

PONTIFICIA UNIVERSIDAD CATÓLICA DEL PERÚ
FACULTAD DE CIENCIAS E INGENIERÍA



PONTIFICIA
**UNIVERSIDAD
CATÓLICA**
DEL PERÚ

**DETECCIÓN DE PERSONAS Y VEHÍCULOS EN IMÁGENES
TOMADAS DESDE UN UAV**

Tesis para optar el Título de Ingeniero Electrónico, que presenta el bachiller:

Jhon Jhojan Rivera Vicente

ASESOR: Pedro Crisóstomo Romero

Lima, Diciembre 2015

RESUMEN

En la actualidad, la detección de objetos de interés en imágenes se ha centrado básicamente en aplicaciones que se desarrollan sobre cámaras estáticas, limitando así el área que es captada por dichas cámaras. Pero con el desarrollo y construcción de los vehículos aéreos no tripulados, ahora tenemos cámaras que se montan en ellos. Los objetos a detectar con una cámara en movimiento generan un nuevo reto y es por eso que esta tesis se enfoca en el análisis de este tipo de imágenes.

En la presente tesis, se desarrolló un sistema que permite la detección de objetos de interés (persona y vehículos) que se encuentran en movimiento, a partir de imágenes tomadas desde un vehículo aéreo no tripulado (UAV). El resultado de la aplicación de este sistema, permitirá conocer el número de objetos en movimiento, así como la trayectoria que siguen a lo largo de una secuencia de imágenes.

En el primer capítulo se presenta el estado del arte existente en la actualidad en referencia a las aplicaciones que existen en reconocimiento de objetos en imágenes, ya sean imágenes individuales o secuencias de imágenes. Además se establecen los objetivos del desarrollo del sistema que se propone en esta tesis.

En el segundo capítulo, se presenta las técnicas existentes en el procesamiento de imágenes para el reconocimiento de objetos, así como también las respectivas etapas con las que cuentan, para detectar y reconocer objetos de interés.

En el tercer capítulo se realiza la comparación entre las técnicas de reconocimiento con el objetivo de seleccionar la más adecuada a nuestros requerimientos. Así mismo se presenta el diseño del sistema propuesto para la detección de personas y vehículos en movimiento en las imágenes tomadas desde el UAV.

Finalmente en el último capítulo, se muestra la implementación del sistema propuesto y sus resultados obtenidos al aplicar el sistema sobre 5 secuencias de fotogramas (videos), cada una de ellas con características distintas, con lo que se establecerá la eficiencia de nuestro sistema en cuanto a escenarios diversos.

Índice general

Introducción.....	1
1 Capítulo 1 LOS UAVS Y SUS APLICACIONES EN CIUDADES	2
1.1 Vehículos aéreos no tripulados (UAV)	2
1.2 UAVs en el Perú.....	2
1.3 Ventajas del uso de UAVs.....	3
1.4 Aplicaciones en ciudades	3
1.5 Estado del arte	4
1.5.1 Objeto de estudio.....	4
1.5.2 Estado de la investigación	4
1.5.1 OpenCV.....	7
1.6 Objetivos	7
2 Capítulo 2 TÉCNICAS DE PROCESAMIENTO DE IMÁGENES REFERIDAS A LA DETECCIÓN Y RECONOCIMIENTO DE OBJETOS	8
2.1 Introducción sobre las técnicas.....	8
2.2 SIFT (Scale Invariant Feature Transform)	8
2.2.1 Obtención de las características de una imagen.....	8
2.2.2 Detección de puntos característicos	10
2.2.3 Asignación de orientación	11
2.2.4 Descriptor de puntos característico	11
2.2.5 Emparejamiento.....	11
2.3 ASIFT (Affine Scale Invariant Feature Transform)	12
2.3.1 Simulación de distorsiones.....	12
2.3.2 Emparejamiento.....	13
2.4 SURF (Speeded up Robust Features).....	13
2.4.1 Detección de puntos claves.....	13
2.4.2 Asignación de orientación	14
2.4.3 Creación del descriptor	14
3 Capítulo 3 SELECCIÓN DE LAS TÉCNICA DE RECOMOCIMIENTO	15
3.1 Análisis comparativo entre las técnicas de reconocimiento	16
3.1.1 Aplicación del método SIFT	16
3.1.2 Aplicación de método SURF	16
3.2 Diseño del Sistema propuesto	18
4 Capítulo 4 IMPLEMENTACIÓN DEL SISTEMA	21
4.1 Banco de datos.....	21

4.1.1	Videos.....	21
4.1.2	Conteo manual de autos y personas.....	23
4.1.3	Extracción de fotogramas.....	24
4.2	Alineamiento entre frames.....	25
4.2.1	Extracción de puntos característicos mediante la técnica de detección.....	26
4.2.2	Emparejamientos de puntos característicos.....	28
4.2.3	Transformación de fotograma previo.....	29
4.2.4	Resultado del alineamiento.....	32
4.3	Consideraciones.....	33
4.3.1	Medidas de objetos a detectar.....	33
4.3.2	Velocidad de desplazamiento.....	34
4.3.3	Traslape por desplazamiento.....	36
4.4	Segmentación de imagen.....	38
4.5	Seguimiento de objetos de interés.....	41
4.5.1	Distancia entre centroides.....	41
4.6	Pruebas y resultados.....	45
4.6.1	Resultados de la primera prueba.....	45
4.6.2	Resultados de la segunda prueba del sistema.....	49
4.6.3	Resultados de la tercera prueba del sistema.....	50
4.6.4	Resultados de la cuarta prueba del sistema.....	51
4.6.5	Resultados de la quinta prueba del sistema.....	53
4.6.6	Análisis de resultados.....	55
	Conclusiones.....	56
	Recomendaciones.....	57
	Bibliografía.....	58

Lista de Figuras

Figura 1.1: UAV denominado Quinde, parte del proyecto PEGASO [1].	2
Figura 1.2: Vista aérea de zona urbana desde un UAV [5].	3
Figura 1.3: Personas detectadas por sistema de visión de computador [2].	4
Figura 1.4: Imagen tomada desde un satélite [4].	5
Figura 1.5: Sección de la imagen. (a) imagen original. (b) imagen original filtrada. (c) puntos claves obtenidos mediante SIFT [4].	5
Figura 1.6: Cambios en escala y rotación de los objetos en imágenes [5].	6
Figura 1.7: Representación del modelamiento de bicicletas [6].	7
Figura 2.1: Pirámide gaussiana [9].	9
Figura 2.2: Pirámide gaussiana, luego de la operación diferencia [9].	9
Figura 2.3: Comparación de pixeles vecinos [9].	9
Figura 2.4: Puntos característicos obtenidos [9].	10
Figura 2.5: Reducción de los puntos encontrados, mediante las condiciones [10].	10
Figura 2.6: Representación de la ventana gaussiana y su generación de histogramas [10].	11
Figura 2.7: Identificación de puntos característicos y emparejamiento de los mismos [11].	11
Figura 2.8: Deformación de losetas de piso, debido a la perspectiva de la cámara [12].	12
Figura 2.9: Representación de las Inclinaciones de un objeto en distintas imágenes [12].	12
Figura 2.10: Comparación entre los emparejamientos usando SIFT (b) y ASIFT(a) [13].	13
Figura 2.11: Generación de regiones cuadradas centradas en los puntos de interés [14].	14
Figura 2.12: Emparejamiento de puntos característicos mediante SURF [15].	14
Figura 3.1: Puntos característicos hallados mediante el método SIFT.	16
Figura 3.2: Puntos característicos hallados mediante el método SURF.	16
Figura 3.3: Representación del problema del desplazamiento del UAV en las fotogramas tomados, (a) fotograma previo, (b) fotograma actual.	18
Figura 3.4: Proceso para la detección de objetos de interés.	19
Figura 3.5: Ejemplo de alineamiento entre dos fotogramas.	20
Figura 4.1: Imagen con presencia de traslape entre los vehículos [16].	21
Figura 4.2: Galería de videos con la que se trabajará.	22
Figura 4.3: Diagrama de flujo del proceso de extracción de fotogramas.	24
Figura 4.4: Extracción de objeto en movimiento.	25
Figura 4.5: Resta entre imágenes tomadas desde un UAV.	25
Figura 4.6: Proceso de alineamiento de fotogramas.	26
Figura 4.7: Representación de los puntos característicos del fotograma previo 4.7a y fotograma actual 4.7b.	27
Figura 4.8: Representación del conjunto de descriptores SURF de una imagen de prueba.	27
Figura 4.9: Proceso de emparejamiento [18].	28
Figura 4.10: Transformación de perspectiva.	29
Figura 4.11: Etapa de obtención de matriz de Homografía.	30
Figura 4.12 Matriz de homografía obtenida entre los fotogramas de la figura 4.7.	31
Figura 4.13 Representación de los emparejamientos correctos entre el fotograma previo y actual.	31
Figura 4.14: Fotograma previo alineado (Usando la matriz de transformación) al fotograma actual.	32
Figura 4.15: Resta de los fotograma previo y actual, sin aplicar alineamiento.	32
Figura 4.16: Resta de los fotograma previo y actual, aplicando el alineamiento.	33
Figura 4.17: Medidas de los objetos a detectar.	34

<i>Figura 4.18: Desplazamiento de vehículos con diferentes velocidades.</i>	35
<i>Figura 4.19: desplazamiento de auto a lo largo de una distancia igual a su longitud.</i>	36
<i>Figura 4.20: Representación de autos en movimiento.</i>	36
<i>Figura 4.21: Representación de autos en movimiento.</i>	37
<i>Figura 4.22: etapa de extracción de puntos característicos.</i>	37
<i>Figura 4.23: Efecto del traslape de desplazamiento.</i>	37
<i>Figura 4.24: Método de selección de los fotogramas a procesar.</i>	38
<i>Figura 4.25: Proceso de segmentación de imagen para la extracción de objetos de interés.</i>	39
<i>Figura 4.26: Objetos detectados luego de la etapa de segmentación.</i>	39
<i>Figura 4.27: Representación de un vehículo en desplazamiento y su respectivo blob.</i>	40
<i>Figura 4.28: Representación de la distancia estimada entre centroides.</i>	41
<i>Figura 4.29: Metodología propuesta para el seguimiento de blobs u objetos aptos detectados.</i>	42
<i>Figura 4.30: Distancias entre los blobs de la imagen inicial y posterior.</i>	43
<i>Figura 4.31: Representación de los centroides de los blobs en las tres imágenes resultantes.</i>	43
<i>Figura 4.32: Primera prueba, donde se muestra los 7 primeros fotogramas tomados de 5 en 5 y la obtención de las 6 primeras imágenes, luego de la segmentación de imagen.</i>	46
<i>Figura 4.33: Trayectoria del vehículo detectado.</i>	47
<i>Figura 4.34: Seguimiento del blob 2 a lo largo de la secuencia de imágenes.</i>	48
<i>Figura 4.35: Trayectoria de los blob 2 y 8.</i>	49
<i>Figura 4.36: Trayectorias de dos blobs detectados.</i>	50
<i>Figura 4.37: Efecto de la selección adecuada de una tasa de muestreo.</i>	51
<i>Figura 4.38: Trayectorias seguida por los blob 73 y 74, que representa a un vehículo y un objeto no correcto, respectivamente.</i>	52
<i>Figura 4.39: Trayectoria de los blobs 104 y 102</i>	54

Lista de Tablas

<i>Tabla 3.1: Comparación entre los resultados obtenidos por métodos SIFT y SURF.</i>	<i>18</i>
<i>Tabla 4.1: Factores controlables y no controlables en el procesamiento de imágenes.</i>	<i>23</i>
<i>Tabla 4.2: Conteo manual de los objetos en movimientos en el banco de videos.</i>	<i>24</i>
<i>Tabla 4.3: Almacenamiento coordenadas de 4 blobs detectados en las primeras 3 imágenes resultantes (16 fotogramas).</i>	<i>44</i>
<i>Tabla 4.4: Parámetros considerados en el procesamiento del video torono_58s.avi.</i>	<i>45</i>
<i>Tabla 4.5: Almacenamiento coordenadas de 4 blobs detectados en las primeras 6 imágenes.</i>	<i>47</i>
<i>Tabla 4.6: Número de objetos detectados como correctos en la primera prueba.</i>	<i>48</i>
<i>Tabla 4.7: Número total de blobs detectados en la primera prueba.</i>	<i>48</i>
<i>Tabla 4.8: Parámetros considerados en el procesamiento del video soccer1.wmv.</i>	<i>49</i>
<i>Tabla 4.9: Almacenamiento de las coordenadas de 4 blobs detectados en las primeras 6 imágenes resultantes.</i>	<i>49</i>
<i>Tabla 4.10: Número de objetos detectados en la segunda prueba.</i>	<i>50</i>
<i>Tabla 4.11: Parámetros considerados para procesar el video toronto_75s.avi.</i>	<i>50</i>
<i>Tabla 4.12: Número de objetos detectados en la tercera prueba.</i>	<i>51</i>
<i>Tabla 4.13: Parámetros considerados para procesar el video via.mp4.</i>	<i>52</i>
<i>Tabla 4.14: Almacenamiento de las coordenadas de los blobs 73 y 74.</i>	<i>53</i>
<i>Tabla 4.15: Número de objetos totales detectados en la cuarta prueba.</i>	<i>53</i>
<i>Tabla 4.16: Numero de objetos detectados como correctos en la cuarta prueba.</i>	<i>53</i>
<i>Tabla 4.17: Parámetros considerados para la prueba con el video toronto_75s.avi.</i>	<i>54</i>
<i>Tabla 4.18: Número total objetos detectados en la quinta prueba.</i>	<i>54</i>
<i>Tabla 4.19: Número de objetos detectados como correctos en la quinta prueba.</i>	<i>54</i>
<i>Tabla 4.20: Cuadro resumen de los resultados obtenidos en las pruebas.</i>	<i>55</i>
<i>Tabla 4.21: Blobs detectados antes y después de la etapa de segmentación de imagen.</i>	<i>55</i>

Introducción

La rápida expansión del uso de robots en casi todas las aplicaciones tecnológicas, ha desarrollado nuevas expectativas en cuanto hasta dónde se puede llegar con el diseño especializado de estos robots. Un ejemplo de ello es el desarrollo de naves aéreas no tripuladas (UAV, por sus siglas en inglés), que se pueden usar en muchas aplicaciones, tales como militares, tránsito, búsqueda, rescate, agricultura, etc.

La utilización de los UAVs en las aplicaciones antes mencionadas ha sido posible gracias al desarrollo de diferentes algoritmos matemáticos referidos a comunicaciones y procesamiento de imágenes. Una de las principales aplicaciones de los UAV, es la detección y/o seguimiento de objetos, que pueden ser autos, personas, animales, estructuras hechas por el hombre, entre otras.

Con respecto al tránsito vehicular, el uso de imágenes aéreas desde UAVs ayuda a controlar el flujo vehicular, así como la búsqueda de determinados tipos de vehículos, mediante sistemas conocidos como *trackeo*. En el ámbito de la seguridad pública también se han hecho grandes avances tales como el desarrollo de sistemas de reconocimiento facial, el cual permite identificar personas mediante imágenes tomadas por los UAVs que muestran el rostro de las personas y las compara con una base de datos. También el uso de imágenes ayuda a los mismos UAVs automáticos, los cuales actuarán de forma pertinente, según sean las características de los objetos que detecten; es decir, la detección de un objeto y luego su posterior traslado generará en el sistema de vuelo del UAV, que siga al objeto, controlando la dirección y altura de vuelo. Y finalmente podemos mencionar que una de las aplicaciones apunta a la detección de objetos específicos, como por ejemplo edificaciones, las cuales deben ser diferenciadas de los objetos naturales

Es por esas razones que el estudio y desarrollo de algoritmos de detección se ha hecho muy importante en el avance tecnológico, y es hacia donde profundiza esta investigación. En la obtención de información del flujo vehicular y peatonal en calles y avenidas.

Capítulo 1

LOS UAVS Y SUS APLICACIONES EN CIUDADES

1.1 Vehículos aéreos no tripulados (UAV)

Los vehículos aéreos no tripulados (UAV), también conocidos como drones, son naves las cuales no necesitan de un piloto a bordo para volar. Los UAV (Unmanned aerial vehicle) poseen un control de vuelo dirigido desde una base de forma manual por un piloto o de forma automática, mediante un sistema de control de vuelo programado.

1.2 UAVs en el Perú

Esta expansión del uso de los UAVs no es ajena al Perú, pues en el país se han construido diversos modelos de UAVs con diferentes características de operación. Esta aérea de investigación se empezó a desarrollar a finales de los 90's, cuando se creó uno de los primeros prototipos [1], ver figura 1.1.

Aunque se presentaron algunos fracasos, ya sea por problemas logísticos o problemas de presupuesto, se siguió apostando por estos sistemas de vuelo no tripulados. Un gran paso en el desarrollo de estos, fue el apoyo recibido por parte de Concytec (Consejo Nacional de Ciencia y Tecnología), el cual permitió la creación de tres modelos de aeronaves.

El impulso de las Fuerza Aérea del Perú, por el desarrollo de estas naves, se basa en objetivos militares, tales como operación en el VRAE (Valle del Rio Apurímac y Ene), pero aunque para este fin también se invirtió dinero en la compra de UAVs de fabricación extranjera.



Figura 1.1: UAV denominado Quinde, parte del proyecto PEGASO [1].

1.3 Ventajas del uso de UAVs

Como se mencionó, los UAVs se han utilizado en la vigilancia, reconocimiento y localización de objetos, su ventaja radica en los siguientes aspectos:

- No arriesgan vidas humanas en cualquiera de sus aplicaciones.
- No están sujetos a ninguna necesidad ergonómica.
- Alberga todo tipo de sistemas de comunicación, control u operación.
- Adaptabilidad a diversas aplicaciones gracias a sus carga útiles (sensores) modulares.
- Gran maniobrabilidad y poder de acceso a lugares inalcanzables.
- Menor coste de mantenimiento (consumo de energía)

1.4 Aplicaciones en ciudades

Los drones o UAVs poseen un gran potencial en diversas áreas proporcionando vistas amplias de un determinado lugar e inclusive recogiendo información mediante los diferentes sensores que se le coloquen al UAV. A continuación se mencionaran algunas de las aplicaciones.

Existen aplicaciones en el área de cartografía donde se recogen imágenes de secciones de terreno o un área poblada, para generar un mapa total tal como se muestra en la figura 1.2. Esto nos permite reducir gastos, tanto en costo de personal encargado de pilotar las aeronaves convencionales y costo en el consumo de combustible.



Figura 1.2: Vista aérea de zona urbana desde un UAV [5].

Otras de las distintas aplicaciones de los UAVs se centran en operaciones de rescate o búsqueda de personas en lugares de difícil acceso como zonas montañosas, edificios colapsados o amplias extensiones de terreno, como desiertos e inclusive en mar abierto.

Dependiendo del tipo de UAV, se puede obtener una grabación de video de todo el terreno que sobrevuela o enviar imágenes fotográficas, las cuales serán procesadas mediante algoritmos de detección, tal como se muestra en la figura 1.3



Figura 1.3: Personas detectadas por sistema de visión de computador [2].

1.5 Estado del arte

1.5.1 Objeto de estudio

La utilización de los UAVs en diferentes campos ha logrado desarrollar aplicaciones como las ya mencionadas. Pero en el Perú existen campos en los cuales sería de ayuda el uso de estos UAVs, pero no se han implementado aún, a pesar de la gran ventaja que conllevaría ello, ya sea por la falta de conocimiento de la utilidad de éstos o por falta de algoritmos que permitan desarrollar una aplicación adecuada. En este sentido, se busca desarrollar un sistema que localice objetos o elementos de interés que se encuentren en movimiento, en las imágenes que serán tomadas previamente por un UAV, estas imágenes ingresarán al sistema, el cual las procesará para poder así detectar objetos y realizar el seguimiento de estos.

El sistema a desarrollar contará con el uso del procesamiento de imágenes, el cual en la actualidad cuenta con una gran variedad de técnicas, basadas en diferentes algoritmos matemáticos, se buscará el adecuado dependiendo en qué tipo de aplicación y objeto de estudio se quiera implementar. Más adelante se verán las características y diferencias existentes entre estos algoritmos.

1.5.2 Estado de la investigación

Existen diversos trabajos previos en los cuales se usaron diferentes algoritmos para la detección de objetos en imágenes; si bien algunas de estas investigaciones no usaron UAVs para obtener sus imágenes, es importante mencionarlos, pues hacen uso de técnicas de detección y reconocimiento que nos serán útiles en cierta medida.

1.5.2.1 Método de detección basado en aprendizaje semi-supervisado

La detección de objetos de interés en imágenes se ha estudiado por años, pero se dan casos donde la variación de tamaños de un mismo tipo de objeto lo hace más complejo y si hace eso se le agrega la variación que existe entre el fondo de a imagen y el objeto de estudio, se hace necesario el uso de un sistema de detección que sea robusta ante este tipo de perturbaciones. Por lo que en este trabajo se aprecia el uso de algoritmos de aprendizaje semi-supervisado, el cual usa el contexto del objeto [3].

