Reconstrucción de la Estructura Tridimensional de una Escena, a partir de trayectorias de cuerpos en movimiento.

Juan de Dios Alonzo Centeno

Licenciatura en Ciencias de la Computación
Facultad de Matemáticas Universidad Autónoma de Yucatán
2007

Resumen

En la presente tesis se presenta el planteamiento y desarrollo de métodos para obtener una representación de características tridimensionales de una escena en la imagen a partir de trayectorias de objetos, en nuestro caso personas en movimiento. Obtener una representación de la escena de este tipo es importante pues representa un primer paso en la obtención de una representación tridimensional de la escena. Aparte, este tipo de información nos permite desarrollar algoritmos eficientes de visión computacional que tengan requerimientos de procesamiento en tiempo real y que sean eficientes, *e.g.* para que un robot autónomo pueda tomar decisiones a partir de imágenes o para detectar comportamientos anormales a partir del estudio de las trayectorias de las personas que se mueven enfrente de la cámara.

Este trabajo tiene fundamentos en la geometría proyectiva, así que las principales referencias de la escena tridimensional en la imagen son los puntos y lineas de fuga, para poder encontrarlos se propone una metodología a seguir:

- Se propone un método adaptivo para extraer el fondo de una manera robusta, de manera que soporte cambios de iluminación naturales en la escena.
- Se propone un método de identificación, etiquetación y seguimiento de objetos en movimiento para obtener sus trayectorias.
- Se utiliza principios de geometría proyectiva para obtener puntos de fuga a partir de las trayectorias de los cuerpos en movimiento. Se utiliza la condición de que la estatura de una persona no varia al caminar.
- Se utiliza un algoritmo robusto para obtener la linea de fuga a partir de los puntos de fuga.
- Se establece la linea de fuga como una referencia para obtener información tridimensional de la escena en la imagen.

La presente tesis cuenta con una parte experimental para evaluar los métodos propuestos y establecer los parámetros adecuados para obtener resultados satisfactorios. Los experimentos también se diseñaron para comprobar que se pueden obtener características tridimensionales en la imagen, principalmente la obtención de puntos y lineas de fuga. Finalmente se presentan conclusiones y trabajos futuros de la presente tesis.

Agradecimientos

- Primero quiero darle gracias a Dios por permitirme llegar hasta este momento tan importante de mi vida y lograr una meta más en mi carrera.
- Quiero dedicar esta tesis a mis padres Armando y Doris ya que siempre me han brindado su apoyo incondicional y siempre me han ayudado a seguir adelante. Gracias papás por todo el amor que me han brindado y por lo mucho que se preocupan por mi.
- También dedico esta tesis a mi hermano sergio, se que puedo contar con el en todo momento y en general se que puedo contar con toda de mi familia.
- Agradezco a mi asesor el Dr. Arturo Espinosa, por sus enseñanzas tan valiosas para poder elaborar la presente tesis, también agradezco su paciencia y sus horas de dedicación.
- Agradezco a mi asesor el Dr Alberto Muñoz, pues siempre ha estado motivándome para seguir adelante en mis estudios, me acercó a la investigación y me ha enseñado innumerables cosas a lo largo de la licenciatura.
- Gracias a Mercy Salas por todo su apoyo y comprensión para permitirme terminar esta tesis.
- Agradezco al programa PRIORI, por el financiamiento de la presente tesis.

Declaración

Por este medio declaro que yo escribí esta tesis y que describe el trabajo de mi tesis de Licenciatura.

Juan de Dios Alonzo Centeno
Mérida, Yucatán
México
28 de abril de 2008

Índice general

Índice de figuras

Índice de cuadros

Capítulo 1

Introducción

Los seres humanos por naturaleza procesamos e inferimos información de nuestro entorno. Muchas veces sin darnos cuenta procesamos información fácilmente, cuando para una computadora es sumamente difícil; por ejemplo el análisis del lenguaje natural, reconocimiento de objetos, manipulación diestra, entre otros.

Una de las cosas que hacemos cotidianamente es obtener la estructura tridimensional de lo que nos rodea, es decir con tan solo observar nuestro ambiente automáticamente tenemos la idea de que hay un piso donde podemos caminar, que existe una pared en nuestro lado izquierdo o que hay un objeto en forma de cubo enfrente de nosotros que posee una lampara y un teléfono. A partir de dicha reconstrucción de la estructura tridimensional en nuestra mente, podemos tomar decisiones como la dirección en que debemos caminar para evadir un obstáculo. Sin embargo aunque tenemos una idea clara de la escena, normalmente no podemos decir con exactitud las medidas de los objetos tridimensionales, solo podemos dar un calculo aproximado.

El planteamiento del problema que se propone en ésta tesis, así como su solución, tiene similitudes con los varios trabajos en el área de visión computacional:

Bruce et al. (1996) explica en su libro una manera de representar la escena muy parecida a la representación echa por ojos humanos, explica el funcionamiento del *arreglo esférico óptico* de acuerdo a la teoría de Gibson, la cual consiste en una esfera alrededor del punto de visión donde inciden los rayos de luz, así como su discretización utilizando pequeñas áreas de la esfera como un solo sensor.

Pascual J. Figueroa et al. (2006) hace un seguimiento de jugadores de fútbol soccer utilizando un análisis cinemático, este artículo hace una representación y etiquetación de los jugadores de fútbol así como de sus trayectorias.

Adan Michael Baumberg (1995) nos muestra un análisis de personas caminando para construir modelos en dos dimensiones a partir de secuencias o imágenes de entrenamiento. Los resultados de este trabajo se pueden ver al ser capaz de seguir personas en movimiento en tiempo real sin el uso de equipo dedicado caro, éste trabajo brinda una forma clara de resolver el problema de seguimiento de personas, aunque esta un poco limitado a las imágenes de entrenamiento y no es muy robusto al hacer el seguimiento, es un buen antecedente y la metodología es adecuada.

Hannah Dee and David Hogg (2004) detectan eventos inusuales o interesantes en secuencias de video de personas caminando y representan zonas con obstáculos y salidas en la misma imagen a partir del estudio de las trayectorias.

1.1. Objetivos

Hay muchos trabajos que tratan sobre reconstrucción tridimensional de la escena, muchas de ellas requieren un proceso de calibración de la cámara en que se tomaron las imágenes o múltiples vistas de la imagen.

1.1.1. Objetivo general

El objetivo general de ésta tesis es obtener características tridimensionales de la escena solamente basándose en las trayectorias de los cuerpos en movimiento, es decir sin proporcionar algún parámetro que ayude este proceso. Se quiere dejar en claro que el objetivo de la presente tesis **no** es obtener un modelo tridimensional de la escena como los que acostumbran hacer los arquitectos con software especializado sino obtener *características tridimensionales de la escena en la imagen*, por ejemplo poder identificar donde se encuentran los pisos en la imagen, donde existe un pasillo o encontrar características de la cámara con que fueron tomadas las fotografías (su posición relativa a alguna referencia en la escena).

