinas de AUP asociadas a la parte de gestión para la variación UCI se cubren con las áreas de procesos que define CMMI DEV v1.3 para el nivel 2, serían CM (Gestión de la configuración), PP (Planeación de proyecto) y PMC (Monitoreo y control de proyecto).

Herramienta de modelado de los artefactos de AUP UCI.

A lo largo del ciclo de vida de desarrollo de un software, un conjunto de programas y ayudas dan asistencia a los programadores, analistas e ingenieros de software. Estas son las llamadas herramientas CASE (Ingeniería de software asistida por computadora). Para que las tareas de un proyecto sean organizadas y completadas de forma eficiente, se necesita de un mecanismo que se encargue de ello. El objetivo principal de estas herramientas es la automatización de esos procesos y facilitar la coordinación de eventos que necesitan ser tratados en el ciclo de vida de desarrollo del software. (Gómez, 2009)

Lenguaje de Modelado (UML) 2.0

UML en su versión 2.0 según (Pressman, 2010) es un lenguaje estándar para escribir diseños de software que se centra en la representación gráfica de un sistema y a su vez indica cómo crear y leer los modelos. Las funciones principales de este lenguaje son las siguientes:

- Visualizar: Permite mostrar un sistema de forma gráfica, facilitando que otra persona ajena a este pueda entenderlo.
- Especificar: Permite definir cuáles son las características de un sistema antes de su construcción.

Construir: A partir de los modelos especificados se pueden construir los sistemas diseñados.

Documentar: Los propios elementos gráficos sirven como documentación del sistema desarrollado que pueden servir para su futura revisión.

Visual Paradigm 8.0

Visual Paradigm en su versión 8.0 es una herramienta profesional para el UML, que soporta el ciclo de vida completo del desarrollo de software: análisis y diseño orientados a objetos, construcción, pruebas y despliegue. El software de modelado UML ayuda a una más rápida construcción de aplicaciones de calidad, mejores y a un menor coste. Permite dibujar todos los tipos de diagramas de clases, código inverso, generar código desde diagramas y generar documentación. Presenta la desventaja de que requiere mantener varios hilos del programa en proceso, y tiene problemas de integración con otras herramientas de desarrollo. (Fuhrmann, 2011)

Microsoft Visual Studio 2013

Microsoft Visual Studio 2013 es un entorno de desarrollo integrado (IDE, por sus siglas en inglés) para sistemas operativos Windows. Soporta múltiples lenguajes de programación tales como C++, C#, Visual Basic .NET, F#, Java, Python, Ruby, PHP; al igual que entornos de desarrollo web como ASP.NET, MVC, y Django, a lo cual se le suman las nuevas capacidades online bajo Windows Azure en forma del editor Monaco. (Halvorson, 2013)

C++ estándar 11 como Lenguaje de Programación

C++ es un lenguaje imperativo orientado a objetos derivado del C. Es una mejoría sobre muchas de las características de C y proporciona capacidades de P.O.O. (Programación Orientada a Objeto) que aporta mucho para incrementar la productividad, calidad y reutilización del software. Este lenguaje es muy eficaz en cuanto a rapidez y uso de memoria en las aplicaciones que se obtienen. Las bibliotecas estándar de C++ proporcionan un conjunto extenso de capacidades de entrada/salida (Deitel, 2016). En el diseño del C++ primó sobre todo la velocidad de ejecución del código. C++ se ha seleccionado para la realización del video sensor ya que al ser uno de los lenguajes más rápidos en cuanto a ejecución, es de vital importancia a la hora de trabajar con flujos de video. Es uno de los más populares y utilizados a nivel mundial por lo que existe mucha bibliografía de apoyo. Además, la biblioteca a utilizar (OpenCV) está desarrollada en C++. Por las características antes mencionadas C++ resulta ser este el lenguaje idóneo para realizar el video sensor.

OpenCV 2.4.13

OpenCV (*Open Source Computer Vision Library*) (team, 2017) es una biblioteca de funciones en C y C++ desarrollado por Intel que proporciona un alto nivel de funciones para el procesado de imágenes, visión artificial, captura de video y visualización de imágenes. Es de código abierto, gratuita, multiplataforma (disponible para entornos MS Windows, Mac OS y Linux), está desarrollada bajo la licencia BSD (*Distribución de Software Berkeley*), es rápida, de fácil uso y está en continuo desarrollo. Estas bibliotecas permiten a los programadores crear aplicaciones poderosas en el dominio de la visión digital.

OpenCV, debido a sus características y capacidades para el procesamiento de imágenes digitales es muy utilizada hoy en día, por ejemplo, compañías bien establecidas como Google, Yahoo, Microsoft, Intel, IBM, Sony, Honda y Toyota. Hay muchas *startups* como Applied Minds, VideoSurf y Zeitera, que hacen uso extensivo de OpenCV. Los usos desplegados de OpenCV abarcan desde la unión de imágenes de *streetview*; detección de intrusiones en video de vigilancia en Israel; supervisión de equipos de minas en China; la ayuda a robots a navegar y recoger objetos en *Willow Garage*; detección de ahogamiento en piscinas en Europa, España y Nueva York; inspección de las etiquetas de los productos de fábricas; detección de rostros en Japón.

Tiene interfaces C++, C, Python, Java y MATLAB y soporta Windows, Linux, Android y Mac OS. OpenCV se inclina principalmente hacia aplicaciones de visión en tiempo real. En estos momentos se están desarrollando activamente una completa gama de interfaces CUDA y OpenCL. Hay más de 500 algoritmos y unas muchas más funciones que componen o apoyan esos algoritmos. OpenCV se escribe de forma nativa en C ++ y tiene una interfaz que funciona perfectamente con contenedores STL. Para la realización del video sensor para detección y seguimiento de personas se seleccionó esta biblioteca debido a que cuenta con funciones que facilitan el trabajo con imágenes que contengan personas. También ayudó para su selección que es la utilizada en el proyecto Video Vigilancia en su versión SURIA 3.0.

1.7. Conclusiones Parciales

Al concluir este capítulo se definieron los principales conceptos que regirán la investigación. Esto permite comprender la situación problemática y los procesos involucrados en el objeto de estudio. Por lo que posibilita comprender la situación actual del problema para un correcto estudio de las soluciones existentes en el campo de la detección y seguimiento de personas.

El estudio de las soluciones existentes permitió el conocimiento de los algoritmos y sistemas usados para la detección y seguimiento de personas. De los algoritmos identificados se escogió Viola and Jones ya que posibilita hacer un entrenamiento de cualquier objeto y su tiempo de ejecución es muy bueno para su uso en un flujo de video digital en tiempo real.

Se seleccionaron varias herramientas y tecnologías que por su compatibilidad y uso en los procesos de desarrollo de software de la Universidad permiten un eficiente avance en la aplicación para lograr el objetivo de esta investigación.

Capítulo 2: Análisis y diseño del sistema

2.1 Introducción

En este capítulo se realiza la propuesta inicial del sistema, describiendo el dominio en el que se enmarca el mismo. Se identifican los requisitos funcionales y no funcionales, con el fin de comprender las características del sistema, en función de procesamiento, rendimiento y sistemas operativos. Se puntualizan los casos de uso del sistema identificados, presentando la descripción de estos. También se definen las clases de análisis y diseño de los casos de uso; así como los diagramas de secuencia. Además, se presenta la arquitectura del sistema, y los patrones de diseño que se manejan para la implementación del sistema.

2.2 Modelo de Dominio

Un Modelo de Dominio es una representación visual de clases conceptuales u objetos reales en un dominio específico. Este consiste en un conjunto de diagramas de clases, sin definición de operaciones. Constituye una base de conocimiento con los principales conceptos asociados al desarrollo del sistema o como expresó Roger Pressman " La meta de análisis del dominio es clara: encontrar o crear aquellas clases o patrones de análisis que sean aplicables en lo general, de modo que puedan volverse a usar". (Pressman, 2010)

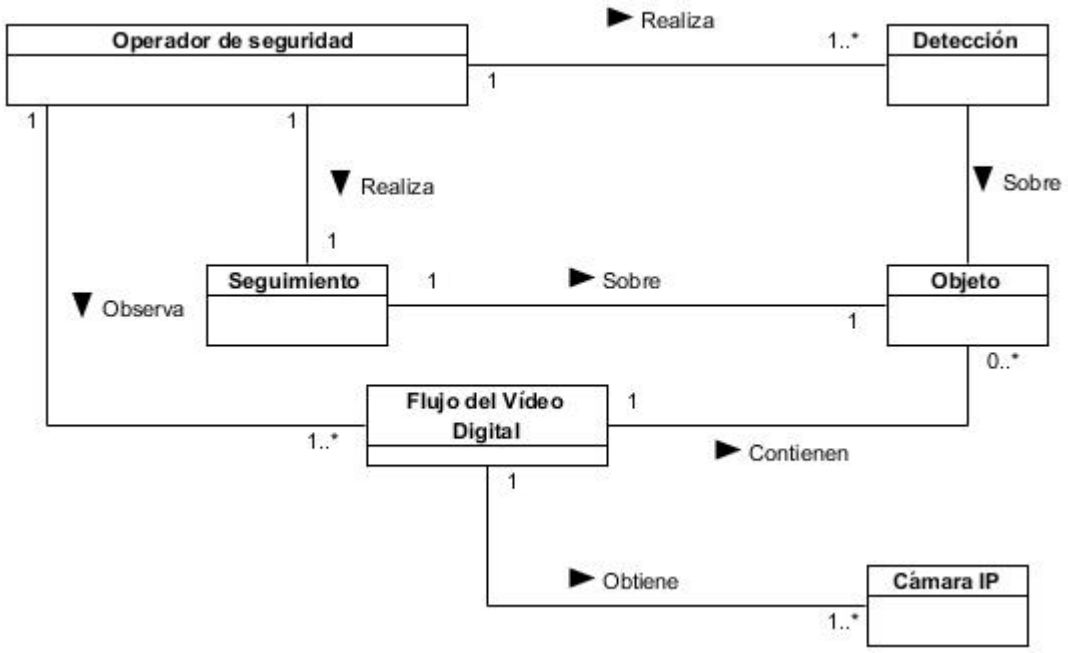

Figura 1: Modelo de dominio

2.2.1 Descripción del Modelo de Dominio

Cámara IP: Es a una cámara que obtiene los videos digitales y los envía directamente por una red hacia un ordenador para que sea observada.