1.5.2.2 Método de detección basado en algoritmo SIFT y grafos

Existen aplicaciones donde no se usaron UAVs, para obtener las imágenes, y esto ocurre en este caso, pues en esta investigación se usó un satélite de alta resolución, con lo cual se quiere encontrar lugares en construcción, ver figura 1.4. Para poder detectar este tipo de objeto se utilizó la técnica SIFT (Scale Invariant Feature Transform), pues esta técnica nos permite obtener información del objeto sin importar la escala de la transformación de este, ver figura 1.5. Esta técnica de detección se explicara en el capítulo 2. Sin embargo, SIFT no es lo suficiente para este tipo de aplicación, por lo que se recurre a la teoría de grafos. Con lo que se puede reconocer los vértices que poseen las construcciones [4].



Figura 1.4: Imagen tomada desde un satélite [4].



Figura 1.5: Sección de la imagen. (a) imagen original. (b) imagen original filtrada. (c) puntos claves obtenidos mediante SIFT [4].

1.5.2.3 Método basado en SIFT para seguimiento de objetos

En esta investigación se hace uso de la técnica SIFT para obtener referencias visuales de objetos para el manejo o control de UAVs. La técnica SIFT usa los puntos más importantes de cada imagen que es procesada, en este método se usa pares de dichos puntos que son cercanos en cada imagen, con lo cual el UAV seguirá el movimiento del objeto cuando vea que cada par de puntos se sigue encontrando junto. El sistema se probó usando secuencias de imágenes donde la imagen ha sido rotada o escalada, como se aprecia en la figura 1.6, y se logró controlar el vuelo y velocidad del UAV.

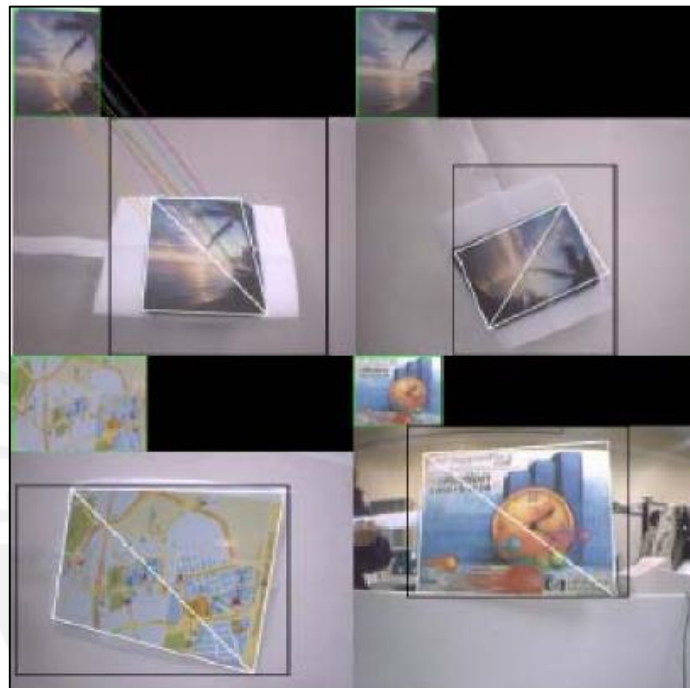


Figura 1.6: Cambios en escala y rotación de los objetos en imágenes [5].

1.5.2.4 Modelamiento de objetos deformables

En este método para poder detectar un objeto, este debe ser la mezcla de modelos de piezas deformables multiescala. Se crea un diseño de un objeto, el cual llamaremos modelo, disgregando en partes que se consideran característicos de dicho objeto, así como se muestra en la figura 1.7. Luego se procesará los objetos que se encuentran en las imágenes tomadas, y entonces se inicia el algoritmo de entrenamiento iterativo que alterna entre los valores latentes, los cuales son del objeto de la imagen, con los valores óptimos que se crearon con el diseño del objeto, conocido como modelo, logrando así una comparación exitosa o fallida y reconociendo si se tiene una imagen u objeto de estudio correcto o no.

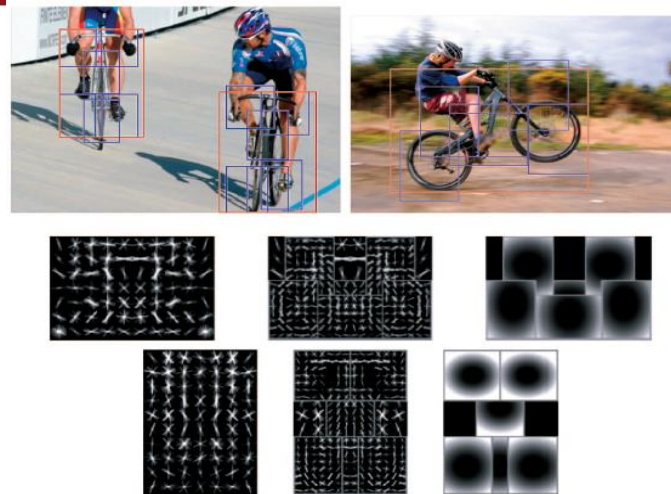


Figura 1.7: Representación del modelamiento de bicicletas [6].

1.5.1 OpenCV

OPENCV es una librería de código abierto para el procesamiento de imágenes desarrollada por Intel en 1999. Esta librería es usada en muchas aplicaciones como:

- Reconocimiento de imágenes.
- Tratamiento de imágenes (contraste, brillo, umbral, etc.)
- Visión robótica.

Una de las ventajas de esta librería es que se puede utilizar sobre cualquier plataforma; es decir puede usarse sobre Windows, Linux y Mac OS. Esta librería cuenta con alrededor de 2500 funciones, las cuales se implementan muy bien en los procesadores Intel.

1.6 Objetivos

Objetivo General

Diseñar e implementar un sistema que detecte o reconozca objetos específicos (vehículos y personas) en imágenes aéreas.

Objetivos Específicos

Estudio de los algoritmos existentes en el campo del reconocimiento de objetos en imágenes.

Obtener un banco de imágenes con el cual se pueda trabajar.

Implementar en Matlab y lenguaje C los algoritmos usando la Librería OPENCV.

Pruebas con imágenes disponibles.

Capítulo 2

TÉCNICAS DE PROCESAMIENTO DE IMÁGENES REFERIDAS A LA DETECCIÓN Y RECONOCIMIENTO DE OBJETOS

2.1 Introducción sobre las técnicas

Las técnicas de reconocimiento, tratan de extraer las características importantes de una imagen, una secuencia de imágenes, etc. las diferencias entre estas técnicas, se basan en la cantidad de características pueden extraer y el tiempo que toma calcularlas. Existen áreas de la ciencia donde se aplican, como se detalló en el capítulo anterior.

2.2 SIFT (Scale Invariant Feature Transform)

SIFT es un método que fue propuesto por David Lowe, donde a una imagen se le hace una transformación de la información en puntos que son invariantes de escala, rotaciones y luminosidad. A continuación se harán mención de las etapas con las que cuenta este método.

2.2.1 Obtención de las características de una imagen

La primera etapa es el detector, el cual obtiene un conjunto de puntos de la imagen, puntos característicos, y realiza un proceso para obtener dichos puntos.

2.2.1.1 Función Espacio-Escala

Para calcular los puntos, se deben hacer variaciones de escala de la imagen, para lo cual se realiza la operación de convolución entre la imagen y la gaussiana de la imagen, filtrado de la imagen. Se definirá la imagen como una función de $I(x, y)$ y la gaussiana como se muestra en la ecuación 1.1 [7] y su vez se define la ecuación 1.2. [8].

$$L(x, y, \sigma) = G(x, y, \sigma) * I(x, y) \quad \text{Ecuación 2.1}$$

Donde:
$$G(x, y, \sigma) = \frac{1}{\sqrt{2\pi}\sigma} e^{-\frac{(x-y)^2}{2\sigma^2}} \quad \text{Ecuación 2.2}$$

Ahora se construye una pirámide gaussiana, ver figura 2.1 convolucionando con diversos filtros, para lo cual se variara el valor de σ , iniciando con un valor de 0.5, el cual nos permite suavizar la imagen, para luego aumentar el valor de σ a 1 y tener la imagen pero con el doble de tamaño. Se construirá la pirámide dependiendo de cuantas veces se haga el filtrado. Pero el penúltimo valor de cada octava debe ser el doble del inicial.

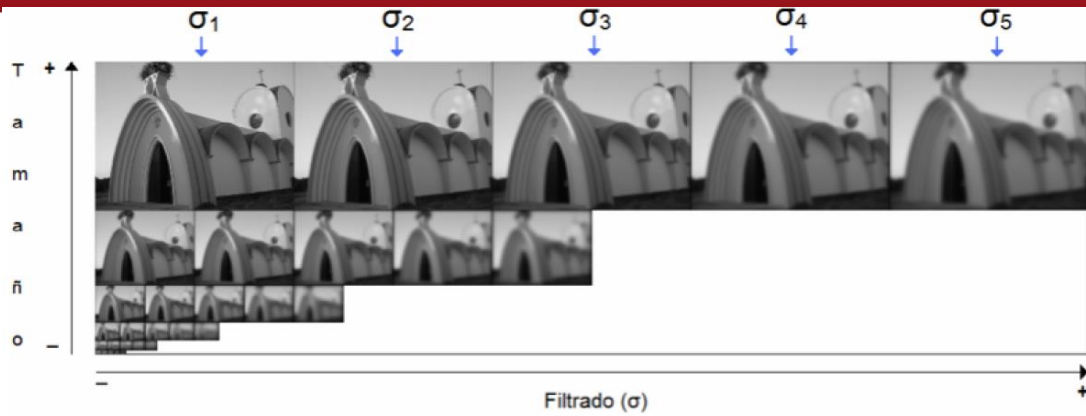


Figura 2.1: Pirámide gaussiana [9].

2.2.1.2 Función diferencia Gaussiana

Lo siguiente es encontrar los puntos característicos que sean estables, por lo que se trabajará con la diferencia entre las imágenes de un mismo tamaño y que sean vecinas, por lo que si en la pirámide teníamos 5 imágenes por cada octava ahora se tendrán 4 imágenes por cada octava, después de la diferencia. Ver figura 2.2.

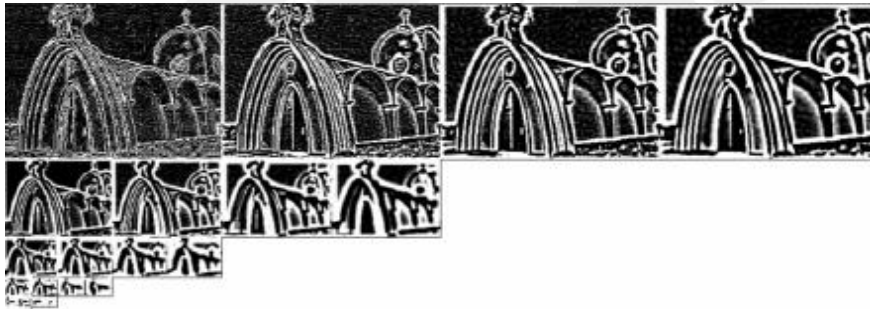


Figura 2.2: Pirámide gaussiana, luego de la operación diferencia [9].

2.2.1.3 Detección local

Una vez que se termina de aplicar la función de diferencia gaussiana, se empezará con los cálculos de los valores máximos y mínimos, pues son estos los que la caracterizan a una imagen. Para el cálculo de estos puntos característicos, cada pixel de una imagen es comparado con los 8 pixeles vecinos y además con los otros 18 que se encuentran en las otras imágenes vecinas, que son de diferentes escalas. Ver figura 2.3.

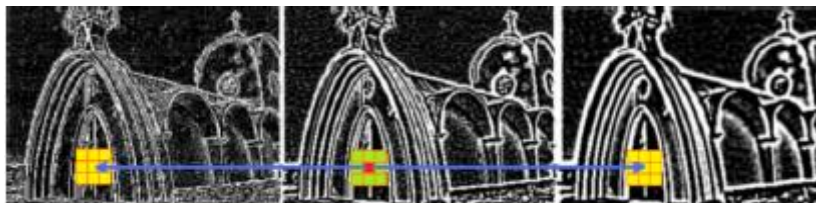


Figura 2.3: Comparación de pixeles vecinos [9].

Entonces un punto característico se obtendrá cuando un pixel sea mayor a los otros 26 pixeles o sea el menor.

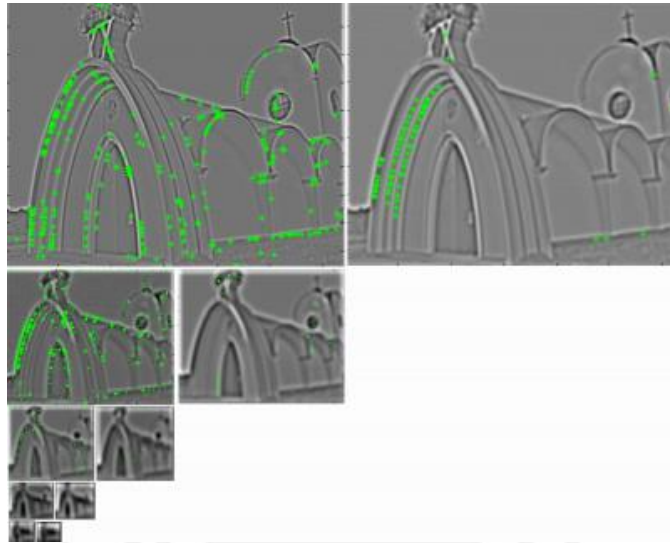


Figura 2.4: Puntos característicos obtenidos [9].

2.2.2 Detección de puntos característicos

A pesar de haber obtenido varios puntos, existe la posibilidad de que se acople ruido en el sistema de obtención de imágenes, provocando distorsiones en la imagen por lo que se generó dos condiciones que permiten descartar algunos puntos que se hallaron anteriormente, pues tal vez un punto es, en realidad el efecto del ruido y no una característica de la imagen. Esas dos condiciones son el bajo contraste y ubicación a lo largo de los bordes.

Para la eliminación de los puntos de bajo contraste se utiliza un proceso de umbralización, en el cual si los puntos son menores a Q serán eliminados, para lo cual Q toma el valor de 0.03 según LOWE y los puntos que se ubican en los borde se eliminan mediante la matriz Hessiana. Luego de aplicar esas dos condiciones se puede apreciar una disminución del número de puntos clave.



Figura 2.5: Reducción de los puntos encontrados, mediante las condiciones [10].

2.2.3 Asignación de orientación

Ahora cada punto tendrá una orientación, la cual estará definida por los puntos que se encuentran a su alrededor. Esto genera en los puntos característicos independencia a la rotación [7]. Para tener un conocimiento de la orientación, se analiza los valores de gradiente y orientación de los pixeles ver ecuaciones 2.3 y 2.4 [8].

$$m(x, y) = \sqrt{[L_{(x+1,y)} - L_{(x-1,y)}]^2 + [L_{(x,y+1)} - L_{(x,y-1)}]^2} \quad \text{Ecuación 2.3}$$

$$\theta(x, y) = \tan^{-1}\left(\frac{L_{(x,y+1)} - L_{(x,y-1)}}{L_{(x+1,y)} - L_{(x-1,y)}}\right) \quad \text{Ecuación 2.4}$$

2.2.4 Descriptor de puntos característico

Se hace uso de una ventana circular gaussiana, para que los efectos de los pixeles más alejados de un punto clave tengan un menor impacto en los pixeles cercanos, se busca la dirección resultante interpolando los tres valores mayores. Una vez usada la ventana gaussiana se hace una distribución en sectores más grandes; es decir de mayor dimensión matricial, con lo que se obtiene un histograma con 8 orientaciones distintas. Con esto logramos que los histogramas se mantengan iguales a pesar de que ocurra una traslación de 4 pixeles. Esto se conoce como robustez, pues el método es útil ante cambios de puntos de vista. Ver figura 2.6.

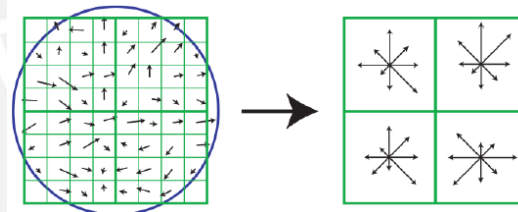


Figura 2.6: Representación de la ventana gaussiana y su generación de histogramas [10].

2.2.5 Emparejamiento

Al tener un descriptor que nos permita saber las principales orientaciones de un punto característico, podremos determinar si en dos imágenes existe similitud. Ver figura 2.7.

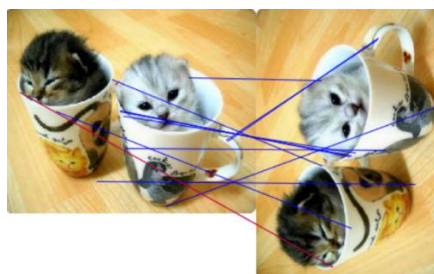


Figura 2.7: Identificación de puntos característicos y emparejamiento de los mismos [11].

2.3 ASIFT (Affine Scale Invariant Feature Transform)

A diferencia de SIFT en esta técnica se usa 6 parámetros, de los cuales 3 son simulados: zoom, ángulo de cámara en latitud y ángulo de cámara en longitud, y los otros son normalizados, los cuales son traslación, rotación y escala.

ASIFT simula todas las vistas de las imágenes posibles en función de los 3 primeros parámetros mencionados. Con esto en cuenta se puede identificar de forma fiable las características de la imagen, que también se hallaban en la técnica SIFT, cuando ha sido sometida a distorsiones. La creación de las simulaciones lleva un coste computacional, lo que lo hace en cierta medida menos eficiente con respecto a SIFT y con la técnica SURF que se explicará más adelante.

En la figura 2.8, se aprecia que el piso rectangular posee una forma trapezoidal y es debido a la proyección, con esto se puede decir que los objetos se pueden deformar.

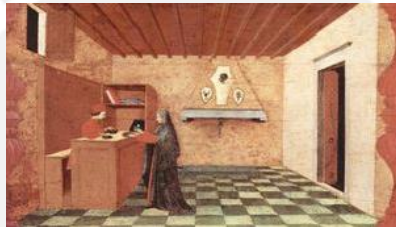


Figura 2.8: Deformación de losetas de piso, debido a la perspectiva de la cámara [12].

2.3.1 Simulación de distorsiones

El primer paso es hacer que la imagen se transforme en todas las posibles distorsiones producidas por la posición de la cámara (latitud y longitud) ver figura 2.9. Por lo que debe ser sometida a una rotación y después a una inclinación de t , según la siguiente relación.

$$t = \left| \frac{1}{\cos \theta} \right| \tag{Ecuación 2.5}$$

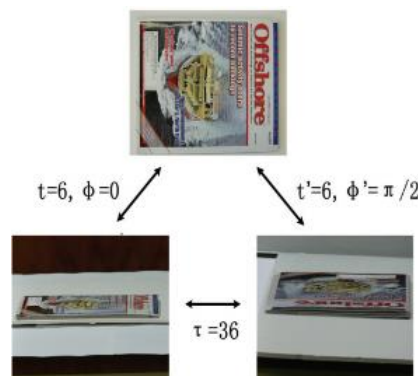


Figura 2.9: Representación de las Inclinaciones de un objeto en distintas imágenes [12].

2.3.2 Emparejamiento

Ahora comparamos entre imágenes, para ello se usa el método ya conocido, SIFT. El resultado se observa en la figura 2.10 donde se aprecia que el método ASIFT permite aun conservar los puntos claves a pesar de que la imagen sufrió una distorsión en cambio el método SIFT no encontró ninguna relación entre las imágenes, pues los puntos entre ambas imágenes no son compatibles por lo tanto fueron eliminados.

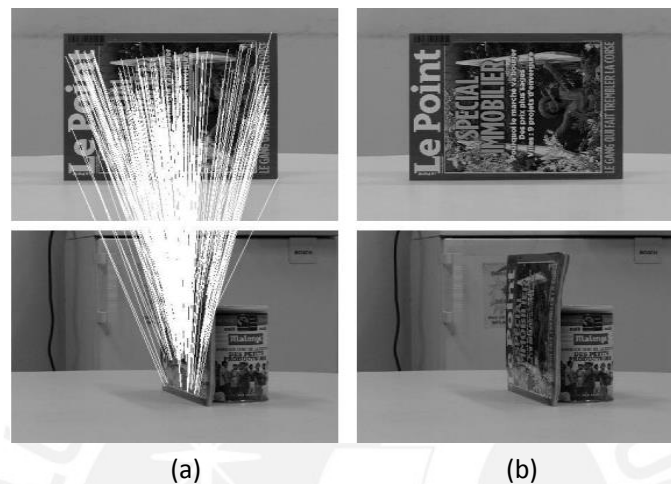


Figura 2.10: Comparación entre los emparejamientos usando SIFT (b) y ASIFT(a) [13].