Las características tridimensionales en la escena se pueden utilizar para tomar decisiones a partir de las imágenes, por ejemplo al conocer las trayectorias de las personas y los planos y pasillos de la escena se pueden identificar comportamientos anormales de personas automáticamente o simplemente detectar intrusos en zonas no autorizadas; se pueden utilizar en proyectos de robótica, de inteligencia artificial, entre otros donde se requiera procesamiento eficiente en tiempo real .

1.1.2. Objetivos específicos

Para lograr el objetivo final, se deben de cumplir los siguientes objetivos específicos:

- Definir una forma compacta de representar una escena. Esto implica que la representación usada sea eficiente en cuestión de memoria y en cuestión de cómputo que se requiera hacer con ella.
- Hacer seguimiento robusto de objetos en movimiento. Esto implica principalmente dos cosas: resolver el problema de segmentación y el de etiquetación de manera explicita, para poder distinguir que objetos se mueven y poder obtener las trayectorias de cada objeto por separado.
- Definir una metodología para transformar las trayectorias de los cuerpos en la imagen a la representación 3D usada.

1.2. Distribución de la Tesis

Los capítulos siguientes de la presente tesis, se encuentran distribuidos de la siguiente manera:

- **Capítulo 2:** Proporciona los fundamentos en los que se basa la presente tesis, se divide en: algoritmos para la extracción del fondo, fundamentos de geometría proyectiva y de imágenes digitales.
- **Capítulo 3:** Establece la metodología de la presente tesis: Describe de una manera detallada los elementos matemáticos usados; los métodos para resolver los

problemas de extracción de fondo, reconocimiento de objetos en movimiento, seguimiento de trayectorias, obtención de características tridimensionales a partir de los objetos y sus trayectorias; y la manera de implementación de los métodos por computadora.

- **Capítulo 4:** En este capítulo se describe de una manera detallada los experimentos para probar lo descrito en los capítulos anteriores, se explican los resultados obtenidos y se dan observaciones.

- **Capítulo 5:** Por último se expresan las conclusiones y trabajo a futuro.

Capítulo 2

Fundamentos teóricos

En este capítulo se describen los fundamentos teóricos de los métodos utilizados en la presente tesis, se divide en tres secciones: extracción de fondo, geometría proyectiva e imágenes digitales.

2.1. Algoritmos para la Extración de Fondo

En una secuencia de imágenes se pueden distinguir principalmente tres clases de regiones (entenderemos por región a un conjunto de pixeles conectados entre sí):

- Regiones que cambian significativamente debido al movimiento de los objetos como es el caso de personas, automóviles, aviones, aves entre otros.
- Regiones prácticamente constantes que corresponden a objetos estáticos en la escena como son paredes, edificios pisos. También son validas aquellas regiones que corresponden a objetos que se consideran estáticos por determinado tiempo como pueden ser autos estacionados.
- Además de los dos tipos de regiones anteriores, existen regiones que cambian de una manera no significativa como es el caso de las hojas de los arboles moviéndose por el viento, o simplemente un cambio de iluminación provocado por el movimiento de las nubes.

El problema de extracción de fondo consiste en determinar aquellas regiones que permanecen constantes o su cambio no es significativo en las secuencias de imágenes. A

simple vista es un problema trivial para los seres humanos pero es muy difícil computacionalmente ya que intervienen muchos factores, entre ellos los errores de medición al obtener las imágenes (muchas veces provocados al digitalizar o al convertir los diversos tipos de formato de imagen), el factor de iluminación que no es constante, los movimientos a distintas velocidades y sobre todo aquellas regiones que no cambian de una manera significativa.

Existen numerosos tipos de algoritmos para la extracción del fondo, por simplicidad los clasificaremos en adaptivos y no adaptivos: Los modelos no adaptivos son aquellos que necesitan de parámetros de inicialización y no se modifican a lo largo del proceso de secuencias de imágenes. Los adaptivos son aquellos en que se van modelando los cambios naturales en la escena principalmente la iluminación es decir sus parámetros van cambiando a lo largo en que se procesan las secuencias de imágenes.

La utilización de métodos adaptivos para la extracción del fondo es de mucha ayuda para obtener buenos resultados, por ejemplo si utilizamos un método no adaptivo se tienen que definir parámetros como la iluminación y por tal motivo los experimentos se tienen que hacer en escenas controladas siendo difícil la utilización de dicho tipo de métodos en ambientes con luz solar por largo tiempo debido a los cambios de iluminación.

Los modelos adaptivos simples calculan un modelo del fondo haciendo un promedio de una secuencia de imágenes previas a la imagen actual, posteriormente se calcula la diferencia entre el modelo y la imagen actual para determinar el fondo y las partes en movimiento. Lo anterior implica que los objetos en movimiento lo hagan a una velocidad no lenta y que sea visible la mayoría del fondo, además los cambios de iluminación son un problema en la mayoría de los modelos.

2.1.1. Algoritmo *EM*

Un algoritmo muy conocido es el llamado *EM* (Expectative-Maximization por sus siglas en Inglés), este algoritmo puede utilizarse en muchas problemas en donde se quiere estimar un conjunto de datos, para ello se utilizan técnicas estadísticas como es el caso de estimadores de Máxima-Verosimilitud. Más información sobre el algoritmo

EM puede encontrarse en el libro de Christopher M. Bishop (1995) y en el libro de David A. Forsyth and Jean Ponce (2003). A continuación se explicara una adaptación del algoritmo *EM* para resolver una segmentación de la imagen.

El problema de segmentación de una imagen consiste en dividir una imagen en G segmentos con características comunes, por ejemplo si segmentamos una imagen que contiene un cuadro de un color y lo demás de otro color en dos segmentos, cada segmento de la imagen contrendria la parte correspondiente a un color. Supongamos que tenemos un conjunto de n pixeles, los datos que queremos encontrar forman un arreglo de n por G donde G es el número de segmentos en que queremos segmentar la imagen, llamemos a este arreglo I y nos indicara la probabilidad en que un pixel pertenezca al segmento g (A este arreglo también se le conoce como mapa de apoyo).

Los segmentos se modelan con distribuciones Gaussianas y a cada una se la asigna un peso (α), los parámetros de las G Gaussianas (varianza Σ y media μ) se representan con la variable $\theta = (\Sigma, \mu)$ y los pesos con la variable α. La variable $\Theta = (\alpha_1, ..., \alpha_g, \theta_1, ..., \theta_g)$ representa en conjunto los parámetros que modelan las Gaussianas y sus pesos.

El algoritmo *EM* consta de dos fases que se repiten hasta encontrar resultados satisfactorios, las fases son la E y la M.