Flujo de Video Digital: Es un flujo de información en formato digital que está formado por un conjunto de imágenes digitales que pueden contener distintos objetos.

Objeto: Son las figuras que se observan en el flujo de video digital.

Detección: Proceso de detección de personas realizado visualmente por el operador de seguridad en su vigilancia.

Seguimiento: Proceso de seguimiento de personas realizado visualmente por el operador de seguridad después de haber detectado una persona de interés.

Operador de Seguridad: Persona encargada de vigilar visualmente las áreas de interés.

2.3 Especificación de los requisitos de software

A lo largo del proceso de desarrollo de software, lograr un entendimiento productivo entre clientes y el equipo de proyecto es de vital importancia. Para lograr esto se deben establecer los objetivos del producto. Con el objetivo de llegar a un consenso entre ambas partes involucradas, se establecen requisitos, estos pueden ser: funcionales y no funcionales. El cumplimiento de los mismos es directamente proporcional al resultado final. (Pressman, 2010)

2.3.1 Requisitos Funcionales

Los Requerimientos Funcionales son condiciones o capacidades que el sistema debe cumplir, suficientemente buenas como para llegar a un acuerdo entre los clientes (incluyendo usuarios) sobre qué debe y qué no debe hacer el sistema. (Jacobson, 2000)

A continuación, se muestran los requerimientos funcionales (RF) del sistema:

RF 1: Recibir flujo de video

Descripción: El sistema debe recibir un flujo de video proveniente de una cámara IP.

RF 2: Detectar persona en el video.

Descripción: El sistema debe detectar si en el video existen una o varias personas.

RF 3: Escoger persona en el video.

Descripción: El sistema debe permitir al usuario escoger la persona que desea seguir.

RF 4: Seguir persona en el video.

Descripción: El sistema debe realizar el seguimiento de la persona detectada que ha sido escogida por el usuario.

RF 5: Emitir notificación.

Descripción: Se envía un mensaje mientras se esté siguiendo la persona.

2.3.2 Requisitos no funcionales

Los requerimientos no funcionales son propiedades o cualidades que el producto debe tener, además son aspectos importantes que el producto debe cumplir para lograr un producto atractivo, usable, rápido o confiable. (Pressman, 2010)

A continuación, se muestran los requerimientos no funcionales del sistema:

RNF 1. Rendimiento

RNF 1.1 Se debe procesar el video en tiempo real.

RNF 2. Usabilidad

RNF 2.1 El componente debe describir con claridad los parámetros de configuración. En caso de no recibir los parámetros correctos se muestra un mensaje de ayuda explicando la forma de introducir los mismos.

RNF 3 Portabilidad

RNF 3.1 El componente debe poder ser utilizado sobre diferentes plataformas como Windows, Linux y Mac OS. La biblioteca OpenCV 2.4.13 y el lenguaje C++ en su estándar 11 son multiplataforma, por lo que se logra la portabilidad compilando el componente en la plataforma que se vaya a utilizar.

RNF 3.2 La plataforma debe ser de una arquitectura de 64 bits. Las bibliotecas de OpenCV fueron compiladas en un sistema operativo con una arquitectura de 64 bits, por lo que no se asegura que el uso del componente en una arquitectura de 32 bits sea factible, debido a que pueden existir dependencias con la arquitectura de compilación.

RNF 4 Hardware

RNF 4.1 Se debe tener un equipo con 8Gb de RAM y un procesador Intel Core I7. Si no se tiene esos requisitos mínimos de hardware la imagen se “pixela” y el resultado del algoritmo no es el correcto. Esto se demostró realizando pruebas en computadoras con menos prestaciones que las mínimas descritas.

RNF 5 Restricciones del diseño

RNF 5.1 Lenguaje: Se utilizará para el desarrollo del sistema el lenguaje C++ en su estándar 11.

RNF 5.2 Biblioteca: Se utilizará la biblioteca OpenCV 2.4.13.

2.4 Definición de los casos de uso.

Según (Pressman, 2010) un caso de uso cuenta una historia estilizada sobre cómo interactúa un usuario final con el sistema en circunstancias específicas. La historia puede ser un texto narrativo, un lineamiento de tareas o interacciones, una descripción basada en un formato o una representación diagramática. Sin importar su forma un caso de uso ilustra el software o sistema desde el punto de vista del usuario final.

2.4.1 Descripción de los actores.

Los actores son las distintas personas (o dispositivos) que usan el sistema o producto en el contexto de la función y comportamiento que va a describirse. Los actores representan los papeles que desempeñan las personas (o dispositivos) cuando opera el sistema. Con una definición más formal, un actor es cualquier cosa que se comunique con el sistema o producto y que sea externo a éste. Todo actor tiene uno o más objetivos cuando utiliza el sistema. (Pressman, 2010)

Tabla 1: Actores del sistema

Actor	Descripción
Usuario	Es un usuario autorizado a activar el video sensor en el sistema de video vigilancia Xilema Suria en su versión 3.0.

2.4.2 Diagrama de Casos de Uso del Sistema

En la Figura 2 se muestra el diagrama de casos de usos del sistema.

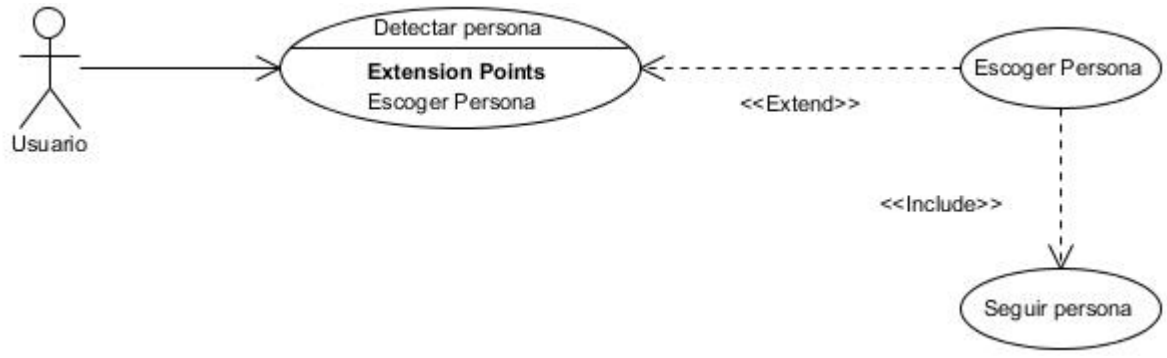

Figura 2: Diagrama de Casos de Uso del Sistema

2.4.3 Especificación de Casos de Uso.

En el presente epígrafe se presenta la descripción textual del CU Detectar persona.

Tabla 2: Descripción del CU Detectar Persona

CU	Detectar persona.	
Objetivo	El objetivo de este CU es permitir la detección de personas en un flujo de video.	
Actores	Usuario.	
Resumen	El caso de uso se inicia una vez que el usuario activa el video sensor y se recibe el flujo emitido por una cámara IP, en el mismo instante que se recibe el flujo de video se comienza a detectar las personas. El CU termina cuando se termine el flujo de video o cuando el usuario desactive el video sensor.	
Complejidad	Alta.	
Prioridad	Alta.	
Precondiciones	Una flujo de video.	
Postcondiciones	No procede.	
Flujo de eventos		
Flujo básico: Detectar persona.		
	Actor	**Sistema**
1	El usuario activa el video sensor.	
2		Se comienza a recibir el flujo de video proveniente de la cámara IP.
3.		Se buscan las características de una persona en el flujo de video.
4		Se resaltan las áreas de las personas existentes en el flujo de video.
5		Termina el CU cuando se termine el flujo de video o cuando el usuario desactive el video sensor.
Flujos alternos.		
2a.Evento: No se recibe flujo de video.		
	Actor	**Sistema**
2a		En la cámara se verá la pantalla de color negro
Flujos alternos.		

3a.Evento: No se encuentran características de una persona.		
	Actor	**Sistema**
3a		En la cámara no se verá ningún área resaltada.
Relaciones	**CU Incluidos**	No procede
	CU Extendidos	Escoger persona
Referencias	RF1,RF2, RnF1, RnF2,RnF3, RnF4, RnF5,	

2.5 Modelo del diseño

El modelo de diseño es un modelo físico y concreto, da forma al sistema y debe ser mantenido durante todo el desarrollo del ciclo de vida del software (Pressman, 2010). Se centra en la realización de los casos de uso a través de los requisitos, tanto funcionales como no funcionales. La entrada esencial a este modelo es el resultado del modelo de análisis, ya que este resultado proporciona una comprensión detallada de los requisitos. El objetivo final del flujo de trabajo de diseño es producir un modelo lógico del sistema a implementar.

2.5.1 Diagrama de clases del diseño

Este tipo de diagrama representa la realización física de los casos de usos, centrándose en cómo los requisitos funcionales y no funcionales tienen impacto en el sistema. Aporta una visión estática o de estructura de un sistema, sin mostrar la naturaleza dinámica de las comunicaciones entre los objetos de las clases (Pressman, 2010). Describe gráficamente las especificaciones de las clases de software y las interfaces y contiene las definiciones de las entidades del software en vez de conceptos del mundo real.