2.4 SURF (Speeded up Robust Features)

Este algoritmo fue planteado por Herbert Bay, se basa en SIFT pero es más robusto, pues utiliza wavelets y determinantes Hessianas [14]. SURF ofrece la detección de objetos con menor coste computacional y ser más eficiente con respecto a SIFT, en cuanto la realización de cálculos para obtener los puntos clave. Al igual que la técnica SIFT, aquí también se usa la función espacio-escala, lo cual permite reducir el ancho de banda de la imagen original, pues a medida que se va filtrando con un valor que va aumentando, la imagen se va haciendo cada vez más borrosa.

2.4.1 Detección de puntos claves

En este método se aplica como se mencionó la determinante Hessiana, donde se encuentran los valores que representan la posición y escala de los puntos importantes. La matriz Hessiana se muestra en la ecuación 2.6 [14], se definió la imagen como $I(x, y)$. Utilizamos parámetros definidos en la técnica SIFT.

$$H(x, \sigma) = \begin{bmatrix} L_{xx}(x, \sigma) & L_{xy}(x, \sigma) \\ L_{xy}(x, \sigma) & L_{yy}(x, \sigma) \end{bmatrix} \quad \text{Ecuación 2.6}$$

2.4.2 Asignación de orientación

Para esto se debe aplicar el filtro Haar, una vez hecho esto, se crea una asignación de orientación, pues los puntos deberán ser invariantes ante cualquier rotación, según el autor se aplica una rotación de longitud con un valor de $t=4$ [14], el cual se aplica para un punto y lo mismo se hace para los demás puntos vecinos, para luego tener una orientación dominante, también se hace un deslizamiento de $\pi/3$, según el autor [14].

2.4.3 Creación del descriptor

Se crea una región cuadrada centrada en cada punto de interés, con un $t=20s$ ver figura 2.11 y luego se calculan las respuestas ante el filtro Haar, finalmente para cada subregión se suman los resultados [14].



Figura 2.11: Generación de regiones cuadradas centradas en los puntos de interés [14].

En la Figura 2.12 se muestra la aplicación del método SURF y como se enlazan los puntos característicos ante una deformación.

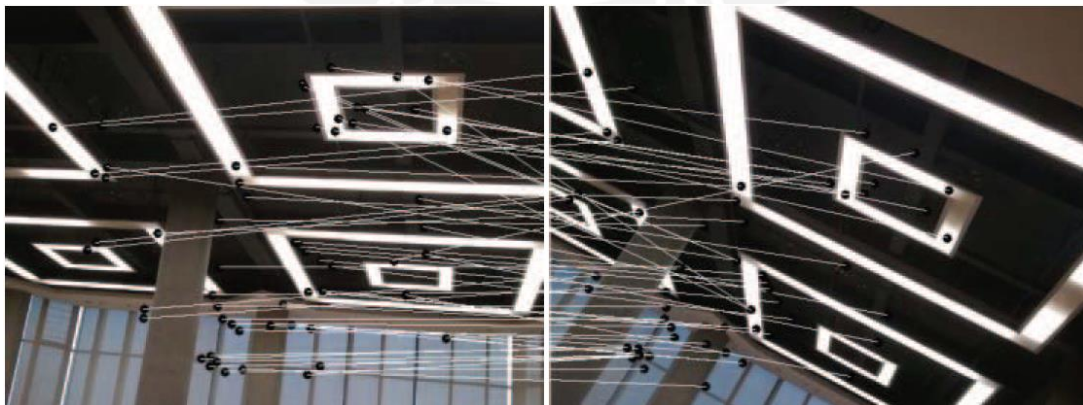


Figura 2.12: Emparejamiento de puntos característicos mediante SURF [15].

Capítulo 3

SELECCIÓN DE TÉCNICA DE RECONOCIMIENTO

En este capítulo se va a seleccionar la técnica de reconocimiento, en base al objeto de interés que queremos detectar, así como que del tipo de escenario que vamos a encontrar en cada video de nuestro banco de videos. Además se presentará el diseño del sistema propuesto en esta tesis, en función de los requerimientos.

El objetivo principal en el desarrollo de esta tesis, es la detección de objetos de interés (vehículos y personas) en imágenes, las cuales han sido tomadas en áreas urbanas. Estas imágenes provienen de videos grabados desde una cámara montada en un UAV, lo que indica que la cámara se encontrará en movimiento.

En el desarrollo de esta tesis, se tuvo en consideración que los objetos de interés, son vistos por la cámara con un ángulo de depresión de 90° (ángulo que va desde la horizontal del UAV y la dirección de la cámara), evitando así el efecto de sobreposición, lo que si ocurre con ángulos de depresión que se encuentran aproximadamente por debajo de 45° . Además con la consideración del ángulo de depresión de 90° , el tamaño de los objetos tampoco genera el efecto de la sobreposición, como se muestra en la Figura 3.3.

Una vez identificado los tipos de escenarios que se encontrarán en las imágenes, y sobre las cual actuará nuestro sistema propuesto, se puede establecer un criterio, con el cual podemos descartar al método ASIFT, ya que este método es muy útil cuando se quiere realizar la detección de puntos característicos de un objeto que se encuentra en una imagen que ha sido deformada por cambio en la posición de la cámara, como se mostró en la Figura 2.9. Con esto en cuenta lo descartamos, ya que en las imágenes que analizaremos ninguna posee esa característica de deformación. Entonces solo se considerará comparar los métodos SIFT y SURF.

Al final del capítulo se tendrá conocimiento de la técnica seleccionada, así como el esquema del diseño del sistema de detección en base a las condiciones que se presentan en imágenes desplazadas por el movimiento de la cámara ubicada en el UAV.

3.1 Análisis comparativo entre las técnicas de reconocimiento

A continuación se procederá a realizar pruebas con los métodos SIFT y SURF sobre algunos fotogramas extraídos de nuestro banco de datos, con el fin de seleccionar el método adecuado para la detección de objetos de interés en esta tesis.

3.1.1 Aplicación del método SIFT

Se aplicará el método SIFT a los siguientes fotogramas, extraídos de nuestro banco de imágenes y los resultados se colocan en la Tabla 3.1. Este método se simulará dos veces, para apreciar si existe alguna variación en cuanto al número de puntos característicos. Ver la Figura 3.1, donde los puntos característicos encontrados por SIFT, se encuentran representados por círculos de color azul.

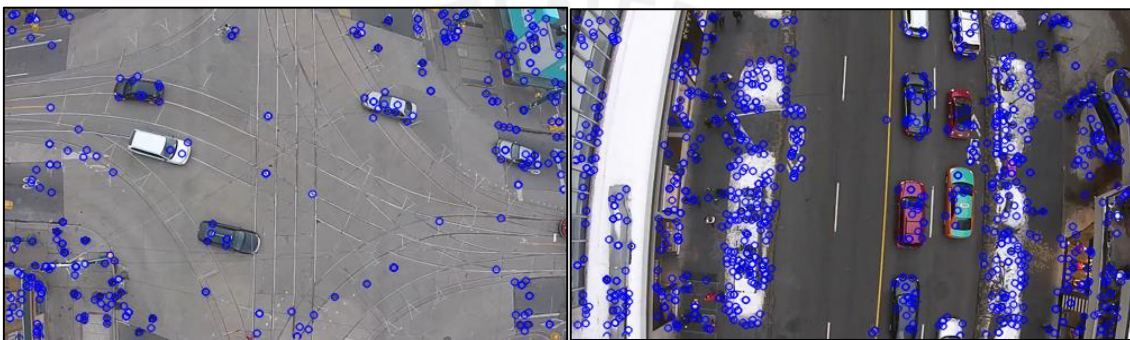


Figura 3.1: Puntos característicos hallados mediante el método SIFT.

3.1.2 Aplicación de método SURF

Al igual que en el punto 3.1.1, en este caso se simulara el método SURF dos veces, sobre las mismas imágenes utilizadas en el figura tal, los puntos característicos se encuentran representados por pequeños círculos de color azul.

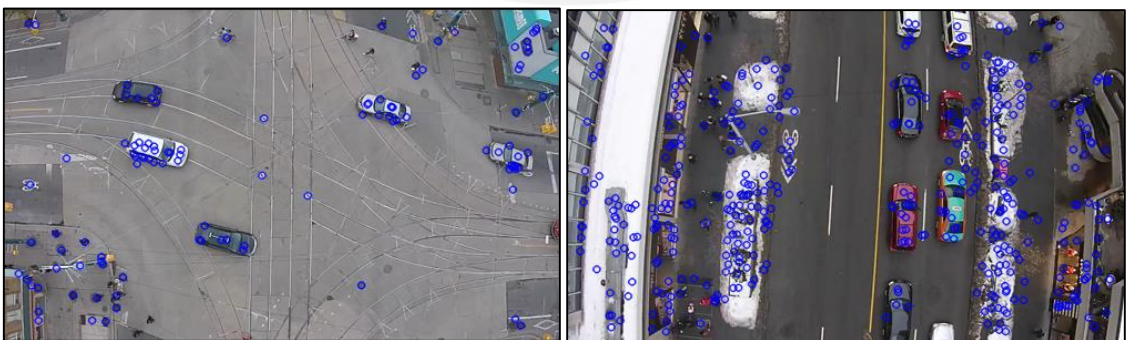


Figura 3.2: Puntos característicos hallados mediante el método SURF.

Según lo visto en 3.1.1 y en 3.1.2, SIFT y SURF obtienen puntos característicos de las imágenes invariantes a escala y orientación. En ambos métodos se obtiene un vector descriptor por cada punto característico encontrado, aunque en ambos se calcula de distinta manera, como se explicó en el capítulo 2.

Una de las principales diferencias entre estos métodos, es la manera en que se almacena la información. En SIFT se almacena la posición, la escala y la orientación, ya que existe la posibilidad que en un mismo punto (x, y) se da la presencia de varios puntos característicos con distintas escalas y orientaciones. Por otro lado en SURF, en un mismo punto (x,y) solo se encuentra un único punto característico, por lo que no es necesario almacenar la escala ni la orientación.

En la comparación entre estos métodos, se tuvo en consideración el tiempo que toma en ejecutarse cada método, así como el número de los puntos característicos que se obtiene, cabe mencionar que las pruebas realizadas se hicieron sobre un procesador Intel core i5 a 2.7GHz.

Durante las pruebas se constató el menor tiempo de ejecución que le toma al método SURF en comparación al tiempo del método SIFT, como se aprecia en la tabla 3.1. Pero en cuanto al número de puntos característicos encontrados, el método SIFT halló un número mayor de puntos que a los encontrados por SURF, como también se muestra en la Tabla 3.1.

La selección de uno de estos métodos dependerá de que es lo que se busca, ya que SURF, nos permite hallar los puntos característicos con menor tiempo y SIFT permite detectar puntos característicos que son robustos a cambios de rotación de la imagen.

El desarrollo del sistema de detección de personas y vehículos propuesto en esta tesis, no es un sistema que trabaje en tiempo real, sino por el contrario, se va a trabajar en base a los videos grabados desde una cámara montada en un UAV y procesarlos en un computador, dicho video puede contar con un gran número de fotogramas, el cual dependerá de su duración, por lo que se prioriza el tiempo de ejecución del método. Con esto en cuenta se optó por usar el método SURF. En un futuro se buscará desarrollar un sistema que trabaje en tiempo real, con el cual se pueda crear aplicaciones de control de flujo vehicular y peatonal.

Tabla 3.1: Comparación entre los resultados obtenidos por métodos SIFT y SURF.

		SIFT		SURF	
		tiempo (s)	Puntos característicos	tiempo(s)	Puntos característicos
Fotograma 1	por 1ra vez	0.165	304	0.039	107
	por 2da vez	0.171	304	0.037	107
Fotograma 2	por 1ra vez	0.1879	863	0.0485	417
	por 2da vez	0.1811	863	0.045	417

3.2 Diseño del Sistema propuesto

El sistema que se propone, se basa en los requerimientos que se desea cumplir para la detección de objetos de interés (personas y vehículos). Esto implica desarrollar una solución a la problemática de trabajar con una secuencia de imágenes que se encuentran en constante cambio de fondo, que es lo que ocurre con una cámara en movimiento. En la figura 3.3, se muestra el problema de grabar un video o tomar imágenes con una cámara en movimiento.

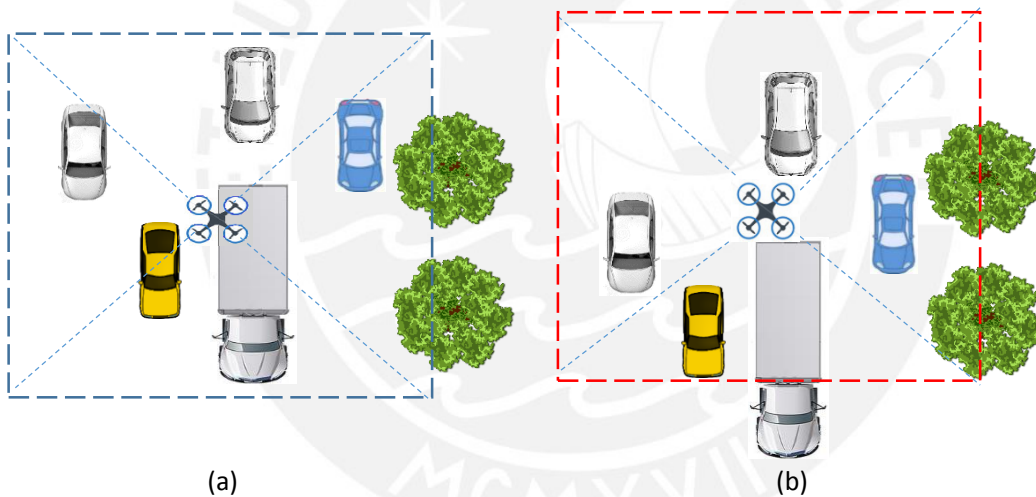


Figura 3.3: Representación del problema del desplazamiento del UAV en las fotogramas tomados, (a) fotograma previo, (b) fotograma actual.

Se observa como el UAV se va desplazando, así también lo hace la imagen que toma, representado por un recuadro con líneas entrecortadas, con lo cual explicamos que la imagen que toma siempre cambia, pero no es lo único que cambia, algunos objetos, como los autos también se mueven, aunque también hay objetos que no se mueven, como los árboles, los cuales se mantienen en una posición fija. Estos objetos estáticos son los que permitirán encontrar una relación entre ambas imágenes, mediante el emparejamiento de los puntos característicos. Luego de emparejar los puntos característicos, se procede a alinear los fotogramas, mediante una transformación. Este proceso consiste en alinear un fotograma a otro, y posteriormente podremos detectar los objetos que se mueven mediante el procesamiento de imágenes.

En la Figura 3.4, se muestra el diseño del sistema de detección de objetos de interés (personas y vehículos) y sus respectivas etapas o procesos que lo componen.

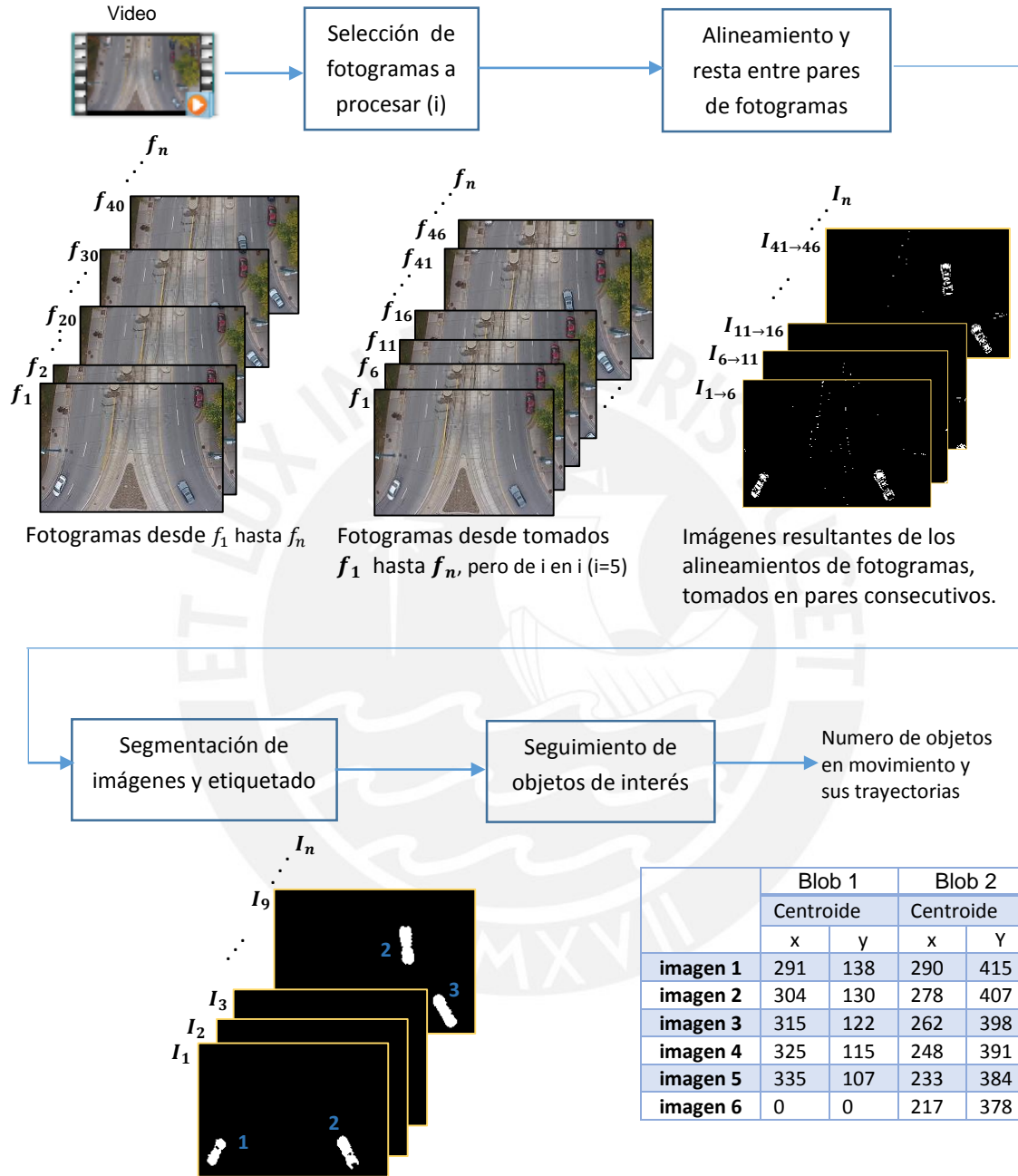


Figura 3.4: Proceso para la detección de objetos de interés.

A continuación se explicará a grandes rasgos el proceso que se sigue para la detección de los objetos de interés para esta tesis, personas y vehículos; se dan más detalles en el próximo capítulo.

El bloque de separación de fotogramas, nos permite establecer cuántos y cuáles de los fotogramas se van a procesar (“tasa de muestreo” de fotogramas seleccionado por nosotros). Este parámetro controla que pares de fotogramas se van a alinear. En la Figura 3.4, se muestra que el parámetro posee el valor de 5, la razón del por qué de este valor se explica en el punto 4.3

Una vez determinado los fotogramas a procesar, la siguiente etapa es la del alineamiento de fotogramas, la cual toma dos de ellos y mediante una transformación, cambia la perspectiva de un fotograma. Con esto se pueda encajar o empalmar un fotograma sobre otro siguiente. En la Figura 3.5 se muestra los fotogramas de la figura 3.3, luego de alinearlos. Una vez se encuentren alineados se procede a restar, lo cual dará origen a una imagen resultante que muestra los objetos que están en movimiento.

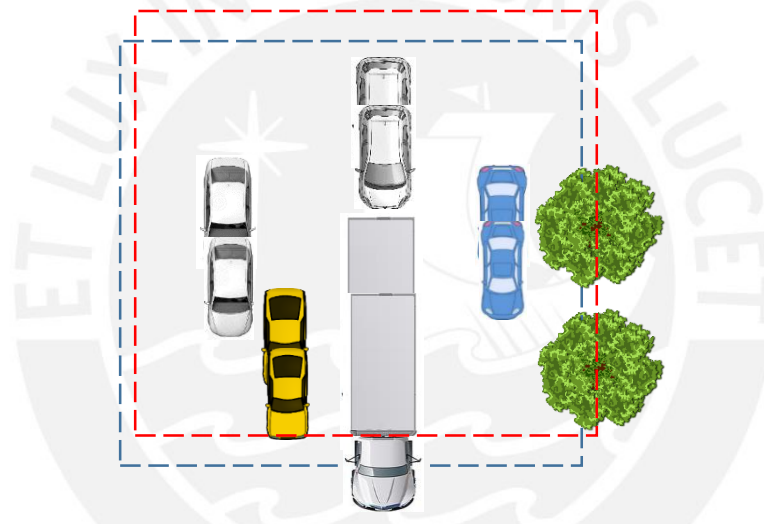


Figura 3.5: Ejemplo de alineamiento entre dos fotogramas.