La fase E del algoritmo consiste en estimar cada uno de los elementos del arreglo I, para ello se utiliza la siguiente formula:

$$\bar{I}_l m = \frac{\alpha_m^{(s)} P_m(x_l \mid \theta_m^{(s)})}{\sum_{g=1}^{G} \alpha_g^{(s)} p_k(x_l \mid \theta_l^{(s)})} \tag{2.1}$$

donde $\bar{I}_l m$ significa la probabilidad estimada de que el pixel l pertenezca al segmento m, x_l es el l-ésimo pixel, la notación $\alpha_m^{(s)}$ significa el valor de α_m en la s-ésima iteración, de la misma manera para θ_m y las variables con la misma notación. El valor $p_i(x, \theta_i)$ corresponde a la siguiente formula que corresponde a una Gaussiana común:

$$p_i(x, \theta_i) = \frac{1}{(2\pi)^{\frac{d}{2}} |\Sigma_i|^{\frac{1}{2}}} e^{-\frac{1}{2}(x-\mu_i)^T \Sigma_i^{-1} (x-\mu_i)} \tag{2.2}$$

La fase M consiste en actualizar los valores de los parámetros de las Gaussianas y de los pesos de las mismas, para ello se utilizan estimadores de máxima verosimilitud. Las formulas para actualizar los parámetros se muestran a continuación:

$$\alpha_m^{(s+1)} = \frac{1}{n}\sum_{l=1}^{n} p(m \mid x_l, \Theta^{(s)}) \tag{2.3}$$

$$\mu_m^{(s+1)} = \frac{\sum_{l=1}^{n} x_l p(m \mid x_l, \Theta^{(s)})}{\sum_{l=1}^{n} p(m \mid x_l, \Theta^{(s)})} \tag{2.4}$$

$$\Sigma_m^{(s+1)} = \frac{\sum_{l=1}^{n} x_l p(m \mid x_l, \Theta^{(s)}) \left\{(x_l - \mu_m^{(s)})(x_l - \mu_m^{(s)})^T\right\}}{\sum_{l=1}^{n} p(m \mid x_l, \Theta^{(s)})} \tag{2.5}$$

donde $p(m \mid x_l, \Theta^{(s)})$ es el valor del m-ésimo mapa de apoyo para el pixel l, denotado anteriormente como I y actualizado en la fase E.

En la adaptación del algoritmo *EM* en la segmentación de imágenes se entiende fácilmente su funcionamiento, modela los segmentos como funciones de distribución Gaussinanas y encuentra los parámetros óptimos de las mismas. Si vemos una secuencia de video como una misma imagen multiplicada por una constante de iluminación y los pixeles de los cuerpos en movimiento se consideran como un ruido que se le añade a la misma imagen , podemos aplicar el mismo algoritmo *EM* para extraer el fondo, solo que ahora se modelaría el fondo con una sola Gaussiana y lo que no abarque la Gaussiana representaría el ruido añadido o en otras palabras lo que no es fondo de la imagen.

En este trabajo se utilizo un método similar al algoritmo *EM* descrito en el artículo de Chirs Stauffer and W.E.L Grimsom (1999). Se modelo cada pixel en la imagen con un conjunto de Gaussianas (no solamente con una como sucedería en un *EM* estricto), basándonos en la persistencia y varianza del modelo, se escogen cuales Gaussianas representan el fondo y cuales no, un pixel que no se ajusta a alguna de las distribuciones Gaussianas del fondo se considera como un pixel en movimiento y sera incluido en el fondo hasta que alguna de las Gaussianas lo incluya. Este método hace menos cálculos que un *EM* más estricto ya que incluye una constante de aprendizaje, además al utilizar varias Gaussianas para modelar el fondo, se pueden modelar fondos mas complejos por ejemplo el fondo de las olas del mar, o las imágenes tomadas a un monitor CRT, pues aunque presentan cambios en los pixeles, estos se modelan con dos Gaussianas distintas.

2.2. Geometría Proyectiva

Gran parte de la presente tesis esta basada en la adquisición de imágenes y en principios de geometría proyectiva descrita en el libro de Richard Hartley and Andrew Zisserman (2000), a continuación se explicara en breve algunos conceptos fundamentales manejados en la presente tesis.

2.2.1. Adquisición de imágenes

En objeto de estudio de la presente tesis son las escenas fotografiadas con personas caminando, es por esto que debemos comprender como se generan las imágenes a partir del mundo tridimensional. La cámara fotográfica más simple recibe el nombre de cámara estenopeica, se puede considerar que fue la primera en ser construida y su funcionamiento describe de una manera sencilla la forma de adquirir imágenes. Una cámara estenopeica consiste en una caja que no permita la entrada de luz, esta caja cuenta con una película o papel fotográfico en una de sus paredes interiores y un pequeño agujero en la pared opuesta. Para que se produzca la fotografía se requiere que el agujero sea abierto por un determinado tiempo para que los rayos de luz entren en la caja y formen la imagen en la película o papel fotográfico. Véase figura 2.1; el tiempo de adquisición y la nitidez de la fotografía dependen del tamaño del agujero.

La proyección de un objeto es la figura que se obtiene al dirigir todas las líneas proyectantes desde dicho objeto hacia un plano.Existen diferentes tipos de proyecciones que se engloban en dos tipos generales: las proyecciones geométricas planas en perspectiva y las proyecciones geométricas planas paralelas. Una proyección paralela, es aquella en la cual las líneas proyectantes de cada uno de los puntos del objeto son paralelas entre si, debido a que el centro de proyección esta ubicado a una distancia infinita del plano de proyección, en el plano de la proyección se conserva el paralelismo. En la proyección en perspectiva las líneas de proyección no son paralelas, ya que esta determinada por el centro de proyección ubicado a una distancia finita. Motivo por el cual los objetos en el plano de visión se transforman a lo largo de trayectorias convergentes en un punto de proyección, de manera que el tamaño de los objetos disminuye con la distancia de forma no uniforme y se pierde el paralelismo. En la figura 2.2 se muestra un diagrama