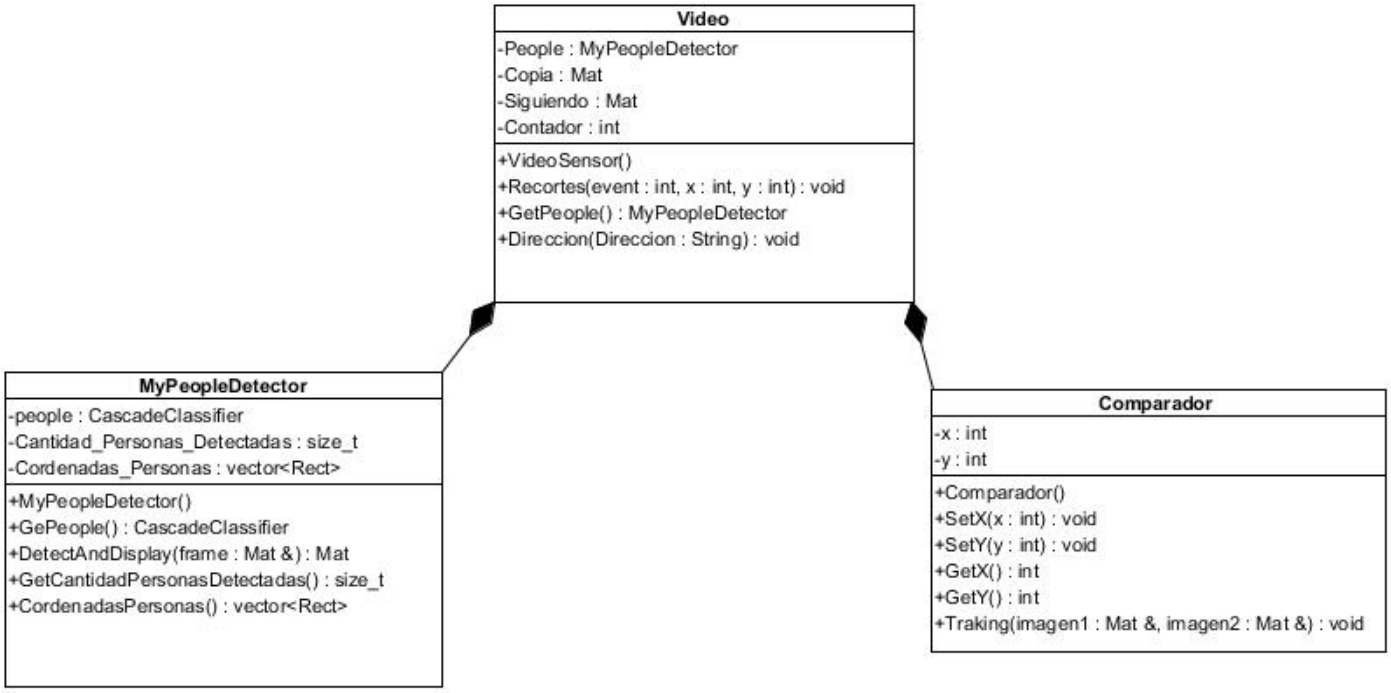

Figura 3: Diagrama de Clases del diseño propuesto

En la Figura 3 se muestra el diagrama de clases del diseño para el video sensor para la detección y seguimiento de personas. Aquí intervienen las clases siguientes:

VideoSensor: permite dividir un video digital en fotogramas y contiene toda la información necesaria para el trabajo con video digital ya que contiene los métodos: “VideoSensor” que es su constructor, “Recortes” que permite que el usuario escoja la persona que quiere seguir, “Direccion” es el método principal que recibe la dirección del flujo de video y “GetPeople” que devuelve el objeto de tipo MyPeopleDetector.

MyPeopleDetector: es la encargada de procesar el fotograma obtenido por el VideoSensor para obtener y dibujar las áreas donde se encuentren las personas detectadas. Posee los métodos “MyPeopleDetector” que es su constructor, “GetPeople” que devuelve un objeto de tipo MyPeopleDetector, “DetectAndDisplay” es el método que detecta las personas y representa su ubicación mediante un dibujo rectangular y “CordenadasPersonas” que devuelve la lista de todas las coordenadas de las personas detectadas.

La clase Comparador es la encargada de rastrear las personas y mostrar su rastro. Posee los métodos “Comparador” que es su constructor, “SetX”, se utiliza para actualizar la coordenada x, “SetY” se utiliza para actualizar la coordenada y, “GetX” se utiliza para obtener la coordenada x, “GetY” se utiliza para obtener la coordenada y, por ultimo posee “Tracking” que es el método que rastreará a la persona y dejará su marca por donde pase.

2.6 Arquitectura y patrones

2.6.1 Descripción de la arquitectura.

La arquitectura del software de un programa o sistema de cómputo es la estructura o estructuras del sistema, lo que comprende a los componentes de software, sus propiedades externas visibles y las relaciones entre ellos. Es decir, es una representación que permite analizar la efectividad del diseño para cumplir los requerimientos establecidos, considerar alternativas arquitectónicas en una etapa en la que hacer cambios al diseño todavía es relativamente fácil y reducir los riesgos asociados a la construcción del software. (Pressman, 2010).

El video sensor desarrollado, es un sistema con una arquitectura de flujo de datos. Esta se aplica cuando datos de entrada van a transformarse en datos de salida a través de una serie de componentes computacionales o manipuladores y está basada en el patrón de tuberías y filtros que está formado por un conjunto de componentes llamados filtros, conectados por tubos que transmiten los datos de un componente al siguiente.

Cada filtro trabaja de forma independiente de aquellos componentes que se localizan arriba o abajo del flujo, se diseñan para esperar una entrada de datos de cierta forma y produce datos de salida (al filtro siguiente) en una forma especificada. Sin embargo, el filtro no requiere ningún conocimiento de los trabajos que realizan los filtros vecinos.

En la figura 4 se muestra una representación de la arquitectura del video sensor para detección y seguimiento de personas basada en el patrón tuberías y filtros. Se muestra cada filtro y su funcionalidad, en la que los datos de entrada van siendo modificados por las actividades de cada filtro hasta la salida.

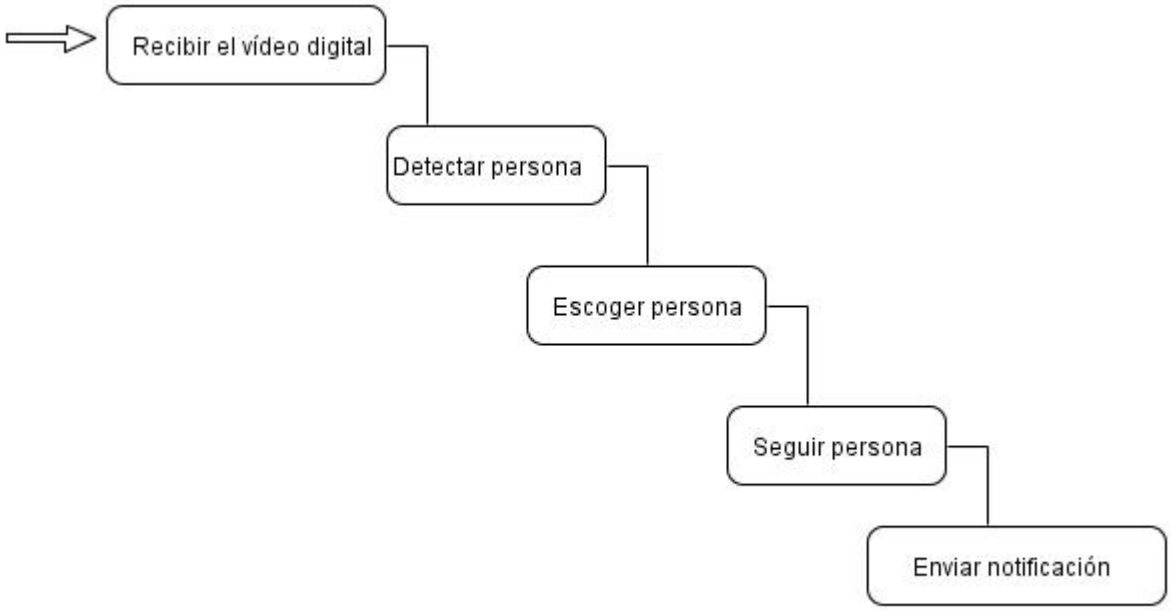

Figura 4: Arquitectura del video sensor

2.6.2 Patrones de Diseño

Los patrones de diseño brindan una solución ya probada y documentada a problemas de desarrollo de software que están sujetos a contextos similares. Es necesario tener presente los siguientes elementos de un patrón: su nombre, el problema (cuando aplicar un patrón), la solución (descripción abstracta del problema) y las consecuencias (costos y beneficios).Entre los patrones de diseño más utilizados se encuentran los patrones GRASP y los patrones GoF. (Mestras, 2004)

Los patrones GRASP (*General Responsibility Assignment Software Patterns*) y GoF (*Gang of Four*) son patrones generales para asignar responsabilidades y diseñar con éxito el software orientado a objetos. Para el desarrollo del video sensor se utilizaron los siguientes patrones pertenecientes a ese grupo:

Experto: Es un patrón para asignar responsabilidades; es un principio básico que suele utilizarse en el diseño orientado a objetos. Expresa simplemente que los objetos realizan operaciones relacionadas con la información que poseen. Esto se observa en la clase “MyPeopleDetector” que es una clase experta en el manejo de sus datos.

Creador: Plantea la necesidad de asignarle a una clase la responsabilidad de crear una instancia de otra clase siempre y cuando agregue los objetos de la clase, los contenga, registre las instancias de estos objetos y los utilice específicamente. En el video sensor desarrollado en la presente investigación esto está presente en la clase “VideoSensor”.

Controlador: Se hace ventajoso llevar el control de la aplicación en una sola clase, esa es la idea del patrón controlador que se evidencia en la clase VideoSensor, ya que desde ella se controlan todas las clases del video sensor. Es un objeto en particular, en una aplicación, que lleva el manejo de la misma. También se utilizan los patrones **Alta Cohesión** y **Bajo Acoplamiento**. El primero es la idea de tener las clases lo menos ligadas entre sí. Así, si se produce una modificación en alguna de ellas, se tenga la mínima repercusión posible en el resto de clases, fortaleciendo la reutilización, y disminuyendo la dependencia entre clases. El segundo indica que la información que contiene una clase debe de ser coherente y esté relacionada con la clase, esto facilita el cambio. Al realizar un cambio en una clase muy cohesionada, todos los métodos que se vean afectados estarán a la vista, en la misma clase.

Prototipo: Se evidencia cuando se crea nuevos objetos clonándolos de una instancia ya existente como se ve en la clase “VideoSensor”

2.7 Conclusiones Parciales

Al concluir este capítulo se representó el modelo de dominio del sistema, que permitió conocer los principales conceptos asociados al desarrollo de la aplicación y las interacciones entre estos. Se especificaron los requisitos funcionales y no funcionales que permiten lograr un entendimiento productivo entre clientes y el desarrollador del video sensor.

El diagrama de casos de uso del sistema permite entender cómo funcionará el sistema de manera global y engloba las funcionalidades que tendrá el software a desarrollar. Las clases del diseño específica las relaciones entre las clases y métodos en un nivel más detallado. La arquitectura definida guía el funcionamiento y el tratamiento que se le dará a los datos que serán procesados en el desarrollo de la aplicación.