La etapa de segmentación de imagen, nos permitirá extraer los objetos de interés (vehículos y personas), separándolas de elementos que pueden ser ruido u objetos que no están en movimiento, pero que están presentes, debido a una inadecuada alineación entre fotogramas. En esta etapa también se realizará el etiquetado de objetos encontrados, lo cual nos será de utilidad en el seguimiento de objetos.

Por último se encuentra la etapa de seguimiento de los objetos de interés (blobs), en esta etapa nos encargamos de contar los blobs y llevar un registro de las ubicaciones de todos los blobs encontrados a lo largo de un video; es decir, se almacenará las coordenadas de los objetos que están presentes en una serie de imágenes así como también se tendrá el registro de los objetos que ya se encuentran fuera de escena.

Capítulo 4

IMPLEMENTACIÓN DEL SISTEMA

En este capítulo se presenta la implementación del sistema propuesto para la detección de personas y vehículos, para lo cual se trabajará con un banco de datos (galería de imágenes). Nos centraremos en la detección y conteo de objetos de interés que se encuentren en movimiento a lo largo de la secuencia de imágenes tomadas desde el UAV. Finalmente se realizarán pruebas al sistema con los diversos videos y en base a sus resultados se establecerán conclusiones acerca de la eficiencia del sistema propuesto.

4.1 Banco de datos

A continuación se detallará la galería de imágenes con las que se cuentan, a partir de los videos extraídos y el proceso que se seguirá para crear dicha galería.

4.1.1 Videos

Los videos utilizados en esta tesis fueron obtenidos a partir de la web, y así se logró tener un banco de pruebas variado respecto a los tipos de formato de video, tales como wmv, avi y mp4. Es importante indicar que en los videos seleccionados no existe la presencia de traslape o sobreposicion entre vehículos, debido al ángulo de depresión de la cámara ubicada en el UAV, como si ocurre en la Figura 4.1. Además cabe resaltar que los videos, corresponden a diferentes horas de la mañana; es decir, secuencias de videos con distintos niveles de luminosidad.

Los videos obtenidos cuentan con una duración variada, que van desde los 75 segundos hasta 8 segundos, y en todos se muestra objetos que están tanto en movimiento como estáticos. En la Figura 4.2 se muestran los videos con los que se va a trabajar.



Figura 4.1: Imagen con presencia de traslape entre los vehículos [16].

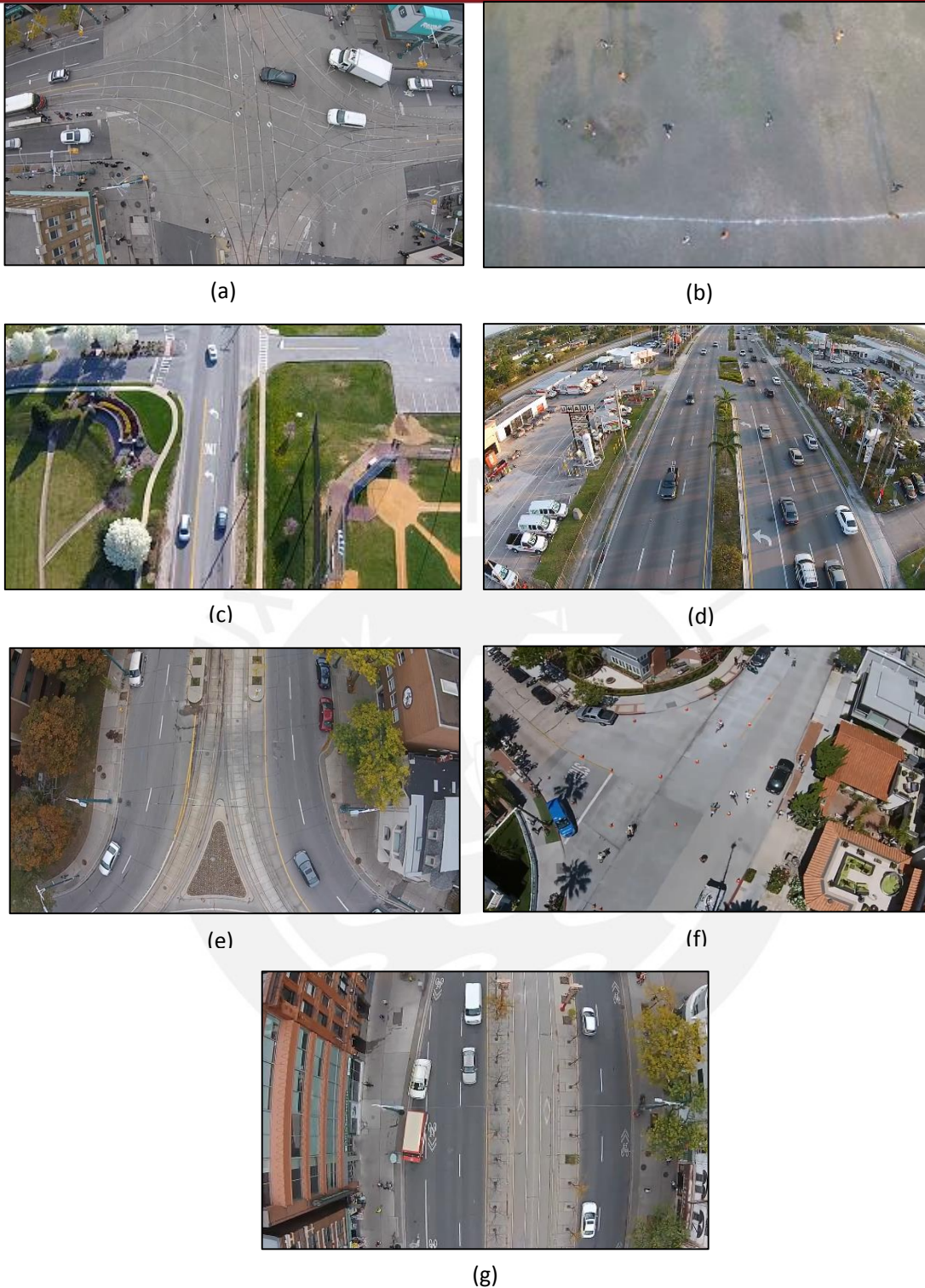


Figura 4.2: Galería de videos con la que se trabajará.

En la Figura 4.2 se muestra los videos con los que se cuenta y servirán para generar nuestra galería de imágenes y a partir de ahí obtener los un banco de imágenes o fotogramas las cuales puedan ser procesadas por nuestro sistema propuesto. Se debe indicar que los videos poseen características distintas, tales como presencia de árboles y edificaciones (4.2c, 4.2d, 4.2e, 4.2f y 4.2g), ángulos de depresión de 45° de la cámara

(4.2d), así como también solo presencia de objetos de interés (vehículos y personas) para esta tesis (4.2a, 4.2b).

4.1.1.1 Factores controlables y no controlables

Puesto que en los videos que se trabajaron en esta tesis no existe un protocolo o procedimiento en la toma de video, se determinaron factores que deben controlarse en futuras grabaciones de videos para lograr así un mayor desempeño del sistema planteado en esta tesis.

Tabla 4.1: Factores controlables y no controlables en el procesamiento de imágenes.

factores		Descripción
Iluminación	Factor no controlable	Se aprecia que en algunos videos, se da la presencia de sombras o exceso de iluminación debido a reflejos del sol.
Ángulo de depresión ¹	Factor controlable	La cámara debe estar ubicada de tal forma que el video sea tomado con un ángulo de depresión de 90°, para evitar sobreposiciones o traslapes de autos, y de personas.
Posición del UAV	Factor controlable	El UAV debe ubicarse por encima de los objetos a detectar, a una altura aproximada de 10 metros y no realizar movimientos de acercamiento o alejamiento.
Estado del tiempo	Factor no controlable	Condiciones de lluvia o neblina afectaran en la calidad del video, agregando ruido y empeorando la iluminación.
Flujo de vehículos	Factor no controlable	No se puede saber si en un momento dado habrá movimiento de objetos (vehículos y personas).

4.1.2 Conteo manual de autos y personas

Se realizará el conteo manual de los objetos de interés para esta tesis, personas y vehículos, con el fin de establecer la eficiencia de nuestro sistema, por lo que en la siguiente tabla se aprecia los números de los objetos encontrados que están presentes en cada video, y se compararán con los resultados que obtendremos al aplicar nuestro sistema y así emitir conclusiones.

¹ Tanto el ángulo de depresión de la cámara y posición del UAV, son parámetros que pueden controlarse siempre y cuando se cuente con un sistema de control robusto y que soporte cualquier cambio en el ambiente, tal como ráfagas de viento que provocarían un cambio en los parámetros por un instante, logrando que la grabación posea instantes donde no se realizó una adecuada toma de datos (cambio repentino entre una imagen y otra).

Tabla 4.2: Conteo manual de los objetos en movimientos en el banco de videos.

Banco de videos	Conteo manual		
	Número de vehículos en movimiento	Número de vehículos estacionados	Número de personas
Toronto_35s.mp4 (4.2a)	26	0	--
Soccer1.wmv (4.2b)	0	0	15
Via.mp4 (4.2c)	6	0	0
avenida_18s.mp4 (4.2d)	34	0	0
Toronto_58s.avi (4.2e)	11	11	12
People_beach2.avi (4.2f)	1	8	13
Toronto_75s.avi (4.2g)	16	11	37

4.1.3 Extracción de fotogramas

Los fotogramas o frames, son una secuencia de imágenes que conforman un video. En el desarrollo de esta tesis, los videos con los que se trabajarán cuentan con una tasa de 30 frames/segundo, pero estos videos no poseen las mismas dimensiones entre sí, por lo que se vio necesario un proceso de extracción y redimensión de frames; y se optó por trabajar con imágenes de 640x360 pixeles, como se muestra en la Figura 4.3

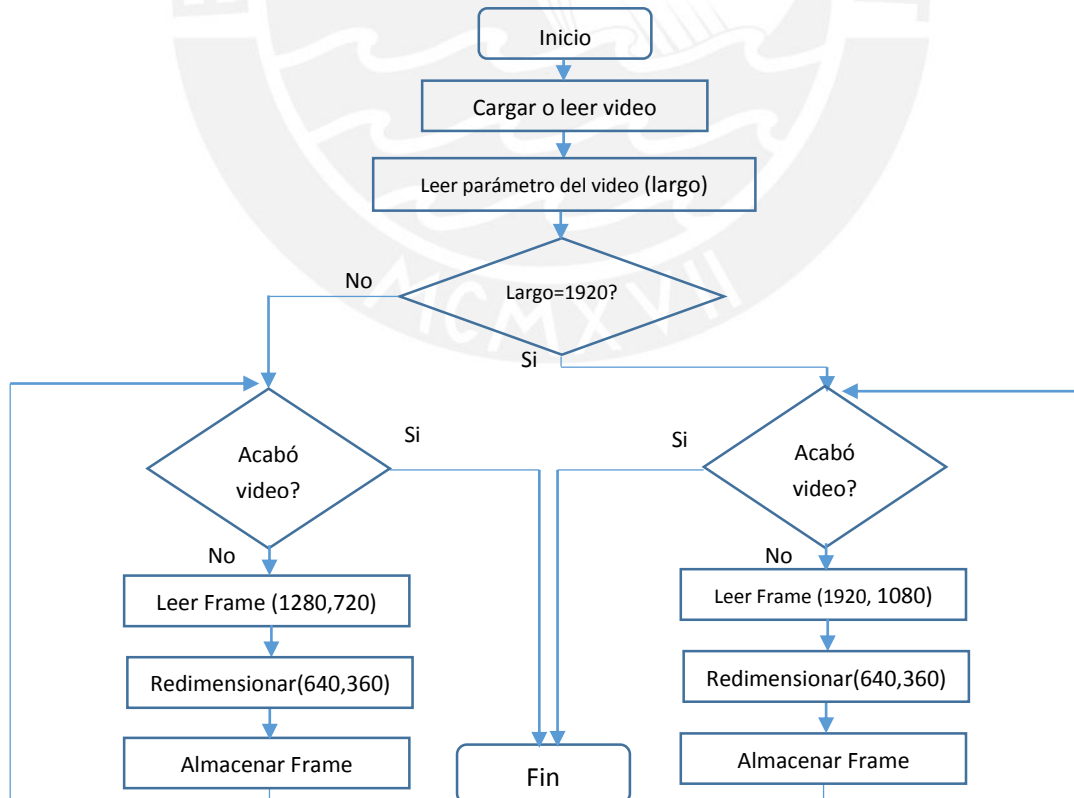


Figura 4.3: Diagrama de flujo del proceso de extracción de fotogramas.

4.2 Alineamiento entre frames

A continuación se explicará la idea principal que se va a desarrollar en esta tesis, para lograr detectar objetos en movimiento (vehículos y personas).

En la Figura 4.4 se muestra una manera de detectar objetos de interés y es realizando la resta entre una imagen que corresponde al fondo (Figura 4.4a) y otra que representa a la secuencia de imágenes (Figura 4.4b), donde aparecen los objetos que se mueven, al realizar esta resta, se obtiene solo el objeto en movimiento (Figura 4.4c).



Figura 4.4: Extracción de objeto en movimiento.

Pero esto no es aplicable en el uso de cámaras en movimiento, como ocurre en un UAV, ya que debido al movimiento mismo del UAV, no existirá una imagen o fotograma que represente el fondo, pues este siempre será cambiante. Para una mayor explicación, ver Figura 4.5

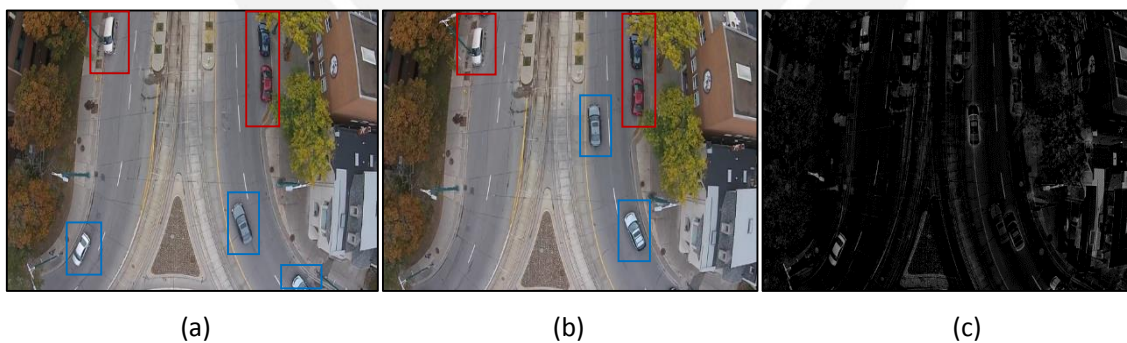


Figura 4.5: Resta entre imágenes tomadas desde un UAV.

En este ejemplo, la muestra fue tomada del video *Toronto_58s.avi*, donde aparecen los autos en movimiento y estacionados, los cuales denotamos mediante los recuadros de color azul y rojo respectivamente. Como se aprecia, el fotograma previo está representado por 4.5a y el fotograma actual por 4.5b.

Se esperaría que en la Figura 4.5c, la imagen de la resta, solo quede zonas que representen a los autos que se han desplazados, pero nos encontramos que también se da la presencia de los autos estáticos, contornos de edificios y residuos de segmentos de los árboles, esto debido a que el UAV también se encontraba en movimiento, por muy ligero que fuese, afectando la resta de imágenes.

La solución que se planteó fue la del alineamiento, mediante el cual el fotograma previo se alineará con el fotograma actual de tal forma que cuando se realice la resta o sustracción, solo aparezcan zonas no negras que representen el movimiento de los objetos de interés (vehículos y personas).

Para realizar el un correcto alineamiento, se hará uso de los puntos característicos o Keypoints de los fotogramas previo y actual.

4.2.1 Extracción de puntos característicos mediante la técnica de detección

Se extraerán los puntos característicos de los dos fotogramas para luego relacionarlos mediante la matriz de homografía (matriz de transformación), la cual nos permitirá alinear una imagen respecto a la otra; es decir, el fotograma previo se alineará al fotograma actual.

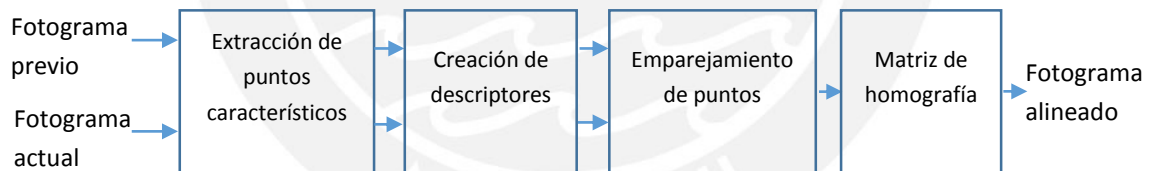


Figura 4.6: Proceso de alineamiento de fotogramas.

En la Figura 4.6 se muestra el proceso que se sigue para hallar los puntos característicos mediante la técnica SURF, seleccionada en el capítulo 3, una vez hallados esos puntos se proseguirá con la creación de los descriptores, los cuales contendrán la información de la vecindad de estos puntos, luego se relacionarán los puntos comunes entre ambas imágenes, para finalmente mediante una transformación proyectiva se pueda calcular la matriz de Homografía, la cual permitirá alinear el fotograma previo al fotograma actual.

4.2.1.1 Extracción de puntos característicos

En esta etapa se extraerán los puntos característicos mediante la técnica SURF, y que permitirán hallar la matriz de homografía, la cual alineará a los fotogramas.

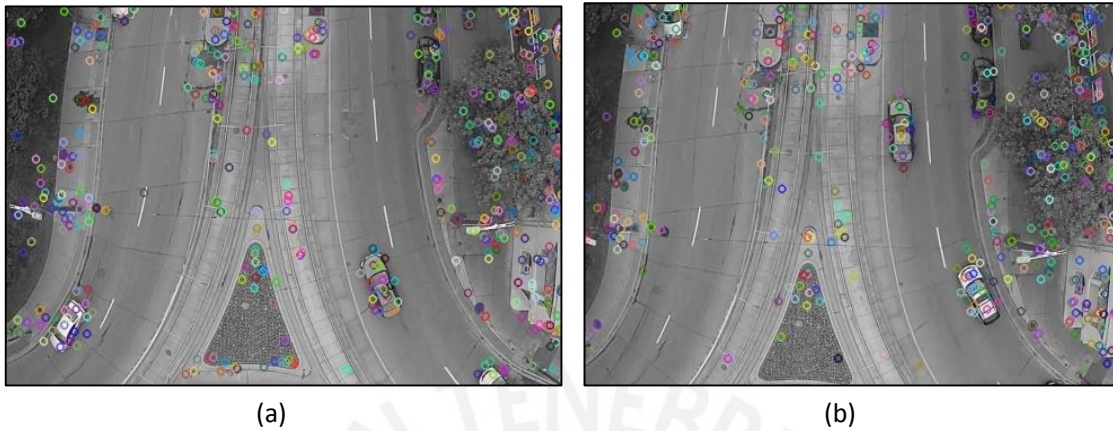


Figura 4.7: Representación de los puntos característicos del fotograma previo 4.7a y fotograma actual 4.7b.

4.2.1.2 Creación de descriptores

Para la creación de los descriptores, el algoritmo SURF, hace uso de una integración de las imágenes, donde se obtiene una respuesta ante el filtro Haar-wavelet [14] en la vecindad de los puntos característicos.

La descripción de los puntos característicos consiste en extraer información sobre la región que rodea el punto de manera simplificada y así poder compararla fácilmente con otros descriptores.

En la Figura 4.8 se muestra la orientación del descriptor basada en la información de una zona circular alrededor del punto de interés. Así se obtiene el descriptor SURF de un fotograma, lo mismo se realiza para el fotograma siguiente.



Figura 4.8: Representación del conjunto de descriptores SURF de una imagen de prueba.

4.2.2 Emparejamientos de puntos característicos

El emparejamiento consiste en identificar los puntos característicos de cada imagen y buscar cuales son iguales entre sí. Cuando se tenga todos los emparejamientos, se podrá calcular la transformación del fotograma previo, para que pueda alinearse con el fotograma actual.

Para la etapa de emparejamiento, entre los puntos característicos de dos imágenes (fotograma previo y actual), se hace uso del algoritmo implementado en la librería FLANN (Fast Library for Approximate Nearest Neighbors) de OpenCV. Este algoritmo hace uso de los descriptores encontrados en el paso anterior por SURF, y su función es encontrar el emparejamiento de los descriptores entre ambas imágenes mediante la similitud de información entre los puntos característicos. FLANN reduce en un orden de magnitud el tiempo de computación con respecto a los demás métodos semejantes [17]. Al encontrarse dentro de una biblioteca de OpenCV, su uso será inmediato.