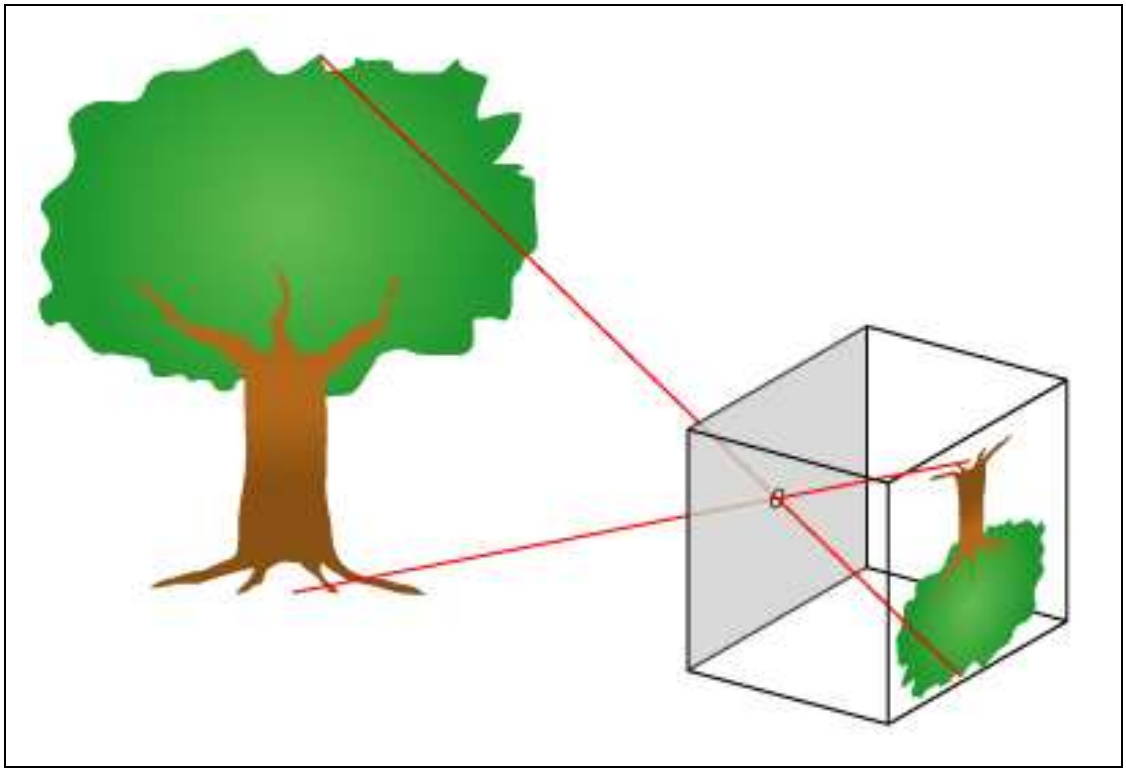


Figura 2.1: Cámara estenopeica

ejemplificando ambos tipos de proyecciones.

En realidad el funcionamiento de una cámara estenopeica es una proyección perspectiva conocida comúnmente como *modelo pinhole*. En la figura 2.3 se muestra una representación del modelo pinhole.

Matemáticamente se pueden obtener la conversión de puntos tridimensionales a puntos en la proyección dando como resultados las formulas 2.6 y 2.7

$$x = X_c f / Zc \tag{2.6}$$

$$y = Y_c f / Zc \tag{2.7}$$

2.2.2. Geometría Proyectiva

En nuestra vida cotidiana, sin darnos cuenta estamos familiarizados con la geometría proyectiva, por ejemplo si fotografiamos una pelota esta no estará completamente circular en la fotografía, de la misma manera ocurre con los cuadrados y rectángulos los cuales pierden sus propiedades de paralelismo al ser fotografiados, sin embargo no nos

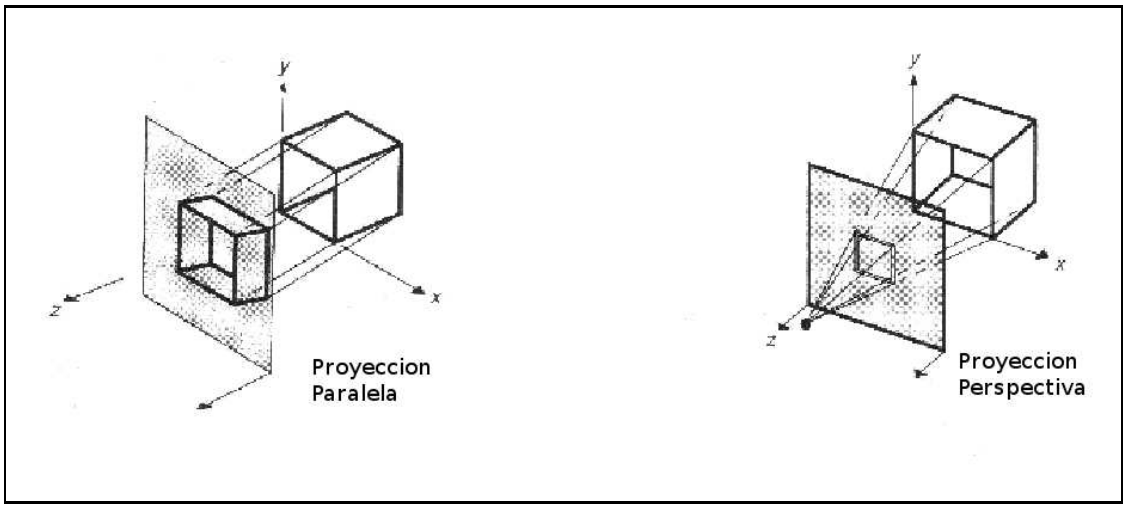


Figura 2.2: Proyecciones

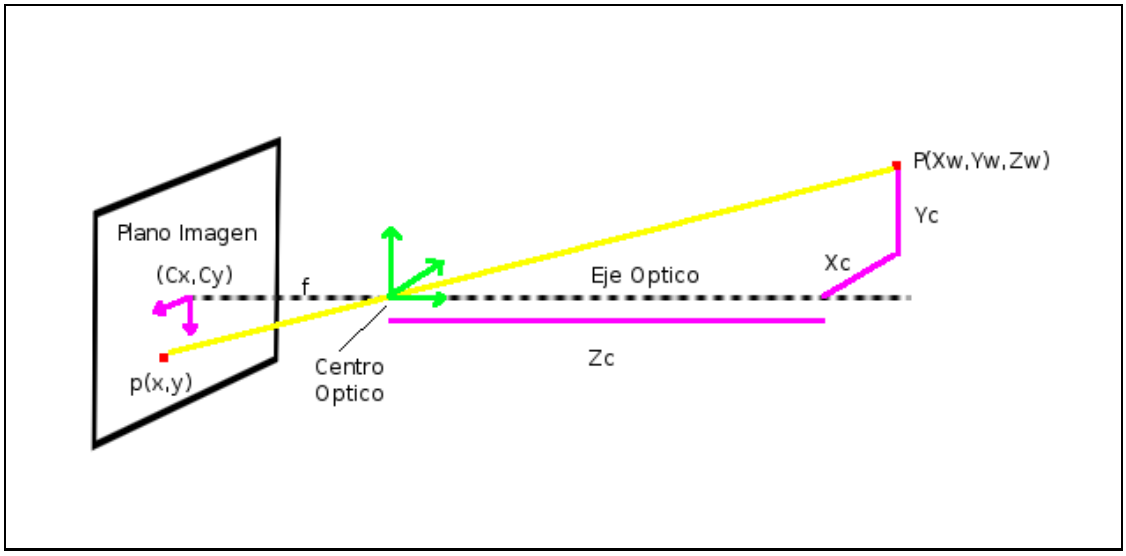


Figura 2.3: Modelo Pinhole

damos cuenta y nuestro cerebro interpreta esta transformación. Las transformaciones proyectivas se encargan de mapear el mundo tridimensional que nos rodea en un plano en dos dimensiones.