Es importante destacar el uso de patrones GRASP y GoF que posibilita la calidad del código fuente, ya que esto permite definir correctamente la asignación de responsabilidades para lograr un alto grado de cohesión con un menor grado de acoplamiento.

Capítulo 3: Implementación y pruebas.

En el presente capítulo se muestra el modelo de implementación como resultado del diseño anteriormente desarrollado. Se describen las pruebas a realizar, con el objetivo de comprobar las funcionalidades, para de esta forma verificar en todos los casos que los resultados de las pruebas sean los esperados y comprobar el correcto funcionamiento del sistema. También, se muestra el diagrama de despliegue para describir el ambiente dentro del cual el sensor será instalado.

3.1 Algoritmos

Los algoritmos utilizados para la implementación del video sensor para detección y seguimiento fueron Viola and Jones en la etapa de detección y en la etapa de seguimiento se utilizó un procedimiento que busca las distancias entre todas las personas que hay en el próximo fotograma y la persona elegida por el usuario para así saber dónde se encuentra la misma con el pasar del tiempo. A continuación, se describen estos algoritmos detalladamente.

3.1.1 Viola and Jones

El detector de objetos utilizado inicialmente para la detección del rostro, fue propuesto por Paul Viola (Viola, y otros, 2001) y mejorado por Rainer Lienhart (Lienhart, y otros, 2003). OpenCV refiere a este detector como “Clasificador en Cascada” y viene con un conjunto de archivos pre-entrenados para la detección. Un clasificador es entrenado con cientos de vistas de ejemplos de un objeto en particular (por ejemplo, personas, carros, aviones) llamados ejemplos positivos, que son escalados al mismo tamaño junto con los ejemplos negativos, que son imágenes arbitrarias que no son las que se quieren clasificar. Después de ser entrenado un clasificador, este puede ser aplicado a una región de interés de una imagen de entrada. El clasificador retorna 1 si la región clasifica como el objeto que se quiere reconocer, si no retorna 0.

El clasificador se dice es en cascada ya que el resultado consiste en varios clasificadores simples que son aplicados subsecuentemente a una región de interés, hasta que en algún estado el candidato es rechazado o todos los estados son pasados. El clasificador es “*boosted*” ya que cada estado es complejo y son construidos a partir de clasificadores básicos usando uno de cuatro diferentes técnicas de *boosting* (*Discrete Adaboost, Real Adaboost, Gentle Adaboost y Logitboost*) (Kejriwal, y otros, 2015). La técnica utilizada en esta investigación es *Gentle Adaboost* que según (Nock, y otros, 2014) se acerca más rápido a mejores soluciones que las otras mencionadas. Los clasificadores básicos son basados en árboles de decisión con

al menos dos salidas. Las características basadas en *Haar* son la entrada a los clasificadores básicos y son calculadas como se describe a continuación.

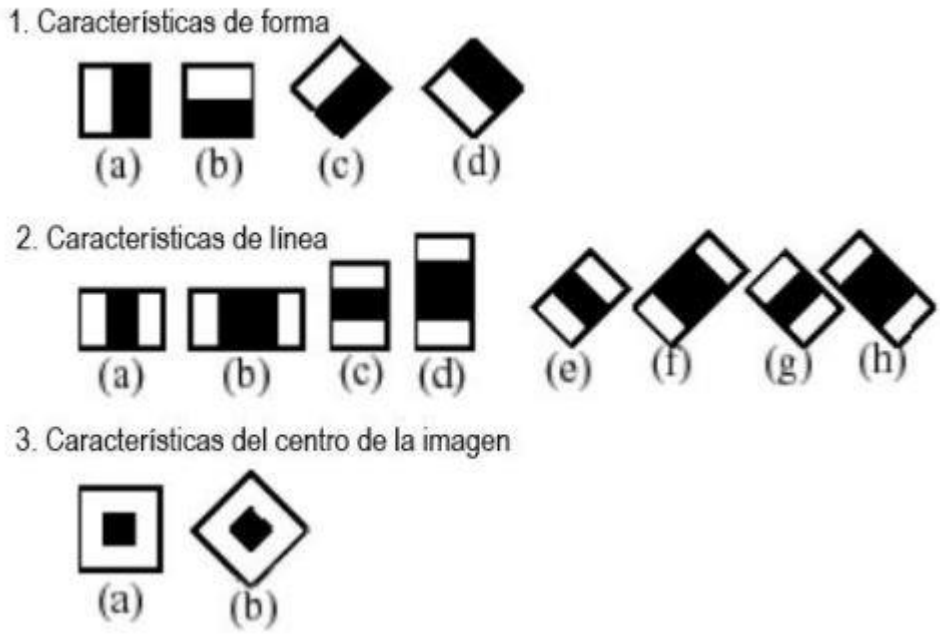

Figura 5: Características de Haar que utiliza el algoritmo *(Lienhart, y otros, 2003)*.

Las características usadas en un clasificador particular se especifican por su forma (1a, 2b, etc), posición dentro de la región de interés y la escala. Por ejemplo, en el caso de la tercera característica de línea (2c) la respuesta es calculada por la diferencia entre la suma de los píxeles de la imagen según el rectángulo que cubre toda la característica (incluyendo las dos líneas blancas y la negra en el medio) y la suma de los píxeles de la imagen dentro de la línea negra multiplicada por tres para compensar la diferencia de áreas.

3.1.2 Entrenamiento de Haar Cascade

Para la realización del entrenamiento del *Haar cascade* se utilizó la base de datos “INRIAperson” encontrada en (Dalal, 2005) de 614 ejemplos de imágenes digitales positivas y 1218 ejemplos de imágenes digitales negativas. El primer paso fue utilizar la herramienta opencv_annotation.exe que está ubicada en la carpeta de *opencv\build\x64\vc12\bin* para segmentar los objetos en el que se desea entrenar el Haar cascade. En este caso los objetos en el que se entrenó fueron personas, como se muestra en la Figura 6.

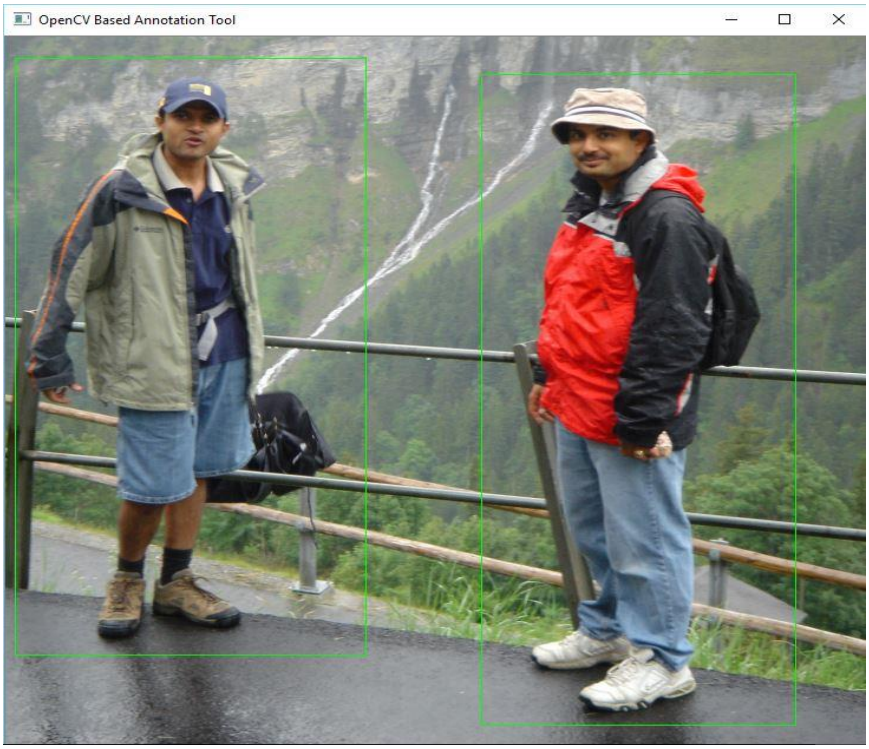

Figura 6: Segmentación de personas

Como se observa en la Figura 7, el paso anterior genera un documento de texto que posee la siguiente estructura: dirección de la foto usada, cantidad de segmentaciones y consecutivamente las coordenadas del rectángulo de las segmentaciones.

```
pos\crop001001.png 2 301 111 206 583 33 324 187 371
pos\crop001002.png 3 995 59 166 530 863 65 298 524 108 84 179 518
pos\crop001003.png 1 315 69 342 625
```

Figura 7: Formato documento generado por la herramienta opencv_annotation.exe

En el segundo paso realizado se utilizó la herramienta *opencv_createsamples.exe*, que se puede encontrar en la misma ubicación, para generar un archivo de extensión “nombre.vec” que se utiliza posteriormente. Para la creación de este archivo se usó la siguiente línea de código en la herramienta:

$$opencv\_createsamples.exe\ -info\ positive/people.txt\ -vec\ vector/people.vec - num\ 614\ -w\ 24\ -h\ 24$$

Como se ve en la línea de código anterior, a la herramienta se le dio como parámetros: la ubicación del primer archivo que se generó con las coordenadas de las personas segmentadas, después se le da la

ubicación donde generará el vector de coordenadas y posteriormente se le indica la cantidad de imágenes positivas con la resolución a que debe escalar dichas imágenes.

La última herramienta que se utilizó fue *opencv_haartraining.exe* que es la herramienta que genera el Haar cascade, el cual es un archivo de extensión *xml*. Para la creación de este archivo se utilizó la siguiente línea de código:

$$\text{opencv\_haartraining.exe} -\text{data MypeopleDetector} -\text{vec vector/people.vec} -\text{bg negative/bg.txt} -\text{npos } 500 -\text{nneg } 1218 -\text{nstages } 20 -\text{mem } 1024 -\text{mode ALL} -\text{w } 24 -\text{h } 24 -\text{nonsym}$$

Como se observa en la línea de código anterior, a la herramienta se le dio como parámetros: el nombre que tendrá el *Haar cascade* generado, la ubicación del vector generado en el segundo paso, la ubicación de las imágenes negativas, la cantidad de imágenes positivas con las que fue creado el vector, la cantidad de imágenes negativas que se encuentran en la ubicación, cantidad de capas o clasificadores independientes que formarán al *Haar cascade*, la cantidad de RAM (*Random Acess Memory*) dedicada al proceso (en este caso fue de 1024MB), la resolución a que serán escaladas las imágenes, el modo ALL para que no se detenga hasta que no use todas las imágenes pasadas por parámetros y $-nonsym$ que se usa si el objeto no es horizontalmente simétrico, como es el caso de las personas.