Figura 4.9: Proceso de emparejamiento [18].

Además este método permite el acceso a la distancia euclídea obtenida de la comparación de ambos descriptores de imágenes. Esto nos permite trabajar solo con aquellos puntos, con distancias menores (lo que indican gran similitud). Para discriminar los puntos que no cumplen con el emparejamiento adecuado, se extrae la distancia entre los puntos más cercanos, una vez se tiene esta distancia, se eliminan todos aquellos puntos que no tengan ningún otro punto a menos de 3 veces la distancia mínima. Después se unen aquellos emparejamientos que si cumplen. Una vez que se obtienen todos los emparejamientos buenos, estos se almacenan en un vector de puntos.

4.2.3 Transformación de fotograma previo

Luego de obtener los emparejamientos, se procederá a realizar el alineamiento entre el fotograma previo y actual, para lo cual se realizará la operación de transformación. Esta operación se llevará a cabo sobre el fotograma previo, con el fin de lograr que se alinee al fotograma actual.

Para esto se debe hallar una matriz que permite su alineación, esta es conocida como la matriz de homografía.

4.2.3.1 Homografía

En procesamiento de imágenes el término homografía está definido como una transformación proyectiva o de perspectiva que determina una correspondencia entre los puntos de un plano a otro, en nuestro caso se hablara de imágenes como planos. “Dado un conjunto de puntos p_i en el plano \mathbb{R}^2 , y el conjunto correspondiente de puntos p'_i de igual forma en el plano \mathbb{R}^2 , la homografía realiza el cálculo de la transformación proyectiva que toma de cada elemento de p_i a p'_i ” [19].

Se puede decir que una homografía es un mapeo invertible de puntos en el plano proyectivo \mathbb{R}^2 . También dan una definición algebraica de homografía probando el siguiente teorema: un mapeo de $\mathbb{R}^2 \rightarrow \mathbb{R}^2$ es una proyectividad si y solo si existe una matriz 3×3 H , tal que para cualquier punto en \mathbb{R}^2 representado por el vector X existe su punto mapeado X' y es igual a HX . Esto nos dice que, para calcular la homografía que mapea cada x_i a su correspondiente x'_i es suficiente calcular la matriz de homografía H .

4.2.3.2 Transformación de perspectiva o Homográfica

Las transformaciones de perspectiva (Figura 4.10) ofrecen mayor flexibilidad, ya que logran convertir una imagen rectangular en una imagen trapezoidal, como los paralelogramos son también trapecios. Existe otro tipo de transformación, tal como la transformación afín, la cual cambia la forma rectangular de una imagen a la de un paralelogramo, por lo que este tipo de transformación puede ser considerado como un subconjunto de las transformaciones de perspectiva.

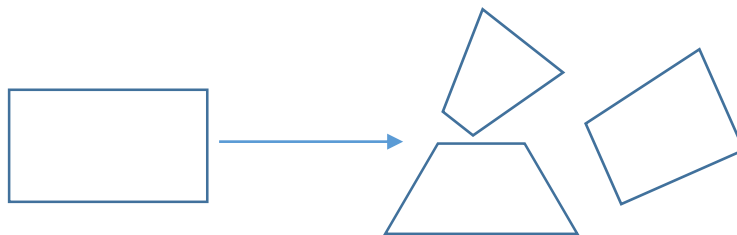


Figura 4.10: Transformación de perspectiva.

4.2.3.3 Metodología y cálculo de la matriz de homografía

Dado que la homografía puede relacionar una imagen con otra, las cuales son corresponden a imágenes tomadas desde distintas posiciones, debido al movimiento o cambio del enfoque de la cámara, entonces hay muchas situaciones en visión por computador o procesamientos de imágenes, donde la estimación de una homografía puede ser requerida. En esta sección exploraremos el cálculo de la matriz de homografía, mediante la ecuación 4.1 que se explicó anteriormente.

$$x' = H \cdot x \quad \text{Ecuación 4.1}$$

$$H = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} \quad x' = \begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} \quad x = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \text{Ecuación 4.2}$$

$$\begin{bmatrix} u' \\ v' \\ 1 \end{bmatrix} = \begin{bmatrix} a_{00} & a_{01} & a_{02} \\ a_{10} & a_{11} & a_{12} \\ a_{20} & a_{21} & a_{22} \end{bmatrix} \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \quad \text{Ecuación 4.3}$$

Como se indicó en el punto 4.2.3.1 que para cada x_i existe su correspondiente x'_i y según la ecuación 4.1, se aprecia que existirá tantas matrices de homografía como emparejamientos de puntos característicos haya, por lo que al final, la matriz de homografía resultante, será el promedio de todas las calculadas.

En esta tesis para cálculo de la matriz de homografía se optó por usar la función **cvFindHomography** de OpenCV. Esta función se describe en el **anexo X**. los parámetros de esta función son: conjuntos de puntos característicos del fotograma previo y actual, y el tipo de iteración a usar, en este caso RANSAC [].

Una vez obtenido la correspondencia de los puntos característicos por el algoritmo SURF, se espera que algunas de estas correspondencias no sean correctas (emparejamientos erróneos). El algoritmo RANSAC hará frente a este problema mediante la estimación de un modelo, y determinará un conjunto de inliers que sean consistentes a esta estimación (emparejamientos correctos) y descartará el conjunto de outliers (emparejamientos erróneos).

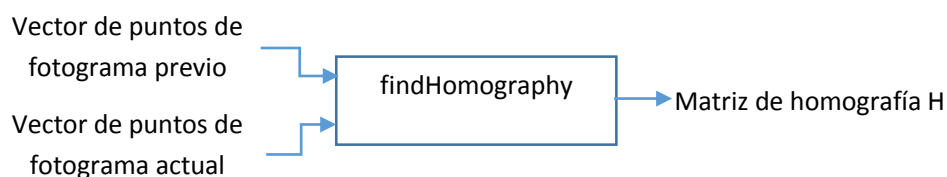


Figura 4.11: Etapa de obtención de matriz de Homografía.

A continuación se detalla un ejemplo de su uso, para lo cual se utilizó los fotogramas de la figura 4.7, los cuales se encuentran separados por lapso de tiempo de 1.408 segundos, se aplicó el proceso de la Figura 4.11, obteniéndose datos importantes, como el total de emparejamientos entre los descriptores de ambos fotogramas, los emparejamientos que fueron seleccionados como correctos luego de aplicar la función *findHomography* (ver Figura 4.12) y la matriz de homografía.

```

jhon@ubuntu: ~/Desktop/OpenCV/opencv-2.4.9/samples/cpp
File Edit View Search Terminal Help
.cpp -o video10 `pkg-config --cflags --libs opencv`
jhon@ubuntu:~/Desktop/OpenCV/opencv-2.4.9/samples/cpp$ ./video10 t
o010.bmp to050.bmp
se cargaron las imagenes
se aplico SURF
se calcularon los extractores
Emparejamientos totales:      640
-- Max dist : 0.623468
-- Min dist : 0.035263
emparejamientos adecuados    114
H=
[0.992708121816704, 0.02628842717057637, -15.75983257987725;
 0.00208411926582852, 1.02698753707866, 13.98621961466505;
 -3.788427105147998e-05, 9.333460205199686e-05, 1]
  
```

Figura 4.12 Matriz de homografía obtenida entre los fotogramas de la figura 4.7.

Una vez obtenida la matriz de homografía, se procederá con la transformación perspectiva del fotograma previo, de tal forma que se alinee con el fotograma actual, para lo cual se hará uso de la función *WarpPerspective* implementada en OpenCV.

Para concluir con la etapa de transformación del fotograma previo, se muestra la Figura 4.13, donde se aprecia los emparejamientos que fueron considerados como correctos, además, se dibujó un marco de color verde sobre el fotograma actual, donde dicho marco representa los puntos del contorno del fotograma previo, logrando así alinearlos y establecer que parte del fotograma actual, es el fotograma previo (ver Figura 4.14).

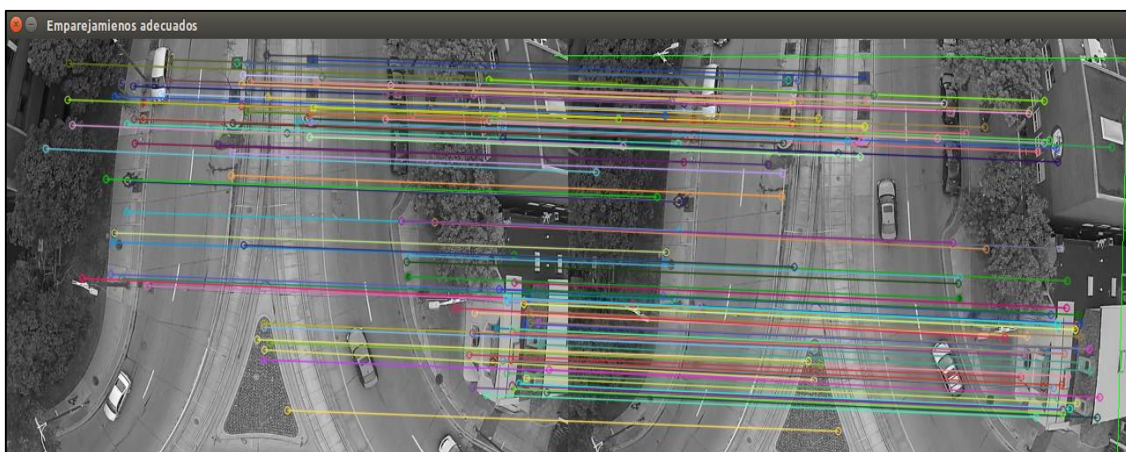


Figura 4.13 Representación de los emparejamientos correctos entre el fotograma previo y actual.



Figura 4.14: Fotograma previo alineado (Usando la matriz de transformación) al fotograma actual.

4.2.4 Resultado del alineamiento

Una vez que se haya obtenido el alineamiento del fotograma previo, se procede con la detección de los objetos en movimiento, para lo cual se realiza la operación de resta de imágenes, entre el fotograma actual y la transformación del fotograma previo. Un ejemplo de esta etapa, se aprecia en la Figura 4.15, donde la resta de los fotogramas sin ninguna transformación, resultó en una simple resta que muestra los objetos estáticos y en movimiento, pero esto no ocurre en la Figura 4.16, en este caso solo se da la presencia de objetos en movimiento mas no los estáticos, tales como los autos estacionados, y esto debido a la alineación del fotograma previo, mediante la matriz de homografía (Transformación).

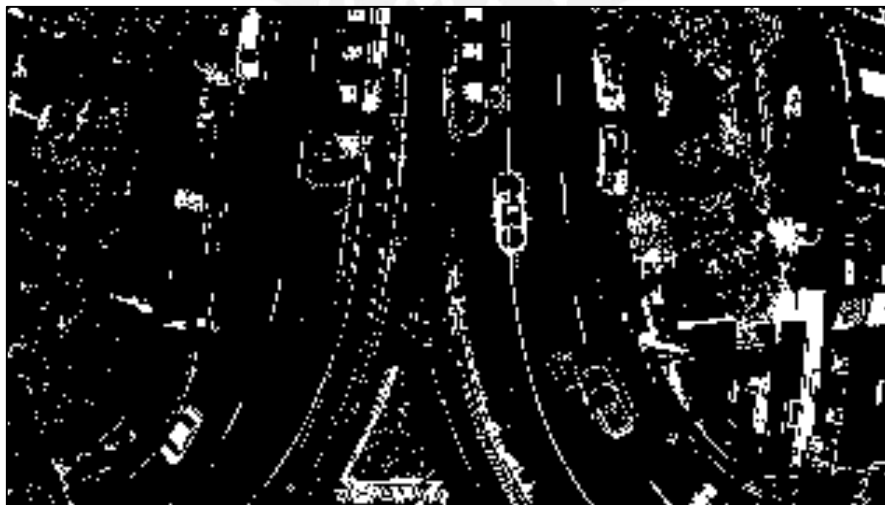


Figura 4.15: Resta de los fotograma previo y actual, sin aplicar alineamiento.

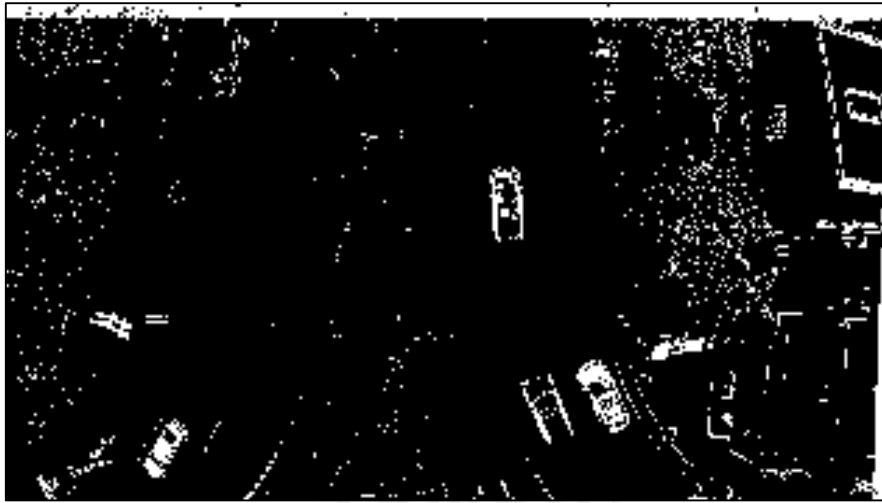


Figura 4.16: Resta de los fotograma previo y actual, aplicando el alineamiento.

4.3 Consideraciones

Como se ha podido observar a lo largo de esta tesis, la detección de objetos de interés mediante el uso del alineamiento de fotogramas ha sido una gran ayuda, cuando se tiene que trabajar con imágenes que han sido tomadas desde una cámara en movimiento. Pero esto no es suficiente para realizar una detección exitosa de un objeto, también se debe tener en cuenta ciertos criterios que están en función a las características mismas de los objetos a detectar. A continuación se detallan conceptos que permiten realizar un mejor alineamiento entre fotogramas.

4.3.1 Medidas de objetos a detectar

Las medidas de los objetos a detectar están en función de la posición de la cámara del UAV, puesto que no se tendrán las mismas dimensiones para una persona o vehículo, en imágenes tomadas con un ángulo de depresión de 10° u otros ángulos. Para el desarrollo de esta tesis se consideró trabajar con ángulos de depresión cercanos a 90° (vertical) de preferencia.

En base al banco de videos con el que se cuenta, se estimó la medida en pixeles de los objetos a detectar, como se detalla a continuación en la Figura 4.17. Se mencionó que los videos fueron grabados con un ángulo de depresión de la cámara de 90° , por lo que las dimensiones de los objetos, serán tomados a partir de la vista horizontal, es así que se estimó un aproximado de 65 pixeles de largo y 25 pixeles de ancho en el caso de un auto (ver Figura 4.17a), considerando una longitud real del auto en 4 y 1.8 metros respectivamente. En el caso de las personas, se consideró dimensiones de 10 y 12 pixeles (ver Figura 4.17b).

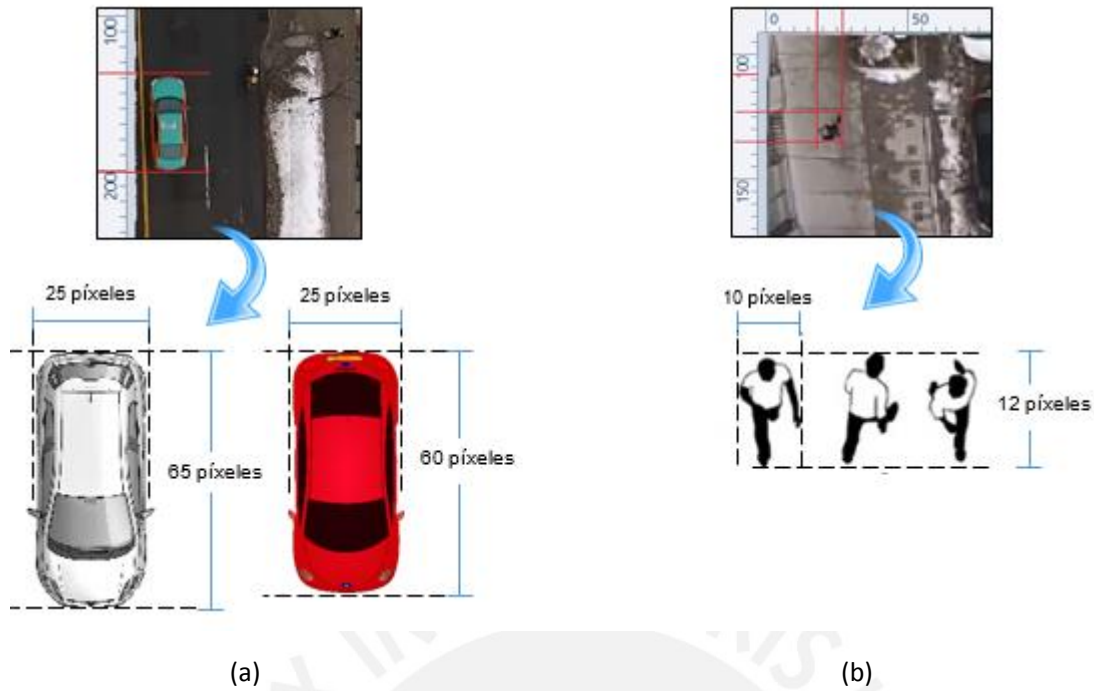


Figura 4.17: Medidas de los objetos a detectar.

$$r = \frac{\text{número de píxeles}}{\text{distancia en metros}} \quad \text{Ecuación 4.4}$$

$$r_a = \frac{65 \text{ píxeles}}{4 \text{ m}} = 16.25 \frac{\text{pixel}}{\text{m}}$$

Al hallar una proporción entre la distancia en metros del largo de un vehículo y su longitud en píxeles, se puede estimar la distancia desplazada de un auto entre una fotograma y otro.

4.3.2 Velocidad de desplazamiento

Se analizará solo la velocidad de los vehículos, puesto que estos son los que tendrán un mayor desplazamiento en comparación a las personas.

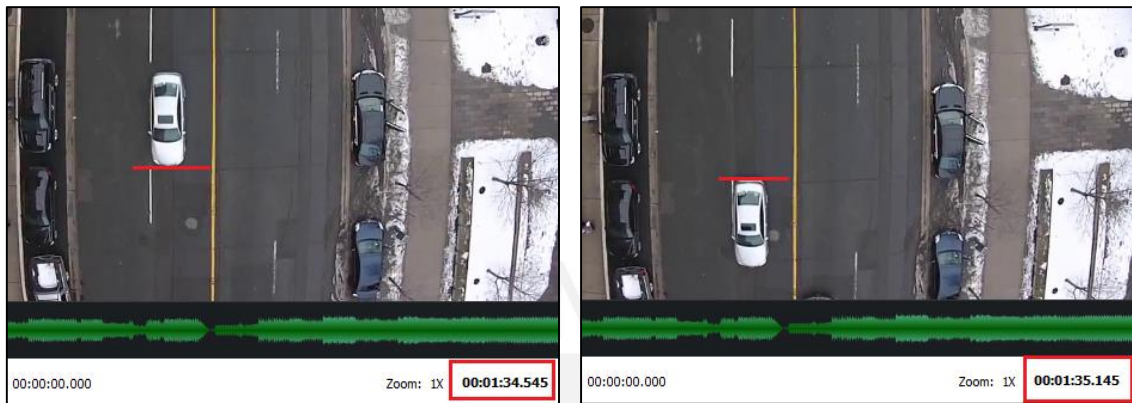
Velocidad de autos: $v = d/t$ Ecuación 4.5

v : Velocidad del auto

d : Distancia de recorrido del auto

t : Tiempo que demora el auto en recorrer la distancia d

Para hallar la velocidad de los autos se usó el banco de videos con los que se cuenta; y mediante el **software Free Video Editor v.1.4.15**, podemos estimar la distancia que se desplazó y hallar los tiempos que le toma al vehículo en recorrer esa distancia, mediante el manejo de las opciones que nos brinda el software. La distancia a recorrer será el largo del vehículo. Ver la Figura 4.18.



(a)



(b)

Figura 4.18: Desplazamiento de vehículos con diferentes velocidades.