La representación de puntos en el espacio proyectivo esta dada por vectores en coordenadas homogéneas, para entender como se forman dichos vectores partiremos de puntos de un plano en el espacio euclidiano (plano cartesiano) y los representaremos en coordenadas homogéneas:

- **Representación homogénea de puntos.-** Un punto en el espacio euclidiano $(x, y)^T$ se dice que esta en la linea $l = (a, b, c)^T$, Si y solo si $ax+by+c = 0$. Esto se puede escribir algebraicamente por el siguiente producto escalar $(x, y, 1)(a, b, c)^T$. Así el punto en el espacio euclidiano se representa en forma homogénea añadiéndole un uno al vector $(x, y)^T$ quedando así $(x, y, 1)^T$, cualquier vector de la forma $(kx, ky, k)^T$ representan el mismo punto $(x, y, 1)^T$ siempre y cuando $k \neq 0$. De lo anterior un punto en coordenadas homogéneas $(x, y, w)^T$ se convierte en coordenadas cartesianas dividiendo los dos primeros elementos del vector entre el tercero dándonos como resultado el par $(x/w, y/w)$, si $w = 0$ entonces el resultado de hacer la transformación al plano euclidiano nos daría lo que comúnmente conocemos como un punto infinito y no lo podemos representar en el plano euclidiano, sin embargo en el plano proyectivo si. Los puntos de la forma $(x, y, 0)^T$ del espacio proyectivo se conocen como punto ideales.

- **Representación homogénea de lineas.-** Una linea en un plano se representa por la formula $ax+by+c = 0$, diferentes valores de a , b y c nos dan distintas lineas, así una linea en coordenadas homogéneas se representa por el vector $(a, b, c)^T$. Además como la linea $ax + by + c = 0$ es la misma que $kax + kby + kc = 0$, $(a, b, c)^T$ y $k(a, b, c)^T$ representan la misma linea.

- **Intersección de lineas.-** Dadas dos lineas en coordenadas homogéneas $l = (a, b, c)^T$ y $l' = (a', b', c')^T$, su intersección esta dada por su producto cruz $x = l \times l'$. Se demuestra fácilmente por la triple identidad del producto escalar $l \cdot (l \times l') = l' \cdot (l \times l') = 0$ si sustituimos $x = (l \times l')$ entonces tenemos que $lx = l'x = 0$ lo que nos indica que el punto x esta en ambas lineas o dicho de otra forma es la

intersección de las lineas l y l'.

- **Linea que une dos puntos.-** El producto cruz de dos puntos p y p' nos da la linea que los une: $l = p \times p'$. Se demuestra fácilmente por la misma identidad del producto escalar: $p \cdot (p \times p') = p' \cdot (p \times p') = 0$, sustituyendo $p \cdot (l) = p' \cdot (l) = 0$ lo que indica que la linea atraviesa tanto por el punto p como por el punto p'.

La acción proyectiva de una cámara sobre un punto en el espacio puede ser expresado en términos de un mapeo linear de coordenadas homogéneas la cual se expresa en la ecuación 2.8, donde la matriz P_{3x4} es conocida como la matriz de la cámara.

$$\begin{bmatrix} x \\ y \\ w \end{bmatrix} = P_{3x4} \begin{bmatrix} X \\ Y \\ Z \\ T \end{bmatrix} \tag{2.8}$$

2.2.3. Puntos y Lineas de Fugas

Dos principales conceptos del espacio proyectivo que nos dan información de la escena tridimensional son los siguientes:

- **Punto de Fuga (punto evanescente).-** Si tenemos un par de lineas paralelas en la escena tridimensional al proyectarse se intersectan en el Punto de Fuga o Punto Evanescente. También se le denominan puntos ideales en el espacio proyectivo o puntos al infinito en la escena tridimensional.
- **Linea de Fuga (linea evanescente).-** Conocida también como horizonte, es la linea formada por la infinidad de puntos de Fuga. En esta línea se intersectan se interceptan los planos paralelos de la escena tridimensional.

2.3. Imágenes Digitales

Debido a que procesamos imágenes digitales, en esta sección se explican algunos conceptos tomados del libro de Rafael C. Gonzalez and Richard E. Woods (2002), del

libro de David A. Forsyth and Jean Ponce (2003) y del libro de Milan Sonka et al. (1993).

Una imagen es un registro de valores organizados en forma bidimensional, generalmente representando intensidad de radiación electromagnética. Para obtener imágenes digitales es necesario convertir los valores captados por los dispositivos de adquisición (cámaras fotográficas) en una representación digital, esto involucra dos procesos principales: muestreo y cuantificación. Una imagen ideal debería ser continua con respecto al eje x y al eje y sin embargo al almacenarla en un formato digital es necesario hacer un *muestreo* y representar los valores infinitos en ambos ejes en un número predeterminado de "pixeles"que conforman la imagen digital.

De la misma manera del muestreo, la intensidad de radiación electromagnética al ser digitalizada es *cuantificada* en un número predeterminado de valores para representarla, es decir si hablamos de imágenes digitales en escala de grises representamos el negro con 0 y el blanco con 255. Otra cuantificación se realiza al tener imágenes digitales a colores pues cuantificamos el espectro en tres colores: rojo, verde y azul.

Cuando cuantificamos los valores de la intensidad de radiación electromagnética en solo dos valores (0 y 1) estamos hablando de una imagen binaria, sirven principalmente para distinguir regiones particulares de una escena (podría representar personas del resto de la escena). El concepto de conectividad se aplica principalmente en imágenes binarias, un pixel no esta conectado si este esta en 1 y ninguno de sus vecinos lo esta. Los vecinos de un pixel se clasifican en:

- Vecindan en 4: corresponde a los vecinos horizontales y verticales del pixel $P = (x, y)$ los cuales son: $(x + 1, y)$, $(x - 1, y)$,$(x, y + 1)$,$(x, y - 1)$. Estos vecinos se denotan como $N_4(P)$.
- Vecindad en diagonal: corresponde a los vecinos diagonales del pixel $P = (x, y)$ los cuales son: $(x+1, y+1)$,$(x+1, y-1)$,$(x-1, y+1)$,$(x-1, y-1)$. Estos vecinos se denotan como $N_D(P)$.
- Una tercera vecindad es el conjunto formado por $N_4(P) \cup N_D(P)$ y se denomina como $N_8(P)$.