La duración de este entrenamiento duró aproximadamente 96 horas y como se puede ver en la Figura 8 se detectaron verdaderos positivos después del entrenamiento.

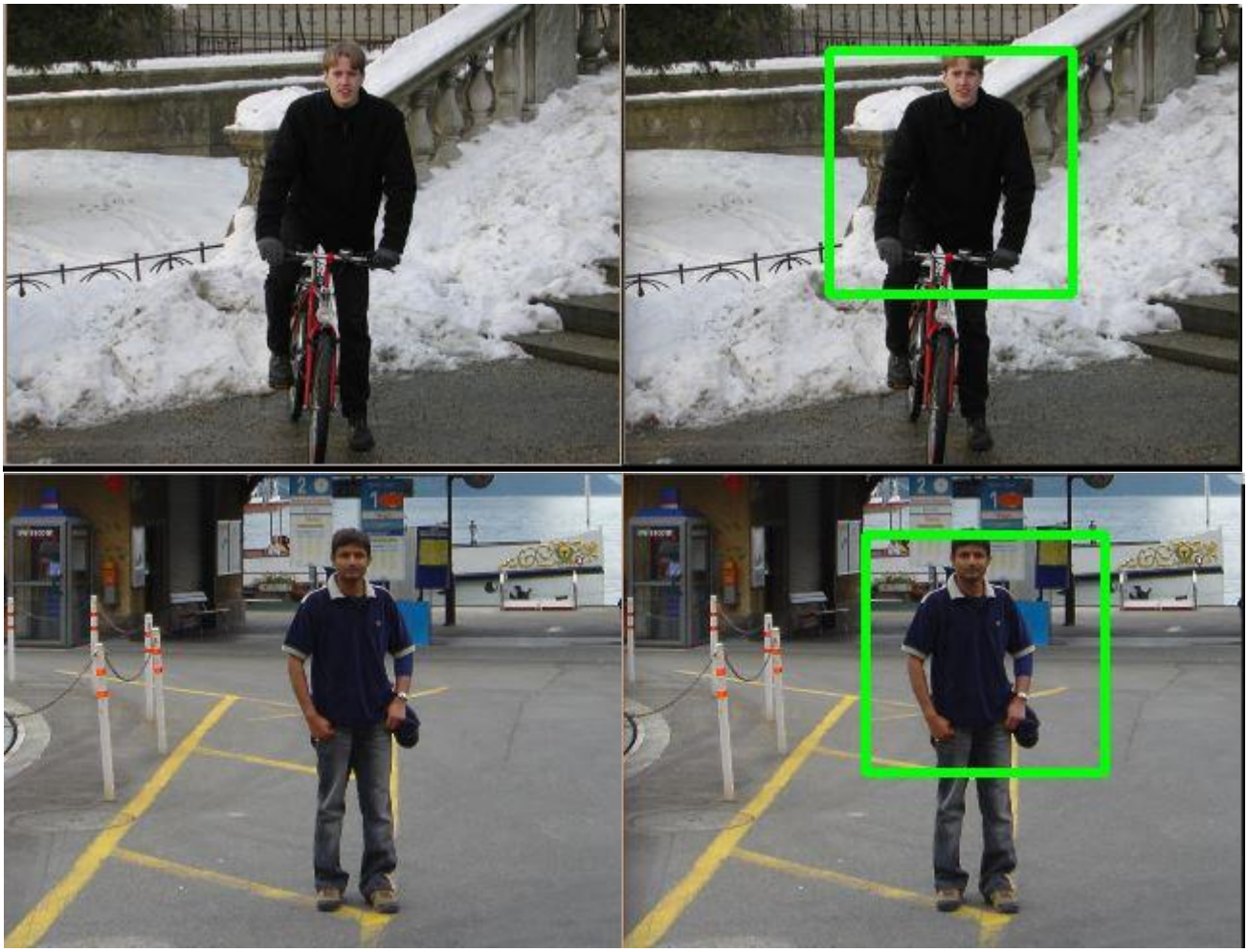

Figura 8: Detección de ciclista y persona de pie estática

Pero también, como se ve en la Figura 9, detecta falsos positivos.

Figura 9: Falso positivo donde se detecta una señal de tránsito como persona

3.1.3 Descripción del algoritmo de seguimiento de persona detectadas

El proceso de seguimiento es iniciado una vez que se escoge una persona. Lo siguiente que se realiza es una extracción de los datos de la persona a seguir, con los cuales se calculan el área del rectángulo envolvente que ocupa esta persona en el fotograma y las coordenadas del punto de intersección de las diagonales de este rectángulo. El área de un rectángulo A_r se calcula con la ecuación:

$$A_r = W_r * H_r$$

En donde W_r es el ancho del rectángulo envolvente y H_r es la altura del rectángulo envolvente.

El punto medio $\mathrm{C_r}(v_1, v_2)$ del rectángulo envolvente de la persona se calcula mediante la fórmula:

$$\mathrm{C_r}(v_1, v_2) = \left(\frac{\mathrm{x_1}+\mathrm{x_2}}{2}, \frac{y_1+y_2}{2}\right)$$

Donde (x_1, y_1) y (x_2, y_2) son los puntos que se muestran en la Figura 10.

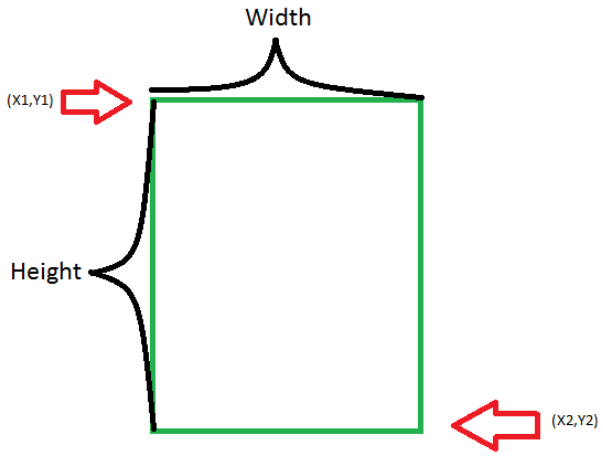

Figura 10: Representación del rectángulo envolvente de la persona.

x_2 y y_2 se calculan de la siguiente forma:

$$\mathrm{x_2} = \mathrm{W_r} + \mathrm{x_1}$$

$$y_2 = H_r + y_1$$

Cuando se analiza un fotograma, se extraen estas características a todas las personas detectadas, es decir se calcula todas las áreas y puntos medios de las personas que el detector identificó. Con estos datos se procede a calcular las distancias euclidianas de donde se encontraba la persona anteriormente a donde se encuentran en el fotograma que se está analizando, ya que la persona que se desea rastrear estará a la

mínima distancia y en caso de alguna coincidencia de las distancias se usará la correspondencia del área del rectángulo envolvente de la persona actual con la anterior. Para esto se define un umbral de correspondencia entre las áreas para que la diferencia no sea muy grande.

La fórmula de la distancia euclidiana $D(P_1, P_2)$ que se utilizó fue para un espacio bidimensional, esta está definida por:

$$D(P_1, P_2) = \sqrt{(V_{x_2} - V_{x_1})^2 + (V_{y_2} - V_{y_1}))^2}$$

Donde el par (V_{x_1}, V_{y_1}) es la coordenada del punto medio en la imagen anterior del rectángulo envolvente de la persona que se está siguiendo, y (V_{x_2}, V_{y_2}) define el punto medio de la persona con que se está comparando la distancia actualmente. Esto se realiza para cada persona detectada en la imagen que se está analizando para determinar cuál es la más cercana a la persona que se está siguiendo.

3.1.4 Descripción del video sensor

El video sensor lo primero que recibirá es una dirección de un flujo de video proveniente de una cámara IP. Posteriormente se cargará el *Haar cascade* "MypeopleDetector.xml" y se comenzará a capturar los fotogramas provenientes de flujo del video digital. Después para el mejor rendimiento del video sensor si es necesario, se escalará los fotogramas a una resolución menor y se empezará a buscar personas en cada fotograma capturado del flujo de video (Figura 11).

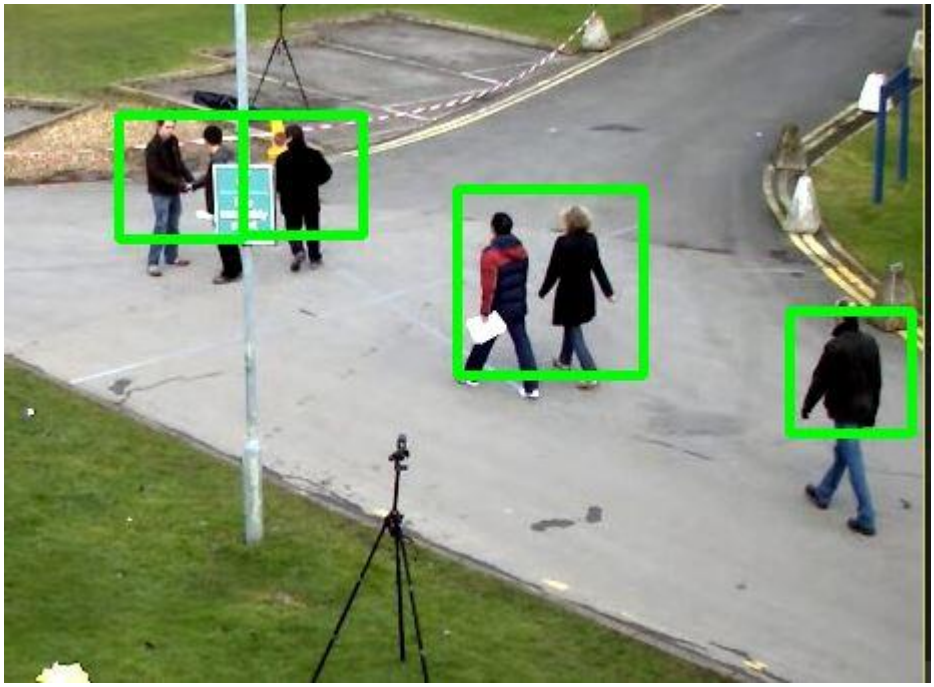

Figura 11: Proceso de detección de personas

Si en algún momento el usuario desea seguir una persona solo tendrá que dar un clic en el lugar donde se esté detectando la persona que desea seguir. Posteriormente se verá en la ventana (Figura12) una línea roja con la trayectoria de la persona que se está siguiendo.