En la Figura 4.18, se muestra el desplazamiento de dos vehículos con distintas velocidades, y esto se demuestra observando el lapso de tiempo que les toma recorrer a cada uno la distancia que corresponde a la longitud del vehículo (se acordó una longitud de 4 metros). Se calculará la velocidad del auto más lento (Figura 4.18a).

$$v_a = \frac{d}{t} = \frac{4m}{(95.128 - 94.545)seg} * \frac{1 km}{1000m} * \frac{3600seg}{1 Hr} = 24.69 \frac{Km}{Hr} = 6.85 \frac{m}{s}$$

Ahora se procederá a calcular la velocidad de otro auto (Figura 4.18b).

$$v_b = \frac{d}{t} = \frac{4 m}{(40.741 - 40.424)seg} * \frac{1 km}{1000m} * \frac{3600seg}{1 Hr} = 45.42 \frac{Km}{Hr} = 12.61 \frac{m}{s}$$

La distancia que recorrió el auto en la Figura 4.18b se dio en un lapso de tiempo el cual está definido en 10 fotogramas, esta información se adquirió mediante el software **Free Video Editor**. Y teniendo la relación entre pixeles y metros, podemos hallar la velocidad en función del número de pixeles.

$$\text{Auto con mayor velocidad: } v_b = 12.61 \frac{m}{s} * 16.25 \frac{pixel}{m} = 204.91 \frac{pixel}{s}$$

$$\text{Auto con menor velocidad: } v_a = 6.85 \frac{m}{s} * 16.25 \frac{pixel}{m} = 111.31 \frac{pixel}{s}$$

Ahora para hallar la distancia que recorre el auto entre un fotograma y otro, bastará con dividir la distancia recorrida entre el número de fotogramas.

$$d_{f_t \rightarrow f_{t+1}} = \frac{65 \text{ pixeles}}{10 \text{ fotogramas}} = 6.5$$

Esto nos indica que la representación de un objeto que se encuentra en movimiento a lo largo de una secuencia de video, estará en una ubicación de fotograma previo y que en el siguiente fotograma, el mismo objeto se encontrará a 6.5 pixeles desplazados.



Figura 4.19: desplazamiento de auto a lo largo de una distancia igual a su longitud.

4.3.3 Traslape por desplazamiento

En la Figura 4.19, el desplazamiento de un auto no lleva mayor análisis en cuanto al recorrido que posee, pero que sucedería si existiesen dos autos cercanos entre si y en movimiento. Como se observa en la Figura 4.20.

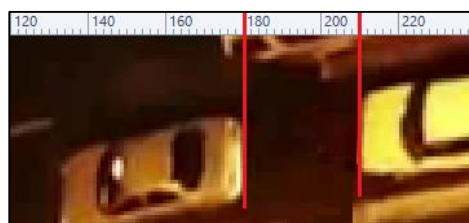


Figura 4.20: Representación de autos en movimiento.

En el video se aprecia la escena del desplazamiento de dos autos contiguos, y se estimó una distancia de separación de los mismos, dicha distancia es de 30 píxeles equivalente a 1.84m. Según lo anteriormente explicado, esto afectaría al análisis de desplazamiento de ambos autos, ya que ocurriría lo que se muestra en la Figura 4.21.

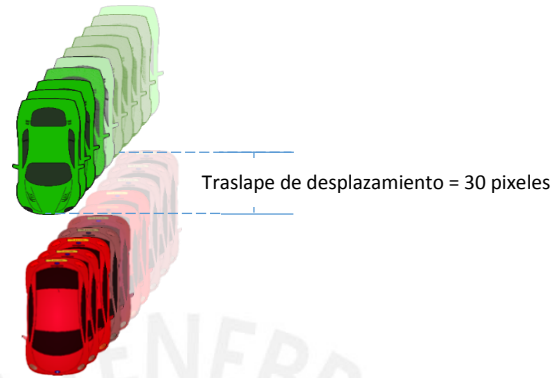


Figura 4.21: Representación de autos en movimiento.

En esta figura se aprecia el traslape de ambos desplazamientos. Para solucionar este problema se decidió realizar el análisis de detección de objetos en movimiento cada 5 fotogramas; es decir, la separación entre los fotogramas a procesar será de 5 fotograma (separación en tiempo de 16.7 msec), el cual denominaremos tasa de muestreo (Tasa de muestreo definido por nosotros), evitando el traslape de desplazamiento.

$$fotograma_{(t)} \rightarrow fotograma_{(t+5)}$$

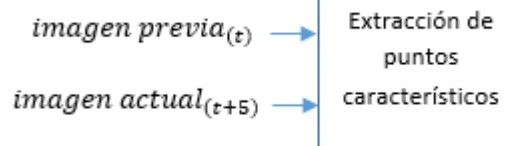


Figura 4.22: etapa de extracción de puntos característicos.

En la Figura 4.23 se muestra el real efecto que produce el traslape por desplazamiento de dos autos que se encuentran cerca, los cuales se denotan en el recuadro de color rojo.



Figura 4.23: Efecto del traslape de desplazamiento.

Entonces, según lo explicado en la Figura 4.22 respecto a la separación entre los fotogramas que se procesan, se muestra en la Figura 4.24 cómo se selecciona los fotogramas y sus respectivos fotogramas resultantes, provenientes de la resta entre el fotograma actual y la transformación del fotograma previo, usando la matriz de homografía.

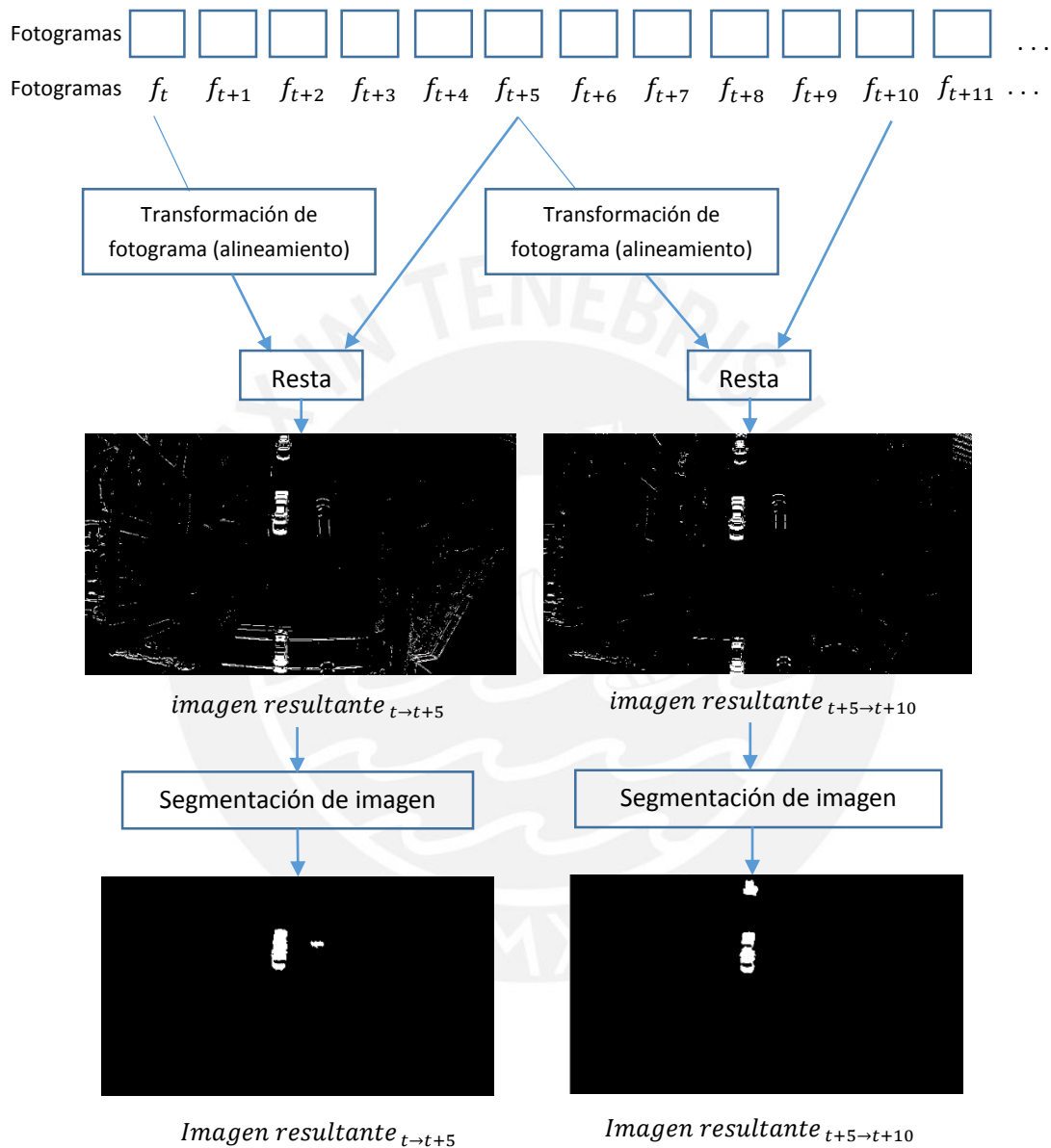


Figura 4.24: Método de selección de los fotogramas a procesar.

4.4 Segmentación de imagen

Luego de haber hallado la diferencia entre el fotograma actual y la transformación en perspectiva del fotograma previo; y teniendo en cuenta las consideraciones hechas en el capítulo 4.3, que indica la separación entre fotogramas a procesar (cada 5 fotogramas), entonces ahora se procederá con la extracción de los objetos de interés, en nuestro caso personas y vehículos.

Esta etapa se realizará mediante el software MATLAB R2015a, y se hará uso de las funciones que brinda para el procesamiento de imágenes digitales, tales como umbralización de imagen, dilatación, erosión, etc. Esto se realiza con el fin obtener los objetos detectados en la imagen, a dichos objetos se les denomina como “Blobs” en el procesamiento de imagen. A continuación se muestra en la figura 4.25 el proceso que se sigue para obtener los blobs.

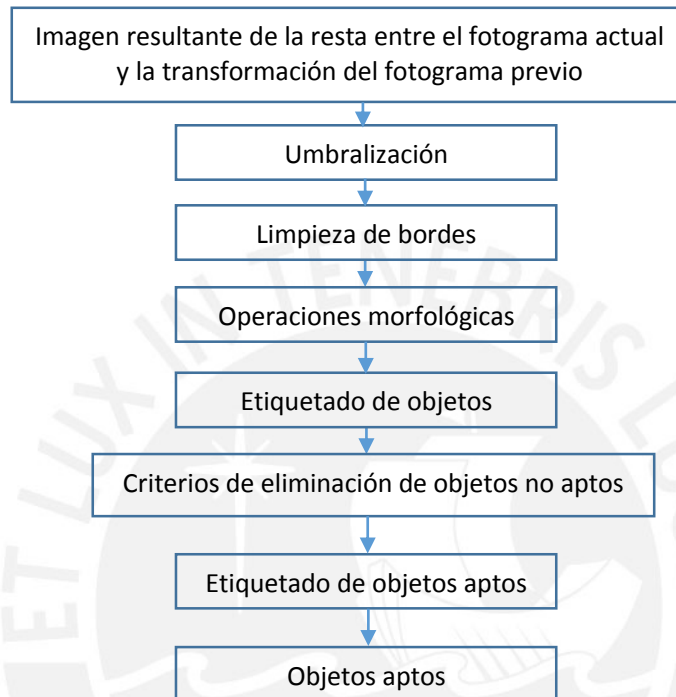


Figura 4.25: Proceso de segmentación de imagen para la extracción de objetos de interés.

El criterio de eliminación de objetos no aptos, se basa en las características que poseen los objetos a detectar, en nuestro caso personas y vehículos; por lo que tomaremos como sus características principales, la relación entre el área en píxeles y el área del contorno que delimita al objeto; y la proporción existente entre el número de píxeles de largo y ancho que ocupan en la imagen. A continuación se muestra un ejemplo de los blobs que se encontraron luego de aplicar la etapa de segmentación de imagen.

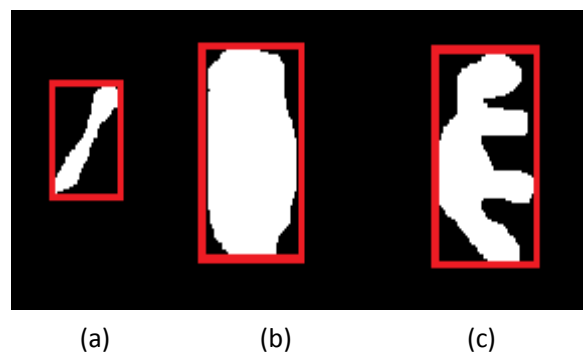


Figura 4.26: Objetos detectados luego de la etapa de segmentación.

En la Figura 4.26, se muestra algunos de los posibles objetos que se obtendrán luego de la etapa de segmentación de imagen, es por eso que se tienen criterios de eliminación, el primero, relación área entre el objeto y área del contorno, y el segundo criterio es la relación entre las dimensiones propias del objeto.

En la figura 4.27, se explica tomando como ejemplo a un vehículo que se desplaza, para lo cual se considera las observaciones hechas en los puntos 4.3.1 y 4.3.3. En esta figura se aprecia el desplazamiento de un vehículo y su representación en pixeles luego de la etapa procesamiento de imagen, con lo que se obtiene lo mostrado en 4.27b.

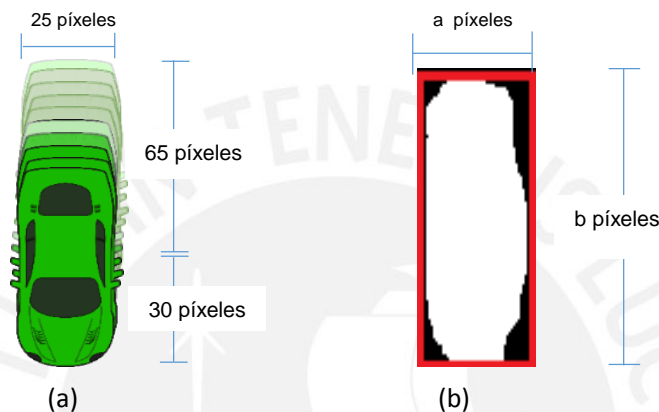


Figura 4.27: Representación de un vehículo en desplazamiento y su respectivo blob.

Como se mencionó, los criterios de eliminación se encargan de descartar todos aquellos objetos detectados que no representan a un vehículo o persona.

En el caso de vehículos, el primer criterio es la relación de áreas, la cual está limitada por un valor estimado de 0.7, lo que indica que los blobs que posean una relación de área mayor a este valor poseen la probabilidad de considerarse como vehículos detectados. El segundo criterio es la relación que existe entre las dimensiones del blob detectado, considerando los valores mostrados en la figura 4.27. Si la relación de largo y ancho es mayor a lo explicado en la ecuación 4.7, se considera como un objeto que podría representar a un vehículo; es decir, si y solo si ambos criterios se cumplen, ecuación 4.6 y 4.7, entonces se considera al blob detectado como un vehículo u objeto apto.

$$relación_{\text{areas}} = \frac{\text{area de pixeles blancos}}{\text{area del contorno}} = \frac{\text{area}_{\text{Blanco}}}{a*b} > 0.7 \quad \text{Ecuación 4.6}$$

$$relación_{\text{largo-ancho}} = \frac{b}{a} > 3.8 \quad \text{Ecuación 4.7}$$

Una vez obtenidos los objetos aptos, se extraerán las coordenadas del centro de cada uno de estos y los almacenaremos en un arreglo de matrices. Estas coordenadas serán de utilidad en el seguimiento de cada blob u objeto apto.

4.5 Seguimiento de objetos de interés

Luego de obtener los blobs, personas y vehículos; se realizará el seguimiento de estos, el cual concluirá en la detección de los objetos de interés (vehículos y personas) a lo largo de toda la secuencia de fotogramas.

Para esta tesis, el objetivo del seguimiento de los blobs es determinar si existe algún movimiento de un objeto (vehículo y personas) en la secuencia de video. Para lograr el seguimiento se optó por considerar la ubicación del objeto detectado en las imágenes resultantes de la etapa de segmentación, por lo cual se planteó el método de ubicación del centroide, y la distancia a la que se encuentra respecto a los centroides de los demás blobs (objetos detectados).

4.5.1 Distancia entre centroides

La distancia entre centroides de los blobs de un par de imágenes resultantes, nos permite saber que blobs se mantienen en la secuencia de fotogramas o si son objetos nuevos que aparecen; es decir, calculando los centroides de todos los blobs correspondientes al fotograma previo y actual; y hallando su distancia entre ellos, podemos establecer que coordenadas del centro pertenecen a un mismo blob. Para establecer esta distancia se tuvo en cuenta el punto medio (centroide) del objeto a detectar, en este caso un vehículo, ya que los centroides de los vehículos que se desplazan uno al lado de otro poseen una separación de 25 píxeles (1.53 metros), por lo que se consideró tomar un valor 30 píxeles (1.83 metros). Como se muestra en la figura 4.222222

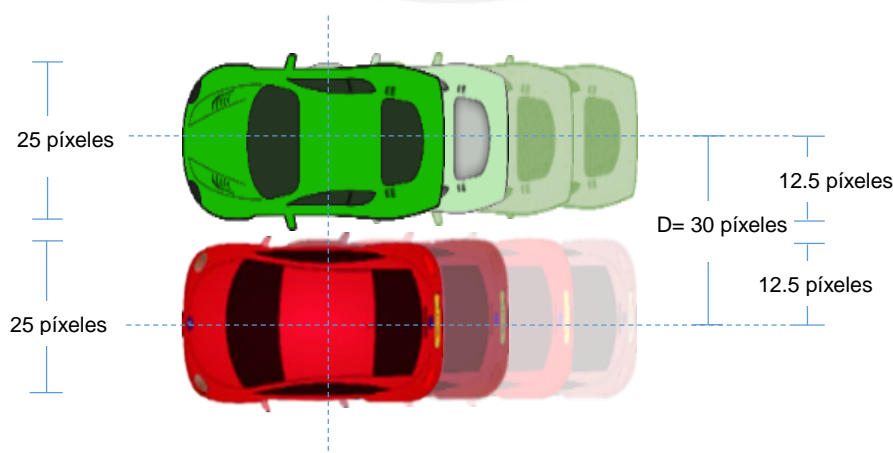


Figura 4.28: Representación de la distancia estimada entre centroides.

A continuación se muestra el proceso de cálculo de distancias entre los centroides de la secuencia de fotogramas

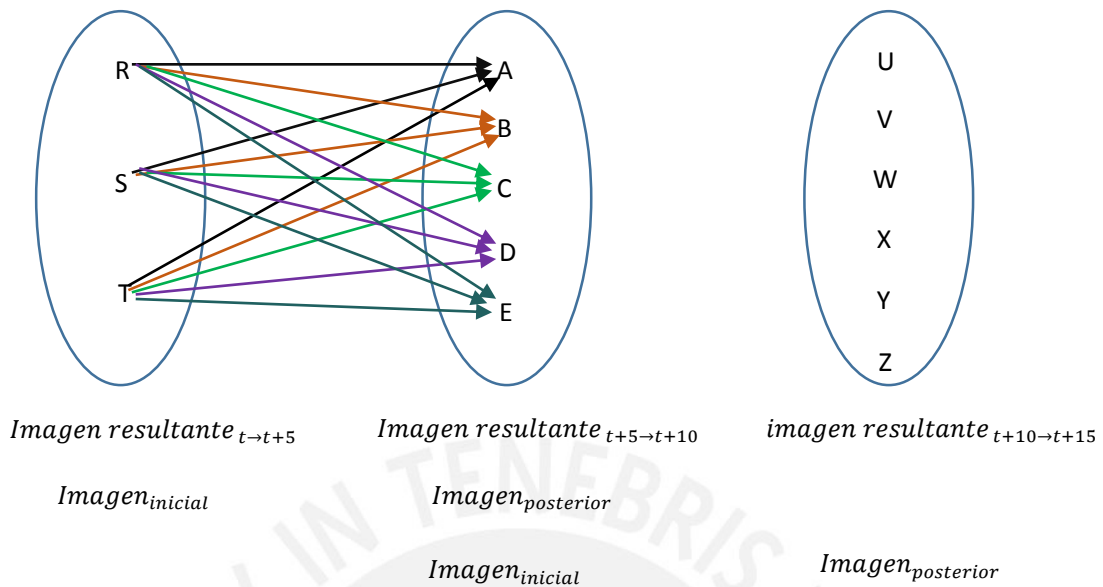


Figura 4.29: Metodología propuesta para el seguimiento de blobs u objetos aptos detectados.

En la Figura 4.29, se muestra el funcionamiento de la etapa de seguimiento de blobs, la cual nos indica los blobs detectados luego de la etapa de segmentación de imagen.

Entonces una vez obtenidos estas imágenes resultantes, se trabajará en pares, donde cada par estará definido como imagen inicial y posterior. Para la explicación de esta etapa se tomará como ejemplo el primer par, conformado por la *Imagen resultante t->t+5* e *Imagen resultante t+5->t+10*, y el resto del proceso se replica para los siguientes pares; es decir, *Imagen resultante t+5->t+10* e *Imagen resultante t+10->t+15*.

En primer lugar se hallan los centroides de los blobs de la imagen inicial, los cuales se han etiquetado en este caso como blob R, blob S y blob T; y también se hallan los centroides de los blobs de la imagen posterior, blob A, blob B, blob C, blob D y blob E.