Debido a los errores de cuantificación y muestreo, muchas veces es necesario mejorar las imágenes digitales antes de trabajar con ellas; una forma de mejorar las imágenes es filtrarlas, existen filtros trabajando solamente con los valores de la imagen (filtros espaciales) y también existen filtros trabajando con las frecuencias en las mismas (conocidos como filtros de Fourier); una de las maneras de mejorar la imagen que se utilizo en la presente tesis es la aplicación de un tipo de filtro espacial conocido como *Filtro Gaussiano* que se describirá a continuación.

Para aplicarle un filtro espacial a una imagen digital una ventana deslizante, conocida como *máscara*, se centra en cada pixel de una imagen de entrada y genera nuevos pixeles de salida. Para aplicar la máscara a esa zona se multiplican los valores de los puntos que rodean al píxel que estamos tratando por su correspondiente entrada o coeficiente en la máscara y luego se suman esos productos. El resultado es el nuevo valor para el píxel central (véase figura 2.4). Este proceso de evaluación ponderada de la vecindad del pixel se le conoce como çonvolución"bidimensional y la matriz como "kernel o ventana de convolución".

En particular el filtro Gaussiano es un filtro espacial cuya máscara es una función de distribución Gaussiana dada por la ecuación 2.9, y sus efectos se aprecian el la figura 2.5

$$G(x,y) = Kexp(-r^2/2\sigma^2),\ r^2 = x^2 + y^2 \tag{2.9}$$

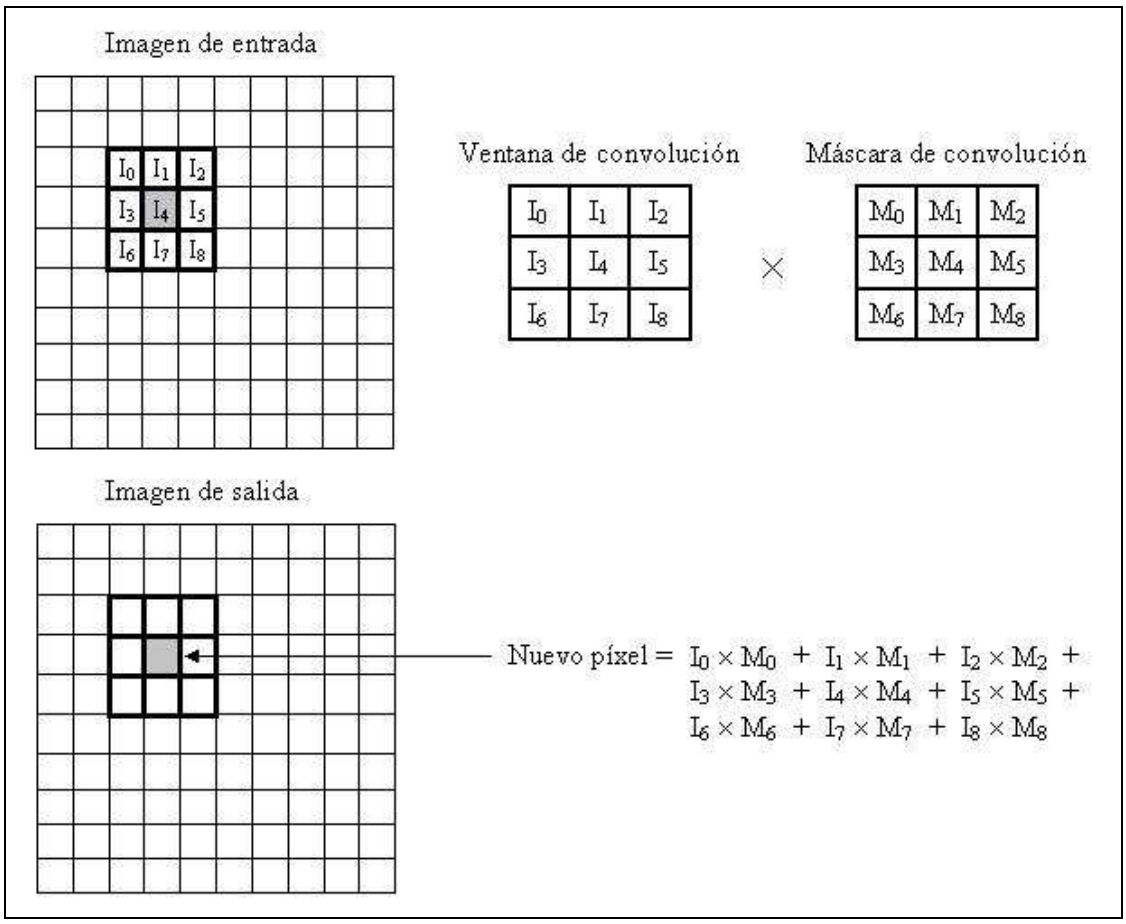


Figura 2.4: Aplicación de un filtro espacial

(a) Imagen Original

(b) $\sigma = 2$ (c) $\sigma = 4$ (d) $\sigma = 8$

Figura 2.5: Ejemplo de utilización de filtrados Gaussianos

Capítulo 3

Metodología.

En este capítulo se describe la metodología seguida en la presente tesis, se describen de una manera detallada los métodos para resolver los problemas de extracción de fondo, reconocimiento de objetos en movimiento, seguimiento de trayectorias, obtención de características tridimensionales como son la línea y puntos de fuga así como la manera de implementar estos métodos por computadora.

3.1. Algoritmo Adaptivo de Extracción de Fondo

Como se mencionó en el capítulo anterior, para la extracción del fondo de la imagen se opto por un método adaptivo que lo modela usando un conjunto de distribuciones Gaussianas (Chirs Stauffer and W.E.L Grimsom, 1999). A continuación se describe mas a detalle este método y su implementación.

Para el método de extracción de fondo se consideró trabajar independientemente en cada pixel, así que la representación del historial de cada pixel a lo largo del tiempo es un arreglo de imágenes, por simplicidad y para el propósito de la presente tesis nos limitamos a trabajar en imágenes en escala de grises.

Cada pixel en una imagen representa la cantidad de luz reflejada por la porción de un objeto en la escena, así que los pixeles que representan el fondo de una escena con fondo e iluminación constante pueden ser modelados con una simple distribución Gaussiana centrada en el valor medio del pixel, sin embargo en la realidad hay cambios de iluminación y es por esto que la distribución Gaussiana debe de adaptarse a estos

cambios, más aun, hay ocasiones en que una simple Gaussiana no es suficiente para modelar el fondo, este es el caso cuando tenemos fondos no tan estáticos como es una escena a la orilla del mar.