Figura 12: Proceso de seguimiento de una persona

Mientras se esté siguiendo a la persona se mostrará un mensaje “Siguiendo” y en el instante que se pierda o se detenga el seguimiento se mostrará el mensaje “Seguimiento detenido”.

3.2 Modelo de implementación

Es comprendido por un conjunto de componentes y subsistemas que constituyen la composición física de la implementación del sistema. Entre los componentes se puede encontrar la biblioteca OpenCV con sus módulos y el código fuente. Fundamentalmente se describe la relación que existe entre ellos.

3.2.1 Diagrama de componentes

Un componente es una parte modular, desplegable y reemplazable de un sistema que encapsula la implementación, puede exponer un conjunto de interfaces y proporciona la realización de los mismos. Son las piezas reutilizables de alto nivel a partir de las cuales se pueden construir los sistemas. Es por esto que, los diagramas de componentes son empleados para describir la estructura física del modelo de implementación del sistema y mostrar las relaciones entre sus elementos, reflejando de este modo, una estructura de alto nivel. Representan las dependencias entre componentes de software incluyendo componentes de código fuente, componentes del código binario, y componentes ejecutables. Se rigen por la filosofía de la alta cohesión interna y el bajo acoplamiento externo (Rumbaugh, y otros, 2009).

En la Figura 13 se muestra el diagrama de componentes de esta investigación. Nótese la relación entre las bibliotecas de los módulos internos que se utilizaron de OpenCV con los componentes de código fuente desarrollados en este trabajo.

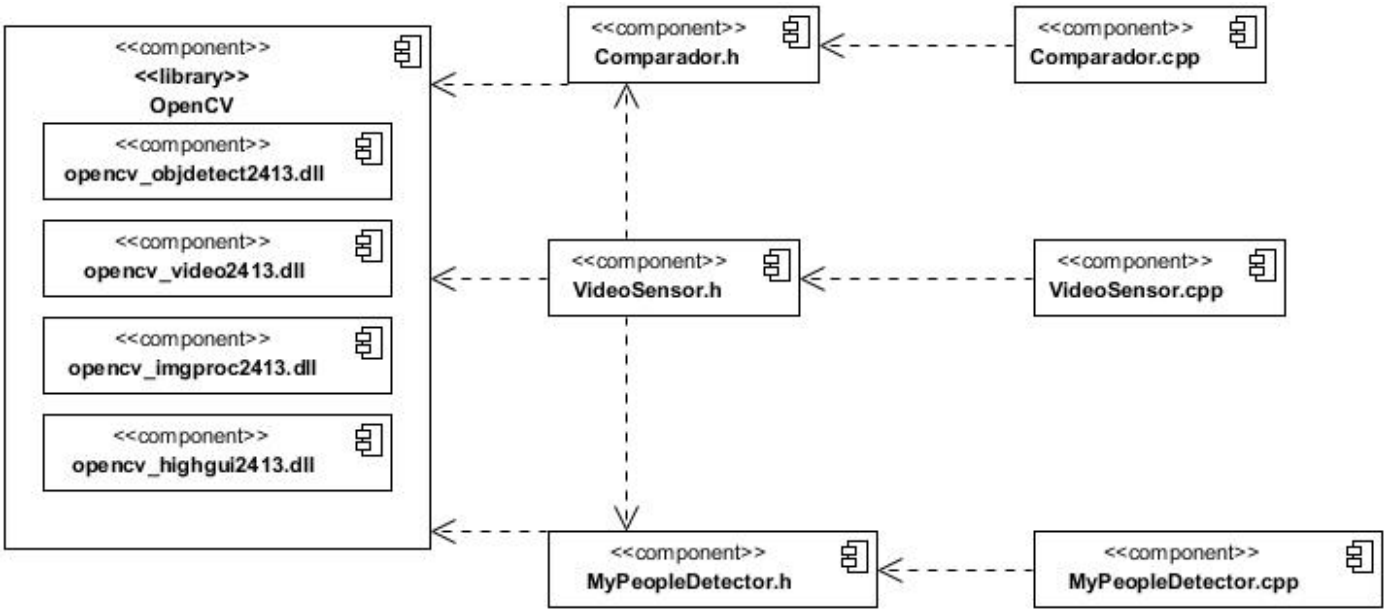

Figura 13: Diagrama de componentes de código fuente

Descripción del diagrama de componentes

El sistema cuenta con los componentes: OpenCV, Comparador.h, VideoSensor.h, MyPeopleDetector.h, Comparador.cpp, VideoSensor.cpp, MyPeopleDetector.cpp que se encargan de distribuir e implementar las funcionalidades necesarias para la realización del video sensor.

OpenCV: En esta biblioteca, que como se ha mencionado anteriormente se utiliza para el procesamiento de imágenes digitales, cada módulo posee una biblioteca que se debe referenciar para utilizar sus funcionalidades. El nombre de cada biblioteca de OpenCV referencia a la funcionalidad del ensamblado y a la versión de OpenCV a la que pertenece. En la Figura 13 se observa el uso de **opencv_objdetect2413.dll** para la detección de objetos, **opencv_video2413.dll** para el trabajo de captura y descomposición del video en fotogramas, **opencv_imgproc2413.dll** para el procesamiento de los fotogramas y **opencv_highgui2413.dll** para mostrar el resultado del procesamiento.

Comparador.h, VideoSensor.h y MyPeopleDetector.h: Tienen como objetivo declarar todas las funcionalidades del componente para posterior implementación.

Comparador.cpp, VideoSensor.cpp y MyPeopleDetector.cpp: Tienen como objetivo implementar todas las funcionalidades de los componentes anteriormente declarados.

3.2 Diagrama de Despliegue

Describe el ambiente dentro del cual el sistema será instalado. Establece una correspondencia entre la arquitectura de software y la arquitectura de hardware del sistema, por lo que el rol encargado de realizarlo es el Arquitecto de Software. Muestran a los nodos procesadores la distribución de los procesos y de los componentes (Larman, 2010). Para el video sensor propuesto se cuenta con dos nodos principales: una computadora en la que se estará ejecutando la aplicación y un dispositivo, en este caso una cámara IP, capturando el flujo de video. Estos nodos se comunican mediante el Protocolo de Control de Transmisión/Protocolo de Internet (TCP/IP). Este protocolo representa todas las reglas de comunicación para Internet y se basa en la noción de dirección IP, es decir, en la idea de brindar una dirección IP a cada equipo de la red para poder enrutar paquetes de datos.

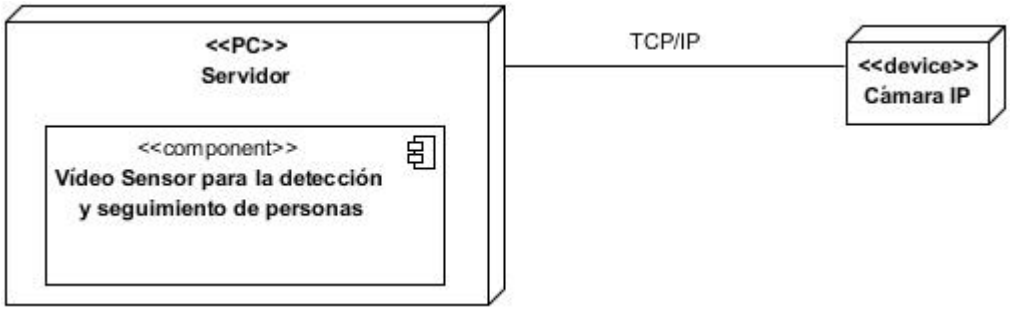

Figura 14: Diagrama de despliegue

3.3 Pruebas del software

Las pruebas son un conjunto de actividades que se pueden planificar por adelantado y llevar a cabo sistemáticamente. Una estrategia de prueba del software debe incluir pruebas de bajo nivel que verifiquen que todos los pequeños segmentos de código fuente se han implementado correctamente, así como pruebas de alto nivel que validen las principales funciones del sistema frente a los requisitos del cliente. De esta forma, el proceso de pruebas se centra en los procesos lógicos internos del software, asegurando que todas las sentencias se han comprobado, y en los procesos externos funcionales para la detección de errores. Además de ser utilizadas para identificar posibles fallos de implementación, calidad o usabilidad de un programa (Pressman, 2010)

3.3.1 Prueba de caja blanca

Las pruebas de caja blanca, en ocasiones llamada prueba de caja de vidrio, según (Pressman, 2010) es una filosofía de diseño de casos de prueba que se usa la estructura de control descrita como parte del diseño a nivel de componentes para derivar casos de prueba que: 1) garanticen que todas las rutas independientes dentro de un módulo se revisaron al menos una vez , 2) revisen todas las decisiones lógicas en sus lados verdadero y falso, 3) ejecuten todos los bucles en sus fronteras y dentro de sus fronteras operativas y 4) revisen estructuras de datos internas para garantizar su validez.

Como parte de este tipo de prueba, se puede utilizar la técnica de camino básico, para lo cual es necesario conocer el número de caminos independientes de un determinado algoritmo mediante el cálculo de la complejidad ciclomática. Este se realiza de tres formas diferentes:

- El número de regiones del grafo de flujo coincide con la complejidad ciclomática.
- La complejidad ciclomática $V(G)$ de un grafo de flujo G se define como: $V(G) = A - N + 2$ donde A es el número de aristas del grafo de flujo y N es el número de nodos del mismo.
- $V(G)$ también se calcula como el resultado de $P + 1$ donde P es el número de nodos predicados (nodos de los cuales parten dos o más aristas) que tiene contenido el grafo de flujo G.