Luego se calculan las distancias entre los centroides de los blobs de la imagen inicial y posterior, con el fin de verificar si algún blob de la imagen inicial está representado en la imagen posterior. Para lo cual se usará la relación de cálculo de distancia entre 2 puntos, ver ecuación 4.8.

$$d = ((x_{imagen\ inicial\ blob\ i} - x_{imagen\ posterior\ blob\ j})^2 + (y_{imagen\ inicial\ blob\ i} - y_{imagen\ posterior\ blob\ j})^2)^{1/2}$$

Ecuación 4.8

Para entender este concepto, se realizará la siguiente explicación: se hallaron las distancias entre los centroides de los blobs (R, S y T) de la imagen inicial y el centroide del blob A de la imagen posterior; y se calculó que todas las distancias entre ellas eran mayores a 30 píxeles, ver Figura 4.30b, entonces esto indica que el blob A, no pertenece a la imagen inicial sino que representa a un blob nuevo. Pero en caso ocurriese que al menos una de las distancias sea menor a 30, por ejemplo, la hallada entre el blob T y el blob A, ver Figura 4.30a, nos indicaría que el blob A representa al mismo objeto (vehículo o persona) que representa el blob T, entonces se almacenará el centroide del blob T.

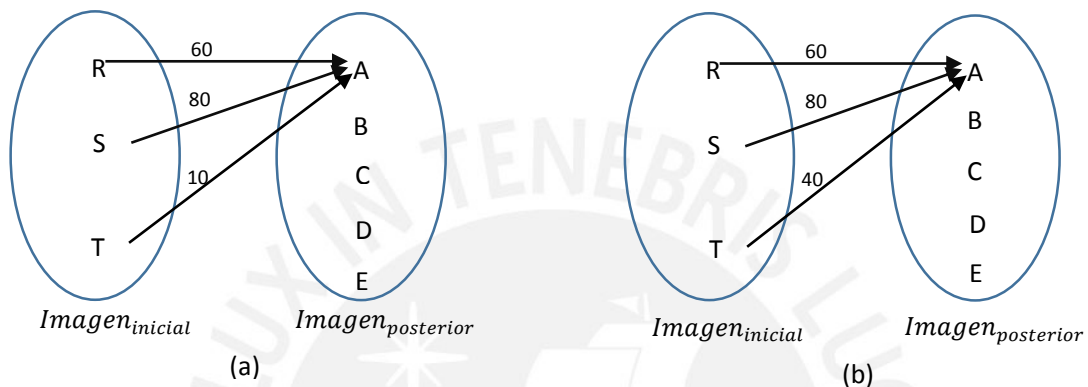


Figura 4.30: Distancias entre los blobs de la imagen inicial y posterior.

La Figura 4.31 muestra lo anteriormente mencionado, en base a tres imágenes obtenidas por medio del proceso de segmentación de imagen, para lo cual se usó el ejemplo de la figura 4.24.

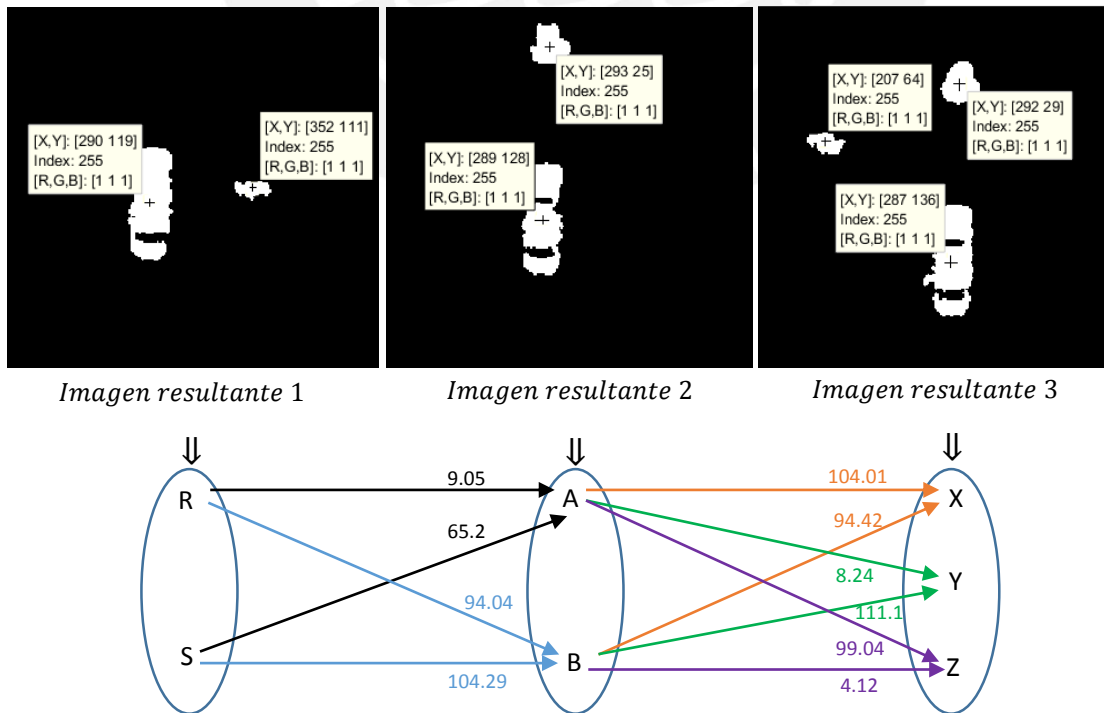


Figura 4.31: Representación de los centroides de los blobs en las tres imágenes resultantes.

Una vez hallados los centroides, se procede a calcular las distancias entre los blobs de cada par de imágenes resultantes mediante la ecuación 4.8. Primero se hallan las distancias entre los centroides de los blobs pertenecientes a la imagen resultante 1 (blob R y blob S) e imagen resultante 2 (blob A y blob B); las cuales se muestran en la figura 4.31. Y una vez obtenidas esas distancias, se empiezan a comparar una por una con la distancia definida de 30 píxeles (1.84 metros), dependiendo si es mayor o menor se presentará alguno de los dos casos posibles. El primero caso es si encontramos que la distancia es menor a 30 píxeles (1.84 metros), esto indicaría que un blob perteneciente a la imagen inicial también se encuentra en la imagen posterior, y almacenaríamos sus coordenadas, como ocurre en nuestro ejemplo con el blob R y blob A, con lo cual definimos que ambos blobs representan al mismo objeto detectado y denominaremos blob 1. Si se diera el caso de que existiese más de una distancia menor a 30 píxeles se escoge al menor. El segundo caso posible es cuando la distancia sea mayor a 30 píxeles; esto indica que un nuevo blob ha aparecido en la imagen posterior, el cual no se encontraba en la imagen inicial; este se almacena, como pasa con el blob B.

Ahora se hallan las distancias entre los centroides de los blobs de la imagen resultante 2 (blob A y blob B) e imagen resultante 3 (blob X, blob Y y blob Z), y se procede de la misma manera. Entre estos blobs se concluyó que el blob A y Blob Y representan al mismo objeto detectado, lo mismo sucede con el blob B y blob Z.

Cuando se analiza todas las distancias se puede dar el caso que algunos blobs de la imagen inicial no encuentren sus respectivos blobs en la imagen posterior; es ahí cuando se dice que el blob ya no está presente en la secuencia de fotogramas, lo que ocurre con el blob S. En la tabla 4.3 se muestra el almacenamiento de centroides de los blobs, para el ejemplo anterior.

Tabla 4.3: Almacenamiento coordenadas de 4 blobs detectados en las primeras 3 imágenes resultantes (16 fotogramas).

	Blob 1 (R,A,Y)		Blob 2 (S)		Blob 3 (B,Z)		Blob 4 (X)	
	Coordenadas del centroide		Coordenadas del centroide		Coordenadas del centroide		Coordenadas del centroide	
	X	y	x	y	x	y	x	y
imagen resultante 1	119.58	290.88	119.36	352.34	0	0	0	0
imagen resultante 2	128.30	289.48	0	0	24.06	293.73	0	0
imagen resultante 3	136.49	287.39	0	0	29.24	292.53	64.98	207.21

4.6 Pruebas y resultados

Se procedió a realizar las pruebas con las distintas secuencias de video de nuestro banco de datos (ver figura 4.2), los fotogramas fueron obtenidos a partir de videos grabados desde un UAV, con un ángulo de depresión de 90°. En estos videos se da la presencia de personas y vehículos, tanto estáticos como en movimiento, estos últimos serán detectados por nuestro sistema y se calculará la eficiencia de este, en función de los resultados.

El sistema planteado cuenta con parámetros, los cuales se deben tener en cuenta para un mejor desempeño. Los parámetros del sistema son:

El número de separación entre fotogramas o tasa de muestreo, como lo denominaremos a partir de ahora (ver punto 4.3). Este parámetro estará definido por las dimensiones del objeto a detectar y a su velocidad de desplazamiento.

La distancia máxima entre los centroides de los objetos, este parámetro es importante, ya que separa un objeto detectado de otro, y así evitamos la asignación incorrecta de coordenadas cercanas de los objetos detectados, ver punto 4.5.1.

El número de fotogramas a procesar, este parámetro hace referencia al número total de fotogramas que se obtiene del video, ya que en esta tesis se realiza el etiquetado de objetos, por lo que es necesario almacenar todas las ubicaciones de un mismo objeto a lo largo de toda la secuencia de video, y si se desea reducir el tiempo de procesamiento, es importante estimar cuantas imágenes resultantes se obtendrán luego de la etapa de segmentación de imagen, con el fin de separar un espacio de memoria, el cual luego será ocupado por todas las posiciones encontradas.

4.6.1 Resultados de la primera prueba.

A continuación se realizará las pruebas del sistema, aplicados al video *Toronto_58s.avi*, y que se aprecia en la Figura 4.2e. Este video se caracteriza haber sido grabado con UAV que se desplaza con una velocidad de 6 km/hr aproximadamente y por mostrar escenas con vehículos y personas en movimiento, así como autos estacionados, de los cuales nuestro sistema detectará solo los objetos en movimiento. Estos parámetros se muestran en la tabla 4.4.

Tabla 4.4: Parámetros considerados en el procesamiento del video torono_58s.avi.

Parámetros	valores
Tasa de muestreo	5
Distancia entre centroides de objetos	1.83 metros (30 pixeles)
Número de fotogramas	500

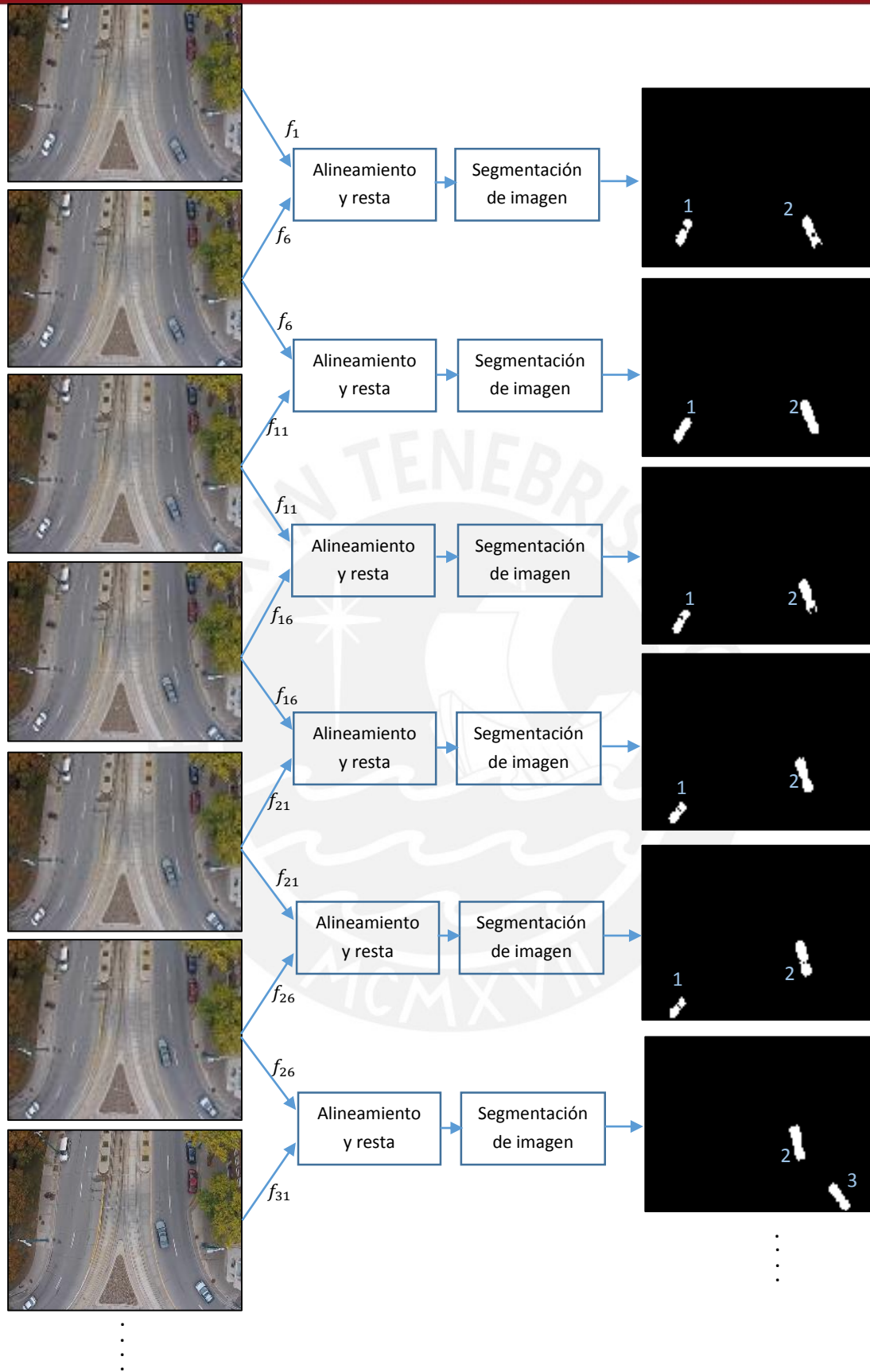


Figura 4.32: Primera prueba, donde se muestra los 7 primeros fotogramas tomados de 5 en 5 y la obtención de las 6 primeras imágenes, luego de la segmentación de imagen.

En la figura 4.32 se muestra las 6 primeras imágenes resultantes del total que se han obtenido luego de la segmentación de imagen, y se aprecia cómo hasta ese momento solo se ha dado la presencia de 3 objetos (blobs), los cuales han sido ubicados, mediante su centroide y almacenados en un arreglo de matrices.

Tabla 4.5: Almacenamiento coordenadas de 4 blobs detectados en las primeras 6 imágenes.

	Blob 1		Blob 2		Blob 3		Blob 4	
	Coordenadas del centroide		Coordenadas del centroide		Coordenadas del centroide		Coordenadas del centroide	
	x	y	x	y	x	y	x	y
imagen 1	291.39	138.47	290.38	415.79	0	0	0	0
imagen 2	304.12	130.27	278.29	407.73	0	0	0	0
imagen 3	315.52	122.61	262.72	398.68	0	0	0	0
imagen 4	325.88	115.18	248.44	391.39	0	0	0	0
imagen 5	335.52	107.61	233.11	384.97	0	0	0	0
imagen 6	0	0	217.50	378.71	329.27	470.64	0	0

En la Figura 4.33 se muestra el seguimiento de uno de los blobs detectados, blob 2, mediante los puntos almacenados a lo largo de la secuencia de imágenes.

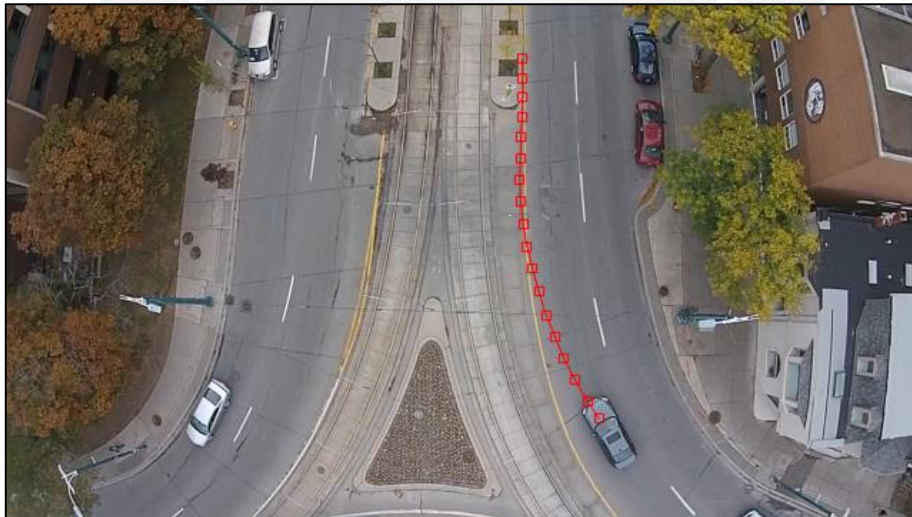


Figura 4.33: Trayectoria del vehículo detectado.

En la Figura 4.34, se muestra las coordenadas del centro del blob 2, obtenidas en cada imagen, representadas en una gráfica 3D. Dicho blob está presente desde la imagen resultante 1 hasta la imagen resultante 18, lo que equivale a 91 fotogramas.

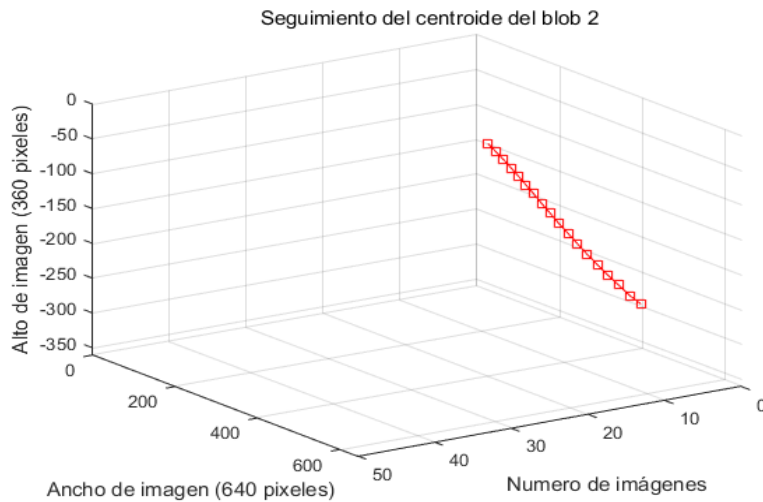


Figura 4.34: Seguimiento del blob 2 a lo largo de la secuencia de imágenes.

En esta primera prueba, realizada a un video que posee personas y autos en movimiento, además de autos estáticos, se detectó un número mayor de vehículos y ninguna persona, con respecto al número por conteo manual, ver Tabla 4.2. En el caso de vehículos se produjo la aparición de falsas detecciones, fue debido a la equivocada alineación de fotogramas en ciertos instantes, ya sea por giros bruscos del UAV o acercamiento y alejamiento del enfoque de la cámara. En el caso de las personas, no se detectó ninguna, debido a las dimensiones de las mismas y al parámetro de separación de fotogramas a procesar, puesto que se estimó este parámetro en relación a la velocidad de los vehículos, el cual es mayor a la de una persona y no permitió apreciar el desplazamiento de la persona.

Tabla 4.6: Número de objetos detectados como correctos en la primera prueba.

Número de vehículos (conteo manual)	Total de objetos detectados	Objetos detectados como correctos (criterios de eliminación)	% de error
11	70	13	$\%error = \frac{13 - 11}{11} * 100\% = 18.18\%$

En la tabla 4.7 se muestra el total de objetos que se detectaron ya sean correctos o no, y nos indica en qué zona de la imagen se encuentran. Hay un mayor número de blobs en la zona de edificaciones, debido a que estos elementos poseen contornos definidos por rectas y que en la etapa de alineación no fueron eliminados por resta de imágenes.

Tabla 4.7: Número total de blobs detectados en la primera prueba.

	Zona de carretera	Zonas de edificaciones y árboles (a los lados de la carretera)
Numero de blobs	28	42

4.6.2 Resultados de la segunda prueba del sistema

En esta segunda prueba, se realizó el mismo proceso seguido en el ensayo anterior (Figura 4.32), haciendo uso del video *soccer1.wmv*, y que se aprecia en la Figura 4.2b. Este video se caracteriza por haber sido grabado con un UAV que se desplaza con una velocidad de 6 km/hr y por contener solo personas en movimiento, por lo que los parámetros serán distintos debido a las dimensiones y velocidad de desplazamiento de las personas. Se muestran los parámetros usados para este video en la tabla 4.8.

Tabla 4.8: Parámetros considerados en el procesamiento del video soccer1.wmv.