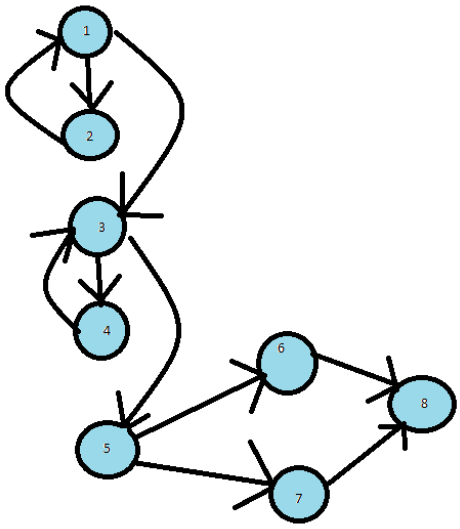

Figura 15: Grafo de flujo asociado a la funcionalidad Rastros

Resultado del cálculo de la complejidad ciclomática:

$$\boldsymbol{V(G)} = A - N + 2 = 10 - 8 + 2 = 4$$

$$\boldsymbol{V(G)} = P + 1 = 3 + 1 = 4$$

A partir del resultado obtenido se determina que la funcionalidad presenta una complejidad ciclomática de 4, lo que deriva que existe a lo sumo 4 caminos lógicos por donde ejecutarse dicha funcionalidad. En la Tabla 3 se muestran los caminos básicos.

Tabla 3: Caminos básicos.

No.	Camino básico
1	1-2-3-4-5-6-8
2	1-2-3-4-5-7-8
3	1-3-4-5-6-8
4	1-3-4-5-7-8

3.3.2 Prueba de rendimiento

Se aplican para descubrir problemas, debido a la falta de recursos del lado del servidor, ancho de banda de red inapropiado, poca capacidad en la base de datos o en el sistema operativo, funcionalidades mal diseñadas y otros conflictos de hardware o software que puedan inducir a un bajo desempeño (Pressman, 2008).

3.3.2.1 Banco de videos de pruebas

Los videos utilizados para probar el algoritmo fueron grabados por el autor ya que no se encontró con una base de datos de videos de prueba que permitiera probar diferentes patrones de detección y seguimiento de personas. Los trabajos estudiados para esta investigación no hacen referencia a ninguna base de datos de videos digitales de detección y seguimiento para probar el algoritmo propuesto, por lo que se conformó una base de datos de videos digitales para probar el algoritmo.

Se grabaron escenas con cámaras fijas de locales vigilados por Xilema Suria 3.0, estos son de áreas exteriores e interiores. Para la realización de las pruebas las grabaciones realizadas se dividieron en dos escenarios que son:

▪**Escenario 1:** 24 grabaciones realizadas en exteriores.

▪**Escenario 2:** 10 grabaciones en interiores.

Los videos fueron grabados de 10 a 30 fotogramas por segundo (FPS) y ajustados a diferentes resoluciones, 320x240, 640x360, 640x480, 720 x 480, 768x576 y 1280x720. Esto permite calcular para las diferentes resoluciones, el tiempo medio que se tarda el procesamiento de cada fotograma para poder determinar la cantidad de imágenes que se pueden procesar en un segundo y verificar el rendimiento de los algoritmos en cuanto a resolución por FPS.

3.3.2.2 Medidas de evaluación de rendimiento

Una manera común de evaluar los resultados en los experimentos de visión por computador es con el uso de *Precision (P), Recall (R)* y el *factor-F* (W Powers, 2007). Para el caso de clasificación binaria se introducen medidas en el contexto de la clasificación binaria dicotómica, donde las etiquetas son por convención + y –, y las predicciones del clasificador se resumen en una tabla de contingencia de cuatro celdas (tabla 4). Cada celda toma los valores de:

- TP (*True Positive*): Total de eventos que son de detección y seguimiento de personas y son detectados como tal.
- FN (*False Negative*): Total de eventos que son de detección y seguimiento de personas, pero no son detectados como tal.
- TN (*True Negative*): Cantidad de eventos que no son de detección y seguimiento de personas y son detectados como tal.
- FP (*False positive*): Total de eventos que no son eventos de detección y seguimiento de personas y son detectados como tal.
- P+ (*Predicted Positive*): Se detecta como detección y seguimiento de personas.
- P- (*Predicted Negative*): No se detecta como detección y seguimiento de personas.
- R+ (*Real Positive*): Es un evento de detección y seguimiento de personas.
- R- (*Real Negative*): No es un evento de detección y seguimiento de personas.

Tabla 4: Notaciones de una tabla de contingencia binaria. Los códigos de color rosa y verde denotan los conteos correctos e incorrectos respectivamente *(W Powers, 2007)*.

	R+	R-
P+	TP	FP
P-	FN	TN

Para obtener los valores de *Recall* y *Precisión* se utilizan las ecuaciones:

$$\text{Recall} = \frac{TP}{TP + FN}$$

$$\text{Precision} = \frac{TP}{TP + FP}$$

Recall o *Sensitivity* (como se le llama en psicología) es la proporción de los casos R+ que son correctamente etiquetados como positivos. *Precision* o *Confidence* (como se le llama en minería de datos), denota la proporción de los casos etiquetados como positivos que son realmente positivos. Para el interés de tener

$$\mathrm{F} = \left[\frac{\alpha}{\mathrm{P}} + \frac{1-\alpha}{\mathrm{R}}\right]^{-1} = \frac{\mathrm{PR}}{(1-\alpha)\mathrm{P} + \alpha\mathrm{R}}, 0 \leq \alpha \leq 1$$

una sola medida que vincule a *P* y *R* se usa la medida *F* que es definida como la media ponderada armónica de *P* y *R.*

El valor más popular correspondiente a α es 0.5 y *F* se reduce a:

Ya que *F* es una figura de mérito, mientras más alto sea su valor mejor es el rendimiento del sistema. Se puede definir $E = 1 - F$ como la correspondiente medida de error.

$$F = \frac{2PR}{P + R}, \alpha = 0.5$$

3.3.3 Resultados de la prueba de rendimiento

En el escenario 1, para la detección y seguimiento de personas, se utilizan flujos de videos digitales provenientes de cámaras IP de locales exteriores vigilados por Xilema Suria en su versión 3.0. Para probar la efectividad se realizaron las pruebas de rendimiento utilizando el algoritmo propuesto, y los resultados arrojados, se pueden observar en la tabla 5.

Tabla 5: Pruebas realizadas al algoritmo en el escenario 1

Parámetro	**Valor**
Conteos Correctos (*TP)*	5482
Falsos Positivos (*FP*)	868
Falsos Negativos (*FN*)	651
Recall rate (R)	0.8633
Precision rate (P)	0.8938
F	0.8354
Error(E)	0.1646

En el escenario 2, para la detección y seguimiento de personas, se utilizan flujos de videos digitales provenientes de cámaras IP de locales interiores vigilados por Xilema Suria en su versión 3.0.Y los resultados se observan en la Tabla 6.

Tabla 6: Pruebas realizadas al algoritmo en el escenario 2

Parámetro	**Valor**
Conteos Correctos (*TP*)	4327
Falsos Positivos (*FP*)	1148
Falsos Negativos (*FN*)	874
Recall rate	0.7903
Precision rate	0.8319
F	0.8105
Error(E)	0.1894

Para concluir se realizaron las pruebas de rendimiento utilizando todos los resultados obtenidos de ambos escenarios viéndose en la Tabla 7.

Tabla 7: Resultados generales obtenidos de las pruebas realizadas al algoritmo.

Parámetro	**Valor**
Conteos Correctos (TP)	9809
Falsos Positivos (FP)	2016
Falsos Negativos (FN)	1525
Recall rate	0.8295
Precision rate	0.8654
F	0.8470
Error(E)	0.1529

Se puede apreciar que, en escenarios interiores, el rendimiento del algoritmo desciende. Esto es debido a la afectación de la iluminación artificial que puede ser poca o mucha. Esto afecta el proceso de detección de la persona. En exteriores los cambios de iluminación no afectan la detección debido a que la luz natural del sol brinda una buena iluminación. No se consideró escenarios nocturnos en exteriores, ni cuando está lloviendo, ya que la visibilidad de la cámara se afecta en exteriores, tampoco se tienen en cuenta las cámaras PTZ (Pan, Tilt and Zoom).

3.4 Conclusiones del capítulo

Al concluir este capítulo se detallaron los algoritmos utilizados para la detección y seguimiento de personas, estos fueron utilizados gracias al aporte de la biblioteca OpenCV que permitió el uso del algoritmo Viola and Jones en tiempo real, además soportó la labor de seguimiento implementado por el autor ya que brinda facilidades para el trabajo matricial con las imágenes.

La descripción del algoritmo implementado ayuda a comprender los pasos realizados para lograr el objetivo de esta investigación y permite ser una guía para futuras referencias a la labor realizada y que este resultado pueda servir de base para futuras mejoras o implementaciones de algoritmos de seguimiento y detección de personas.

El cálculo de la complejidad ciclómatica brinda la posibilidad de conocer si el código implementado se ejecuta en su totalidad y cuáles son las posibles vías para que esto pueda ser posible. Las pruebas de rendimiento mostraron que la iluminación en interiores afecta el desempeño del algoritmo. Los resultados arrojados mostraron un rendimiento general de aproximadamente 0.85 con una tasa de error aproximada de 0.15.

Conclusiones

Para la realización del presente trabajo se definieron los principales conceptos que guiaron la investigación. Esto permite comprender la situación problemática y los procesos involucrados en el objeto de estudio. Por lo que posibilita comprender la situación actual del problema para un correcto estudio de las soluciones existentes en el campo de la detección y seguimiento de personas.

El estudio de las soluciones existentes permitió el conocimiento de los algoritmos y sistemas usados para la detección y seguimiento de personas. De los algoritmos identificados se escogió Viola and Jones por su demostrada efectividad en detectar diferentes tipos de objetos entre los que se encuentran las personas.

Las herramientas y tecnologías fueron seleccionadas teniendo en cuenta las facilidades que estas brindan para el proceso de desarrollo de software, además de tener presente las que se utilizan en el desarrollo de software de la UCI. Eso permite el soporte y compatibilidad de los diferentes softwares desarrollados.

La aplicación resultante de esta investigación fue estructurada siguiendo una arquitectura centrada en flujo de datos, guiado por el patrón arquitectónico “Tuberías y Filtros”. Esto posibilitó la implementación de un componente con alto grado de reusabilidad. Dicho componente permite la detección y seguimiento de personas en un flujo de video obtenido de una cámara que cumple con todos los requisitos especificados.