Parámetros	valores
Tasa de muestreo	4
Distancia entre centroides de objetos	1 metro (16.25 pixeles)
Número de fotogramas	200

A continuación se muestra en la tabla 4.9 el almacenamiento de los primeros 4 blobs detectados en las 6 primeras imágenes, luego de la segmentación de imagen.

Tabla 4.9: Almacenamiento de las coordenadas de 4 blobs detectados en las primeras 6 imágenes resultantes.

	Blob 1		Blob 2		Blob 3		Blob 4	
	Coordenadas del centroide		Coordenadas del centroide		Coordenadas del centroide		Coordenadas del centroide	
	X	y	x	y	x	y	x	y
imagen 1	238.78	76.75	183.92	121.92	155.48	129.03	57.46	179.36
imagen 2	243.15	79.75	180.29	118.27	159.46	133.48	59.33	178.03
imagen 3	244.54	78.13	174.04	116.27	162.43	137.81	58.85	176.17
imagen 4	244.92	80.07	164.75	116	167	146.91	57.15	174.88
imagen 5	243.96	80.86	157.94	114.83	172.29	157.84	53.81	173.60
imagen 6	242.85	83.47	150.28	113.76	174.45	165.81	55.45	174.48

El seguimiento de los blobs detectados se muestra en la Figura 4.35, y la ubicación de los centroides de los blobs 2 y 8, obtenidas a lo largo de todas las imágenes resultantes.



Figura 4.35: Trayectoria de los blob 2 y 8.

En esta segunda prueba, realizada a un video que solo posee personas en movimiento, se pudo detectar un número mayor de personas, ver tabla 4.10, con respecto al número por conteo manual, esto debido a dos posibles factores, el primero se relaciona con el hecho de que las dimensiones del objeto a detectar (personas) eran mucho menores en relación a las dimensiones del fotograma, por lo que en algunas imágenes, luego de la segmentación de imagen, el objeto desaparecía en una imagen y volvía a aparecer en otra imagen, generando así la creación de un nuevo blob detectado, cuando en realidad se refiere al mismo objeto; y el segundo factor se basa en la aparición de falsas detecciones; es decir, objetos que no representan una persona, pero por dimensiones, el sistema lo consideró como tal, luego de la etapa de segmentación de imagen.

Tabla 4.10: Número de objetos detectados en la segunda prueba.

Número de personas (Conteo manual)	Total de objetos detectados	Objetos detectados como correctos (criterios de eliminación)	% de error
15	20	16	$\%error = \frac{16 - 15}{15} * 100\% = 6.66\%$

4.6.3 Resultados de la tercera prueba del sistema

En esta tercera prueba, se procedió como se hizo en las pruebas anteriores. Pero en este caso, haremos uso del video Toronto_35s.mp4. Este video se caracteriza por contener solo vehículos en movimiento y que el UAV se encuentra estático; es decir, una velocidad de 0 km/hr. Se muestran los parámetros usados en la Tabla 4.11.

Tabla 4.11: Parámetros considerados para procesar el video toronto_75s.avi.

Parámetros	valores
Tasa de muestreo	5
Distancia entre centroides de objetos	1.83 metros (30 pixeles)
Número de fotogramas	1062

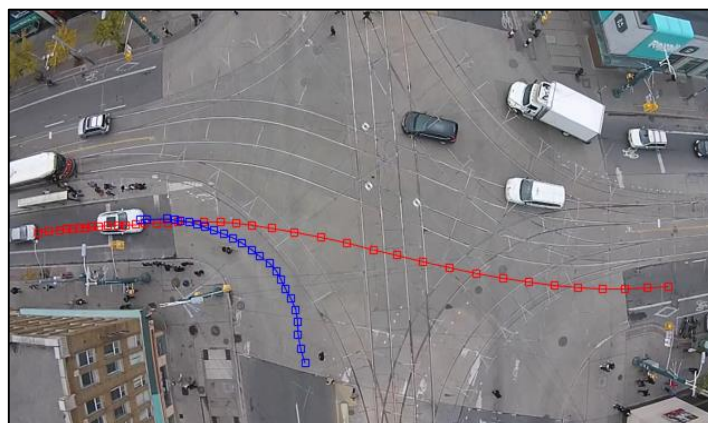


Figura 4.36: Trayectorias de dos blobs detectados.

Tabla 4.12: Número de objetos detectados en la tercera prueba.

Número de vehículos (Conteo manual)	Total de objetos detectados	Objetos detectados como correctos	% de error
26	122	29	$\%error = \frac{29 - 26}{26} * 100\% = 11.53\%$

4.6.4 Resultados de la cuarta prueba del sistema

En esta cuarta prueba se usó el video *via.mp4*, y se realizó el mismo proceso mostrado en la Figura 4.32. Este video se caracteriza por contener autos en movimiento y poseer escenas de abundante vegetación. Además se debe mencionar que en este video el UAV se desplaza con una velocidad de 40 km/hr, afectando los objetos que se detectan luego de la etapa de alineamiento, tal como se muestra en la figura 4.37.



Figura 4.37: Efecto de la selección adecuada de una tasa de muestreo.

En la figura 4.37 se muestra el alineamiento de 2 fotogramas con diferentes tasas de muestreo. La imagen obtenida del alineamiento con una tasa de muestreo de 5, el cual toma el $fotograma_t$ y $fotograma_{t+5}$, produce una duplicidad de un mismo objeto como se aprecia dentro del círculo rojo. En cambio cuando se usa una tasa de muestreo igual a 3, un vehículo es representado como un único objeto, como se ve en el círculo verde.

A continuación se muestran los parámetros usados para este video en la tabla 4.13.

Tabla 4.13: Parámetros considerados para procesar el video via.mp4.

Parámetros	valores
Tasa de muestreo	3
Distancia entre centroides de objetos	1.84 metros (30 pixeles)
Número de fotogramas	400
Datos del UAV	
Velocidad del UAV	40 km/hr

Una vez realizada la prueba se procede a ubicar los centroides de todos los blobs detectados como se muestra en la figura 4.38, en la cual se representa la trayectoria del blob 73 (vehículo). Así mismo se dibuja las coordenadas del blob 74, que corresponden a la trayectoria seguida por un blob que no es un vehículo, pero aun así es detectado. Esto se debe a la presencia de edificaciones y árboles en los videos, ya que cuando se realiza la etapa de alineamiento entre la transformación del fotograma previo y el fotograma actual y luego su posterior resta, aún quedan remanentes de la presencia de estos elementos, los cuales son considerados como objetos detectados.



Figura 4.38: Trayectorias seguida por los blob 73 y 74, que representa a un vehículo y un objeto no correcto, respectivamente.

En la tabla 4.14 se muestran las coordenadas de los blobs 73 y 74 detectados a lo largo de las imágenes resultantes.

Tabla 4.14: Almacenamiento de las coordenadas de los blobs 73 y 74.

	Blob 73		Blob 74	
	Coordenadas del centroide		Coordenadas del centroide	
	x	y	x	Y
imagen 76	29.35	371.58	0	0
imagen 77	45.53	371.66	137.52	215.81
imagen 78	63.63	371.71	150.44	215.27
imagen 79	76.31	372.89	157.71	231.32
imagen 80	101.11	371.15	167.56	245.39
imagen 81	116.64	371.47	175.12	230.12
imagen 82	142.95	370.93	189.45	224.60
imagen 83	159.82	370.71	201.25	217.43
imagen 84	183.55	371.30	220.42	243.21
imagen 85	207.12	370.99	0	0

A continuación se muestra en la tabla 4.15 todos los blobs detectados, los cuales fueron separados por su ubicación en zonas determinadas de un fotograma.

Tabla 4.15: Número de objetos totales detectados en la cuarta prueba.

	Zona de carretera	Zonas de edificaciones y árboles
Numero de blobs	16	49

Finalmente se muestra en la tabla 4.16 el número de objetos detectados como correctos, eliminando los objetos que no corresponden a un vehículo, mediante los criterios de eliminación.

Tabla 4.16: Numero de objetos detectados como correctos en la cuarta prueba.

Número de vehículos (Conteo manual)	Total de objetos detectados	Objetos detectados como correctos	% de error
6	65	7	$\%error = \frac{7-6}{6} * 100\% = 16.6\%$

4.6.5 Resultados de la quinta prueba del sistema

Para esta última prueba se usó el video *Toronto_75s.avi*. Este video se caracteriza por contener escenas con edificaciones y árboles, autos estacionados, autos en movimiento y haber sido grabado con un UAV que posee una velocidad de desplazamiento de 10 km/hr. En la tabla 4.17 se muestra los parámetros a usar en esta prueba.

Tabla 4.17: Parámetros considerados para la prueba con el video toronto_75s.avi

Parámetros	valores
Tasa de muestreo	5
Distancia entre centroides de objetos	1.84 metros (30 pixeles)
Número de fotogramas	870
Datos del UAV	
Velocidad del UAV	10 km/hr

En la figura 4.39 se muestra la trayectoria seguida por dos blobs detectados que corresponden a vehículos.

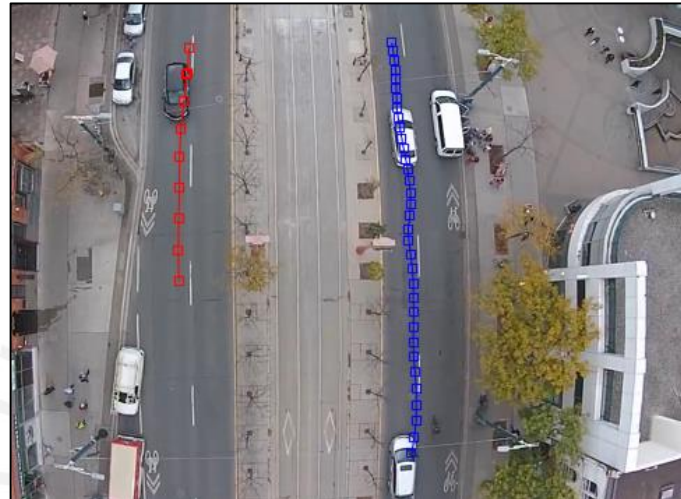


Figura 4.39: Trayectoria de los blobs 104 y 102

Al igual que en la prueba anterior se identifican todos los blobs detectados y se separan según las zonas del fotograma, las cuales son zona de carretera y zona de edificaciones. El número de blobs detectados en cada zona se muestra en la tabla 4.19.

Tabla 4.18: Número total objetos detectados en la quinta prueba.

	Zona de carretera	Zonas de edificaciones y árboles
Numero de blobs	23	87

Luego de hallar todos los blobs, se prosigue a usar los criterios de eliminación con el fin de la obtener solo los objetos detectados que correspondan a vehículos. Ver tabla 4.20

Tabla 4.19: Número de objetos detectados como correctos en la quinta prueba.

Número de vehículos (Conteo manual)	Total de objetos detectados	Objetos detectados como correctos	% de error
16	110	19	$\%error = \frac{19 - 16}{16} * 100\% = 18.75\%$

4.6.6 Análisis de resultados

Una vez realizadas las pruebas se procede a analizar los resultados obtenidos, los cuales se muestran en la tabla 4.21.

Tabla 4.20: Cuadro resumen de los resultados obtenidos en las pruebas.

	Velocidad de UAV	Numero de objetos contados manualmente	Numero de objetos detectados como correctos	% de error
Primera prueba	6 km/hr	11	13	18.18%
Segunda prueba	6 km/hr	15	16	6.66%
Tercera prueba	0 km/hr	26	29	11.53%
Cuarta prueba	40 km/hr	6	7	16.6%
Quinta prueba	10 km/hr	16	19	18.75%

Para todas las pruebas realizadas se debe recordar que la velocidad de desplazamiento de los vehículos se estimó en máximo de 45.42 km/hr, permitiendo establecer una tasa de muestreo igual a 5 como se mostró en el punto 4.3.2 y 4.3.3.

Como se observa en la tabla 4.21 existen factores que afectan el desempeño del sistema propuesto; el primero es la velocidad de desplazamiento del UAV, el cual se observó en la cuarta prueba, ya que esto produciría que se detecte el doble de objetos como se explica en la figura 4.37. Para solucionar este problema se consideró usar una tasa menor de muestreo ($i=3$), reduciendo el número de objetos detectados. El segundo factor es la escena presente en la secuencia de video, la cual contiene elementos en movimiento y estáticos. Entre los elementos estáticos se encuentran las edificaciones y árboles, los cuales luego de la etapa de alineamiento y resta, generan blobs que no corresponden a un vehículo o persona en movimiento y estarán presente solo hasta antes de aplicar los criterios de eliminación (ecuaciones 4.6 y 4.7). En la tabla 4.21 se muestra el número de blobs (objetos) antes y después de los aplicar los criterios.

Tabla 4.21: Blobs detectados antes y después de la etapa de segmentación de imagen.

	Blobs totales detectados	Blobs detectados como correctos (uso de criterios de eliminación)	Porcentaje de blobs eliminados mediante los criterios
Primera prueba	70	13	$\frac{73 - 13}{70} * 100\% = 81.42\%$
Tercera prueba	122	29	$\frac{122 - 29}{122} * 100\% = 76.23\%$
Cuarta prueba	65	7	$\frac{65 - 7}{65} * 100\% = 89.23\%$
Quinta prueba	110	19	$\frac{110 - 19}{110} * 100\% = 82.72\%$

En la tabla 4.21 no se colocó los datos de la segunda prueba, ya que no presenta edificaciones en la escena de fondo.

Conclusiones

El sistema de detección propuesto en esta tesis, desarrolla su mayor eficiencia en escenarios tales como los presentados en los videos de la segunda y tercera prueba. Con un porcentaje de error de 6.66% y 11.53% respectivamente, el sistema demuestra capacidad para trabajar en áreas donde no haya objetos que dificulten el alineamiento entre fotogramas, como pueden ser edificaciones y árboles.

El sistema propuesto reduce el porcentaje de error en la detección de vehículos cuando se trabaja con una tasa de muestreo igual a 5 (se toman fotogramas separados en el tiempo cada 16.7 ms) siempre que los vehículos que se desplazan poseen velocidades menores a 45.42 km/hr y una velocidad de un UAV menor a 10 km/hr.

Se debe considerar la separación que existe entre los objetos que se quieren detectar, con el fin de evitar el efecto del traslape de desplazamiento. Para una separación entre vehículos, antes de iniciar su desplazamiento, considerada en 30 pixeles que equivale a 1.83 metros, se debe trabajar con una tasa de muestreo de 5, eliminando así el traslape.

El número de objetos totales detectados con el sistema, en escenarios con presencia de edificaciones y árboles, incluyen objetos que no corresponden a vehículos. Una vez aplicados los criterios de eliminación, el número de objetos detectados se reduce en más del 76 % como se vio en el resumen, lo cual mejora el desempeño del sistema.

Para velocidades de desplazamiento de un UAV mayores a 40 km/hr la tasa de muestreo establecido como 5, el sistema genera una duplicidad de los objetos en movimiento. Se establece trabajar con tasas de muestreo iguales a o menores a 3 para estos valores de velocidades.

Recomendaciones

En algunos casos el UAV tendrá una velocidad de desplazamiento mínimo, con lo cual podría considerar estático. Es en estos casos donde el cálculo de la matriz de homografía puede fallar, indicando que los fotogramas tomados no poseen una relación de transformación. Es por esa razón que se sugiere implementar un algoritmo que identifique estos casos, con el fin de seleccionar fotogramas en las que se puedan aplicar el alineamiento.

Se debe enfocar en buscar soluciones para disminuir el porcentaje de error, teniendo como referencia las observaciones brindados en esta tesis, como identificar los tipos de escenarios que se van a grabar.

Se deben realizar estudios de detección de objetos cuando ocurre un cambio drástico de la escena, puesto que estos cambios son propios de un UAV y pueden ser giros bruscos o alejamiento de la escena. Esto implica que el alineamiento entre los fotogramas no sea el adecuado.

Cuando se desee realizar la detección y seguimiento de personas a partir de imágenes tomadas desde un UAV. Se recomienda que en dicha secuencia de imágenes, las personas se encuentren a una distancia adecuada, evitando así que el sistema de detección considere como un objeto correcto a un grupo de personas.

Bibliografía

- [1] FUERZA ÁREA DEL PERÚ
Consulta: 10 de Setiembre del 2013
<http://www.fap.mil.pe/>
- [2] INTELLIGENCIA DYNAMICS
Consulta: 10 de setiembre del 2013
<http://www.iuavs.com/>
- [3] JIAN YAO; ZHONGFEI ZHANG
2005 *Semi-supervised learning based object detection in aerial imagery*. IEEE Computer Society Conference on Computer Vision and pattern recognition, 2005.CVPR
Digital Object Identifier: [10.1109/CVPR.2005.318](https://doi.org/10.1109/CVPR.2005.318)
- [4] SIRMAECK, B; UNSALAN, C.
2009 *Urban-Area and Building Detection using SIFT Keypoints and Graph Theory*. IEEE Transactions on Geoscience and remote sensing. 2009
Digital Object Identifier: [10.1109/TGRS.2008.2008440](https://doi.org/10.1109/TGRS.2008.2008440)
- [5] MONDRAGN, I.F; CAMPOY, P.; CORREA, J.F.; MEJIAS, L.
2007 *Visual Model Feature Tracking for UAV control*. IEEE International Symposium on Intelligent Signal processing, 2007
Digital Object Identifier: [10.1109/WISP.2007.4447629](https://doi.org/10.1109/WISP.2007.4447629)
- [6] FELZENSZWALB, P.F.; GIRSHICK, R.B.; MCALLESTER, D.;
2010 *Object Detection with Discriminatively Trained Part-Based models*. IEE Transactions on Pattern Analysis and machine Intelligence. 2010
Digital Object Identifier: [10.1109/TPAMI.2009.167](https://doi.org/10.1109/TPAMI.2009.167)
- [7] CLEMONS, Jason
Scale invariant feature transform by David Lowe.
- [8] LOWE, DAVID G.
1999 *Object recognition from Local Scale-Invariant Features*. The Proceedings of the Seventh IEE International Conference on computer Vision, 1999
- [9] ENEBRAL GONZALEZ, JAVIER
2009 *Detección y asociación automática de puntos característicos para diferentes aplicaciones*. Tesis para la obtención del título de Ingeniera de Telecomunicaciones. Universidad Politécnica de Cataluña. 2009
<http://upcommons.upc.edu/pfc/bitstream/2099.1/8052/1/memoria.pdf>

- [10] NICOLAS PINA, ANTONIO
 2010 *Clasificación y Búsqueda de Imágenes usando características visuales*. Proyecto de fin de carrera de Ingeniería Informática, Murcia. Universidad de Murcia. Facultad de Informática. Página 40
<http://dis.um.es/~ginesgm/files/inv/pfc-anp/pfc-anp.pdf>
- [11] HERIGERT, MARILINA; BOUCHET, NAHUEL; PIANETTI, IVÁN
 2010 *Reconocimiento de Imágenes mediante Scale Invariant Feature Transformation (SIFT)*. Universidad Tecnológica Nacional. Facultad Regional Concepción Del Uruguay
http://www.frsf.utn.edu.ar/cneisi2010/archivos/04Reconocimiento_de_Imgenes_SIFT.pdf
- [12] SAPIRO, GUILLERMO
 2011 *ASIFT: An algorithm for fully affine Invariant comparison*.
<http://www.ipol.im/pub/art/2011/my-asift/>
- [13] GOUSHEN YU; MOREL, JEAN-MICHEL
 2009 *ASIF: A new Framework for fully affine invariant image comparison*. IEEE International Conference on Acoustics, Speech and Signal Processing, 2009
Digital Object Identifier: [10.1109/ICASSP.2009.4959904](https://doi.org/10.1109/ICASSP.2009.4959904)
- [14] HERBER BAY; TINNE TUYTELAARS; LUC VAN GOOL
 2006 *Surf: Speeded up robust feature*. Technical report, Katholieke University Leuven.
- [15] ZHANG HUIJUAN; HU QIONG
 2011 *Fast image matching based-on improved SURF algorithm*. IEE International Conference on Electronics, Communications and Control (ICECC), 2011
Digital Object Identifier: [10.1109/ICECC.2011.6066546](https://doi.org/10.1109/ICECC.2011.6066546)
- [16] MARTIN, FISCHLER; ROBERT BOLLES
 1981 *Random sample consensus: A paradigm for model fitting with applications to image analysis and automated cartography*. Commun. ACM, 24(6):381–395.
- [17] M. MUJA; D. LOWE
 2009 *“Fast approximate nearest neighbors with automatic algorithm configuration,”* in International Conference on Computer Vision Theory and Applications (VISSAPP'09), pp. 331–340, 2009.
- [18] ROBERTO MUÑOZ
 2011 *Sistema de visión artificial para la detección y lectura de matrículas*. Universidad de Valladolid. Escuela de Ingenierías Industriales. España.
- [19] S. JUA
 2014 *Navegación de cuadrirrotor en corredores mediante la estimación de la dirección de vuelo*. Universidad Politécnica de Madrid. España, p.39