Gracias al aporte de la biblioteca OpenCV se utilizó el algoritmo Viola and Jones para el proceso de detección, además se implementó el seguimiento de la persona utilizando características como el área del objeto y la distancia entre personas. El algoritmo implementado fue caracterizado para comprender los pasos realizados y además guiar futuras investigaciones en este campo. Las pruebas de rendimiento mostraron que la iluminación en interiores afecta el desempeño del algoritmo. Los resultados arrojados mostraron un rendimiento general de aproximadamente 0.85 con una tasa de error aproximada de 0.15.

Recomendaciones

El autor de la presente investigación recomienda:

1. Utilizar técnicas de programación que permitan manejar mejor las oclusiones, ya sea por objetos o por otras personas.

2. Mejorar el comportamiento del algoritmo con los cambios de iluminación, principalmente en escenarios interiores.

Referencias

Abbes, Mounir Tahar, Mohamed, Belhirech y Mohamed, Senouci. 2016. Adaptation of a routing algorithm in wireless video sensor network for disaster scenarios using JPEG 2000. *Wireless Networks.* Springer, 2016, Vol. 22, 2.

Alegsa, Leandro. 2016. *Diccionario de la informatica y la ténologia.* Argentina : Santa fe, 2016.

Álvarez, Ángel Francisco del Río. 2015. Sistema aéreo no tripulado para el reconocimiento de matrículas de coche. 2015.

Bhanu, Bir, y otros. 2011. *Distributed video sensor networks.* s.l. : Springer Science \& Business Media, 2011.

Bhatia, Akshay. 2007. *Hessian-Laplace Feature Detector and Haar.* Ottawa, Canada : Akshay Bhatia, 2007.

Chávez, Jorge Lira. 2010. *Tratamiento Digital de Imágenes Multiespectrales.* Ciudad Universitaria, México : s.n., 2010. 978-1-105-04502-8.

Dalal, Navneet. 2005. INRIA Person Dataset. *INRIA Person Dataset.* [En línea] 2005. [Citado el: 8 de abril de 2005.] http://pascal.inrialpes.fr/data/human/.

Deitel, Paul J. Deitel Harvey. 2016. *C++ How to Program.* s.l. : Prentice Hall PTR, 2016. 9780134448237.

Fischer, Walter. 2010. *Digital Video and Audio Broadcasting Technology.* Berling : Springe, 2010. 978-3-642-11611-7.

Fría, Criminalidad y desajustes de las políticas de seguridad: un panorama de América Latina del post-Guerra. 2016. Miranda, Jaseff Raziel Yauri. *PRODUCE• EKOIZLEA.* 2016, Vol. 61, págs. 97--12.

Fuhrmann, Hauke A. L. 2011. *On the Pragmatics of Graphical Modeling.* 2011.

García, Miguel Redondo. 2005. *Ilustración Digital.* 2005. 978-84-369-4848-6.

Gómez, Ruth Priscila Landeros. 2009. *Herramientas Case.* s.l. : El Cid Editor, 2009.

González, Alfredo Rolando Hernández León Sayda Coello. 2011. *El Proceso de Investigación Científica.* Ciudad de la Habana : Universitaria, 2011. 978-959-16-1307-3.

Gonzalez, Rafael C. 2011. *Digital Image Processing .* 2011.

Guerra Carapas, Edgar Reinaldo. 2016. *Prototipo de un sistema web para el conteo de vehículos en tiempo real utilizando la librería de visión artificial OpenCV.* s.l. : Quito, 2016.

Halvorson, Michael. 2013. *Microsoft Visual Basic 2013 Step by Step.* s.l. : Pearson Education, 2013. 9780735673410.

Herbert Bay, Tinne Tuytelaars and Luc Van Gool. 2006. *SURF: Speeded Up Robust Features.* Luc.Vangool : s.n., 2006.

Herbert, Bay, Tinne Tuytelaars , Luc Van Gool. 2016. *Speeded up robust features.* 2016. pág. 14. Vol. 3951.

Hueber, Nicolas, y otros. 2014. SPIE Defense+ Security. s.l. : International Society for Optics and Photonics, 2014, págs. 90790B--90790B.

Jacobson, Ivar, Booch, Grady and Rumbaugh, James. 2000. *El proceso unificado de desarrollo de software.* Madrid : Pearson educación, 2000.

Jones, Viola and. 2004. *Robust Real-Time Face Detection.* 2004.

Kejriwal, Mayank y Miranker, Daniel P. 2015. Semi-supervised instance matching using boosted classifiers. *European Semantic Web Conference}.* s.l. : Springer, 2015, págs. 388--402.

Larman, Craig. 2010. *UML y Patrones. 2a Edición.* 2010.

Lienhart, Rainer, Kuranov, Alexander y Pisarevsky, Vadim. 2003. Empirical analysis of detection cascades of boosted classifiers for rapid object detection. *Joint Pattern Recognition Symposium.* s.l. : Springer, 2003, págs. 297--304.

Lowe, David G. 1999. Object Recognition from Local Scale-Invariant Features (SIFT). *International Conference on Computer Vision.* 1999, Vol. 2, págs. 1150-1157 vol.2.

Lucena, Antonio Jesús Domínguez, José María Rodriguez, Solís Gómez Ibarlucea, Francisco Muños Usano, Luis del Castillo Torres. 2004. *Auxiliares de Seguridad de la Junta de Andalucia.test Ebook.* Sevilla : MAD, S.L., 2004. 84-665-4113-6.

Mata, Francisco Javier García. 2010. *Videovigilancia: CCTV usando vídeos IP.* Málaga : Vértice, 2010. 978-84-9931-356-6.

Mestras, Juan Pavón. 2004. *Patrones de diseño orientado a objetos.* 2004.

Mullo López, Estefanía Dayana. 2016. *Diseño e implementación de un sistema para la contabilización de la cantidad de objetos que circulan por el espacio captado por una cámara utilizando librerías de OPENCV.* 2016.

Nock, Richard, y otros. 2014. Gentle nearest neighbors boosting over proper scoring rules. *IEEE transactions on pattern analysis and machine intelligence.* 2014, Vol. 37, 1, págs. 80--93.

Omid, Hosseini Jafari y Dennis Mitzel, Bastian Leibe. 2014. *Real-time RGB-D based people detection and tracking for mobile robots and head-worn cameras.* Hong Kong, : s.n., 2014.

Parody, Auris, y otros. 2015. Implementing of a system of counting of students, using wireless sensor networks Implementación de un sistema de conteo de estudiantes utilizando redes de sensores inalámbricos. *Actas de Ingenierí}a.* 2015, Vol. 1.

Pineda, Fernando Boronat Seguí Miguel García. 2010. *IPTV, la televisión por Internet.* Málaga, España : Vértice, 2010.

Pressman, Roger. 2010. *Ingenería de software. Un enfoque práctico.* Mexico : McGRAW-HILL INTERAMERICANA, 2010.

Reina, Damián. 2016. Historia de la videovigilancia. *Historia de la videovigilancia.* [En línea] Racalarm, 25 de Mayo de 2016. [Citado el: 05 de junio de 2017.] http://racalarm.com/blog/cctv/historia-de-la-videovigilancia/.

Rumbaugh, James, Jacobson, Ivar y Booch, Grady. 2009. *El Lenguaje Unificado de Modelado. Manual de Referencia.* España : s.n., 2009. ISBN: 84-7829-037-0.

Russell Stewart, Mykhaylo Andriluka, Andrew Y. Ng. 2016. *End-To-End People Detection in Crowded Scenes.* 2016.

team, OpenCV. 2017. OpenCV Open Source Computer Vision. [En línea] 2017. http://docs.opencv.org/ref/2.4.13/.

Torres, AlejandroDomínguez. 1996. Procesamiento digital de imágenes. 1996.

Tracking Groups of People. **McKenna, Stephen J., y otros. 2000.** 2000, Journal, págs. 42-56.

Txarterina, Jon Intxaurbe. 2013. *Detección de personas.* 2013.

Viola, Paul y Jones, Michael. 2001. Computer Vision and Pattern Recognition, 2001. CVPR 2001. Proceedings of the 2001 IEEE Computer Society Conference on. *apid object detection using a boosted cascade of simple features.* 2001, Vol. 1.

W Powers, David M. 2007. *Evaluation: From Precision, Recall and F-Factor to ROC,.* Australia : Adelaide, 2007. 24.

Anexos

Tabla 8: Descripción del CU Seguir Persona

CU	Seguir persona.	
Objetivo	El objetivo de este CU es permitir el seguimiento de una persona en un flujo de video.	
Actores	Usuario	
Resumen	El caso de uso inicia una vez extraídas las características de la persona que se desea seguir. El CU termina cuando el usuario decida terminar el seguimiento.	
Complejidad	Alta	
Prioridad	Alta	
Precondiciones	Un flujo de video	
Postcondiciones	No procede	
Flujo de eventos		
Flujo básico: Seguir persona en el video.		
	Actor	**Sistema**
1		Se obtienen los datos de la persona escogida.
2		Se inicia el rastreo de las características guardas.
3		Se inicia el seguimiento y se muestra el mensaje “Siguiendo”
4	Termina el CU cuando el usuario decida dejar el seguimiento y se muestra el mensaje “Seguimiento detenido”.	

Relaciones	**CU Incluidos**	No procede
	CU Extendidos	No procede
Referencias	RF4, RF5, RnF1, RnF2,RnF3, RnF4, RnF5, RnF6.	

Tabla 9: Descripción del CU Escoger persona en el vídeo

CU	Escoger persona en el vídeo	
Objetivo	El objetivo de este CU es permitir que el usuario pueda escoger una de las personas detectadas.	
Actores	Usuario	
Resumen	El caso de uso inicia una vez escogida una de las personas que se ha detectado. El CU termina cuando el usuario decida terminar el seguimiento.	
Complejidad	Alta	
Prioridad	Alta	
Precondiciones	Un flujo de video	
Postcondiciones	No procede	
Flujo de eventos		
Flujo básico: Escoger persona en el vídeo		
	Actor	**Sistema**
1.	Usuario escoge la persona	
2		Se extraen las características de la persona escogida.
3	Termina el CU después de extraídas las características.	
Relaciones	**CU Incluidos**	Seguir persona

	CU Extendidos	No procede
Referencias	RF3, RnF1, RnF2,RnF3, RnF4, RnF5	