Por lo anterior, el historial del pixel (x_0, y_0) en el tiempo t se definirá de ahora en adelante como:

$$\{X_1, X_2, ..., X_t\} = \{I(x_0, y_0, i) : 1 < i < t\} \tag{3.1}$$

De la misma manera, el historial mas reciente de un pixel $\{X_1, X_2, ..., X_t\}$ se modela con un conjunto de K Gaussianas que deberán adaptarse y dar mas peso a las observaciones recientes. La probabilidad de observar el valor de un pixel en el tiempo t es:

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, \Sigma i, t) \tag{3.2}$$

donde K es el número de distribuciones Gaussianas, ω es el peso o proporción de datos que son soportados por la iésima Gaussiana en el tiempo t, μ y Σ son la media y matriz de coovarianza de la iésima Gaussiana y η es una función de densidad de probabilidad Gaussiana:

$$\eta(X_t, \mu, \Sigma) = \frac{1}{(2\pi)^{\frac{n}{2}} |\Sigma|^{\frac{1}{2}}} e^{-\frac{1}{2}(X_t - \mu)^T \Sigma^{-1} (X_t - \mu)} \tag{3.3}$$

La matriz de coovarianza Σ, es de 3 renglones por tres columnas si consideramos imágenes a color, sin embargo como ya se había mencionado, por simplicidad trabajaremos solamente con imágenes en escala de grises entonces podemos simplificar las ecuaciones 3.2 y 3.3 reduciendo la matriz de coovarianza a una simple varianza (σ^2), de esta manera las ecuaciones se reducen a:

$$P(X_t) = \sum_{i=1}^{K} \omega_{i,t} * \eta(X_t, \mu_{i,t}, {\sigma_{i,t}}^2) \tag{3.4}$$

$$\eta(X_t, \mu, \sigma^2) = \frac{1}{\sigma\sqrt{2\pi}} e^{-\frac{(X_t - \mu)^2}{2\sigma^2}} \tag{3.5}$$

Para ir cambiando los parámetros de las Gaussianas conforme hay nuevas imágenes se sigue el siguiente algoritmo para cada pixel:

1. Cada pixel X_t es comparado con las K distribuciones Gaussianas hasta que se encuentre una en la cual el pixel este a una máxima distancia de 2.5σ de la media de dicha Gaussiana:

$$|X_t - \mu_t| < 2.5\sigma \tag{3.6}$$

2. Si no se encontró alguna Gaussiana que contenga al pixel X_t , se sustituye la Gaussiana que tenga menos probabilidad por una nueva con $\mu = X_t$, una varianza alta y un peso bajo.

3. Los pesos de las K distribuciones son actualizadas con la siguiente formula.

$$\omega_{k,t} = (1 - \alpha)\omega_{k,t-1} + \alpha(M_{k,t}) \tag{3.7}$$

   donde $M_{k,t}$ es uno cuando hubo coincidencia con alguna Gaussiana o cero de otra forma. α es la constante de aprendizaje de este método.

4. Los parámetros μ y σ de las Gaussianas que no tuvieron coincidencia con el pixel permanecen iguales. Los parámetros de la Gaussiana que si tuvo coincidencia se actualizan de la siguiente manera:

$$\mu_t = (1 - \rho)\mu_{t-1} + \rho X_t \tag{3.8}$$

$$\sigma_t^2 = (1 - \rho)\sigma_{t-1}^2 + \rho(X_t - \mu_t)^2 \tag{3.9}$$

   donde ρ es una segunda variable de aprendizaje y esta dada por:

$$\rho = \alpha\eta(X_t|\mu_k, \sigma_k) \tag{3.10}$$

Además de actualizar los parámetros de las Gaussianas es necesario establecer un criterio para determinar cuando un pixel es fondo y cuando no, para lo cual definiremos lo siguiente:

- Para cada pixel ordenamos de mayor a menor las Gaussianas correspondientes tomando como criterio por del valor ω/σ, ya que este valor incrementa cuando hay mas peso y cuando la varianza decrece.

- Escogemos B distribuciones que nos van a modelar el fondo, para ello se hace la suma acumulativa de los pesos de las primeras B distribuciones de nuestras Gaussianas ordenadas que nos den un valor menor a un parámetro T o en forma matemática:

$$B = argmin_b \left(\sum_{k=1}^{b} \omega_k > T \right) \tag{3.11}$$

- Si el pixel X_t pertenece a una Gaussiana dentro del conjunto de las B Gaussianas (ecuación 3.6), entonces este pixel corresponde al fondo de la imagen, de lo contrario es muy probable que este pixel corresponda a un objeto en movimiento.

Como podemos apreciar en las formulas anteriores, los únicos parámetros para este algoritmo son el número de Gaussianas K , el factor de aprendizaje α y la constante T (ecuación 3.11), sin embargo se le pueden añadir otros procedimientos para tener mejores resultados que requieren más parámetros, estos procedimientos podrían ser evaluación de conectividad, filtrados Gaussianos antes del proceso y después de calcular el fondo.

3.1.1. Implementación del Algoritmo

La implementación del algoritmo para calcular el fondo se hizo a través de una clase llamada *EM* por su semejanza al algoritmo *EM* (vease sección 2.1.1).

El diagrama de dicha clase se muestra en la figura 3.1.1, en él se aprecian los atributos públicos (marcados con +) que representan los parámetros del algoritmo, los atributos *Var*, *Med* y *Ais* representan los parámetros y pesos de cada una de las K Gaussianas(σ , μ y ω respectivamente en las formulas 3.4 y 3.5). El Vector *Cola* guarda una arreglo circular de las últimas N imágenes procesadas.

Antes de hacer las iteraciones del algoritmo se inicializan la *Cola* y los parámetros del algoritmo, para esto se utiliza el método llamado *IniciaColas*.

El método *IngresaFrame* inserta una nueva imagen en la clase y después se utiliza el método *Itera* para hacer una iteración del algoritmo incluyendo la nueva imagen. El método *CalculaBG* extrae el fondo de la escena y para visualizar el estado actual del mismo se utiliza el método *PrintBG*.

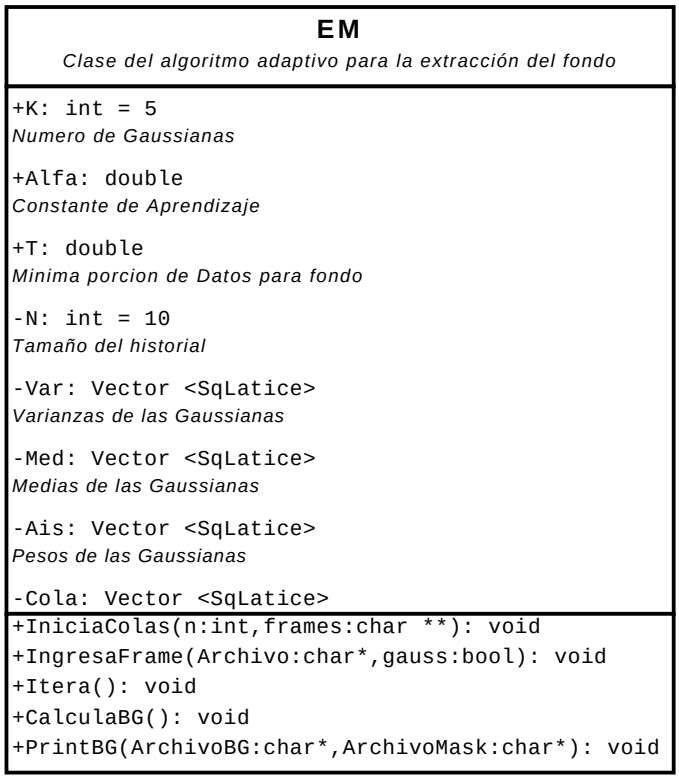


Figura 3.1: Diagrama UML de la clase *EM*

3.2. Extracción de *Blobs*

Llamaremos *blobs* a un conjunto de pixeles conectados en la imagen que representan con una alta probabilidad un objeto en movimiento, en esta tesis los objetos en movimiento son personas caminando. En la presente tesis el estudio de los *blobs* es fundamental ya que nos permiten determinar las trayectorias de los objetos en movimiento y a partir de ella inferir información tridimensional de la imagen tomada. A continuación se describirá el proceso de extracción de *blobs* de la imagen:

1. Se aplica el proceso de extracción de fondo (descrito anteriormente en la sección 3.1) para obtener una imagen binaria en la cual cada pixel pueda tomar el valor de cero o uno indicando si pertenece al fondo o es un pixel en movimiento respectivamente, el resultado esperado se puede ver en la figura 3.2

2. Teniendo la imagen binaria de las partes en movimiento se procede a analizar los pixeles no conectados (véase sección 2.3). En la presente tesis se utilizo principalmente la vecindad $N_8(P)$, y aquellos pixeles no conectados se descartan para

Figura 3.2: Ejemplo de la extracción de fondo de una imagen y su representación en una imagen binaria

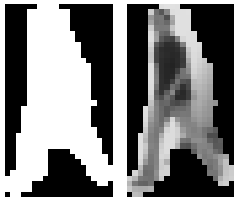


Figura 3.3: Ejemplo de un *blob* extraído, se muestra en imagen binaria y con un empalme de textura

extraer los *blobs*.

3. Se procede ahora a analizar los pixeles conectados, un conjunto de pixeles conectados entre si forman una región conectada de pixeles, la cual llamaremos *blob*. Para la presente tesis se utilizo la conectividad en 8 para extraer los *blobs*. Una imagen de un *blob* extraído se puede apreciar en la figura 3.3.

4. Dichos *blobs* se guardan en una estructura adecuada para procesarlos posteriormente.

3.2.1. Implementación de la extracción de *blobs*

La clase *Blob_ Item*, cuyo diagrama uml se muestra en la figura 3.4, se utilizó para representar un *blob* extraído de una imagen y manipularlo adecuadamente. Básicamente consiste en dos imágenes: una imagen binaria del *blob* y una con información de la

intensidad luminosa del mismo (textura), las coordenadas de su posición en la imagen original, las coordenadas del centro geométrico del *blob* y de los puntos de mayor y menor ordenada que representan la cabeza y el pie de una persona representada por el *blob*. El único método establecido calcula todos los parámetros a partir de la imagen binaria que resulta de extraer el fondo de la escena. La clase *Blob* mostrada en el diagrama 3.5 se utilizó para clasificar los *blobs* encontrados en diferentes imágenes, de tal manera que los *blobs* que se encontraron en imágenes consecutivas que correspondan a una misma persona en movimiento se almacenen en una misma clase y con esto podamos definir su trayectoria.

Esta clase contiene un atributo que guarda el historial de la persona representada por el *blob* (*Historical_ Map*), dicho atributo establece una relación entre el *blob* y el numero de imagen del que fue extraído con esto establecemos la trayectoria del objeto.

Los métodos que contiene esta clase sirven para determinar si una instancia de la clase *Blob_ Item* extraída de la imagen N representa al mismo objeto en movimiento que representa la clase *Blob* y si es así lo añade al mapa histórico.

Para determinar si un *Blob_ Item* pertenece a un objeto *Blob* se escogió como principal factor la proximidad de sus centros geométricos debido a que consideramos que de un frame a otro ha pasado muy poco tiempo y que el movimiento de la persona no es mucho en dicho tiempo, otro factor importante es el área que ocupa un *blob* ya que por la misma justificación no se permiten grandes incrementos o decrementos en el área del *blob*. Para determinar falsos *Blobs* se estableció un área mínima que debe de cubrir dicho *blob*.

En resumen, si en nuestra escena aparecen N objetos moviéndose, deberíamos tener exactamente N instancias de *Blob* y cada instancia de *Blob* debería tener en su atributo *Historical_ Map* un elemento *Blob_ Item* por cada vez que el objeto en movimiento representado por *Blob* aparezca en la escena, es decir si el objeto en movimiento solo aparece en 5 frames, el mapa histórico solamente debería tener 5 elementos.

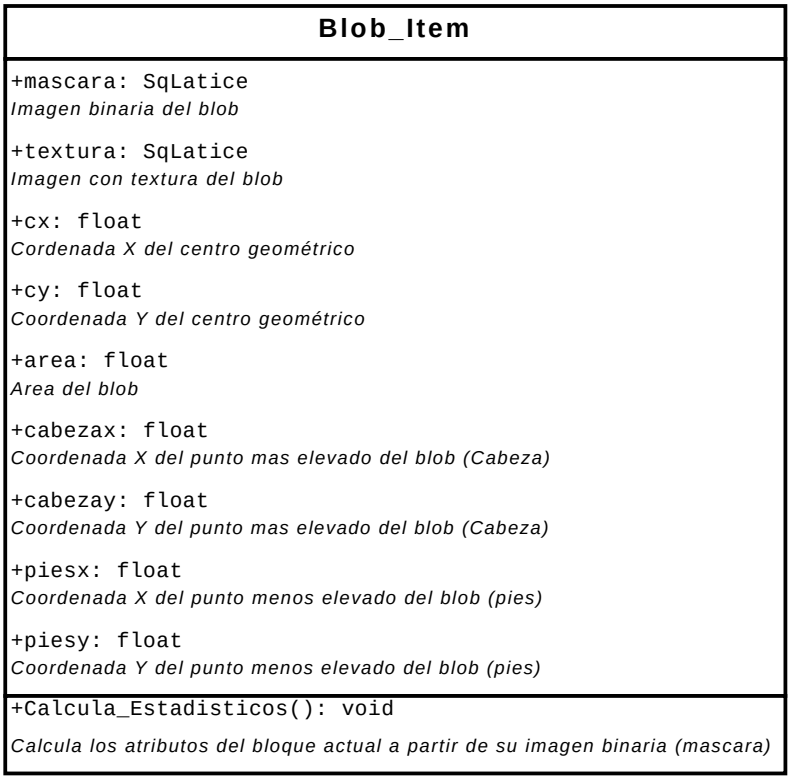


Figura 3.4: Diagrama UML de la clase BlobItem

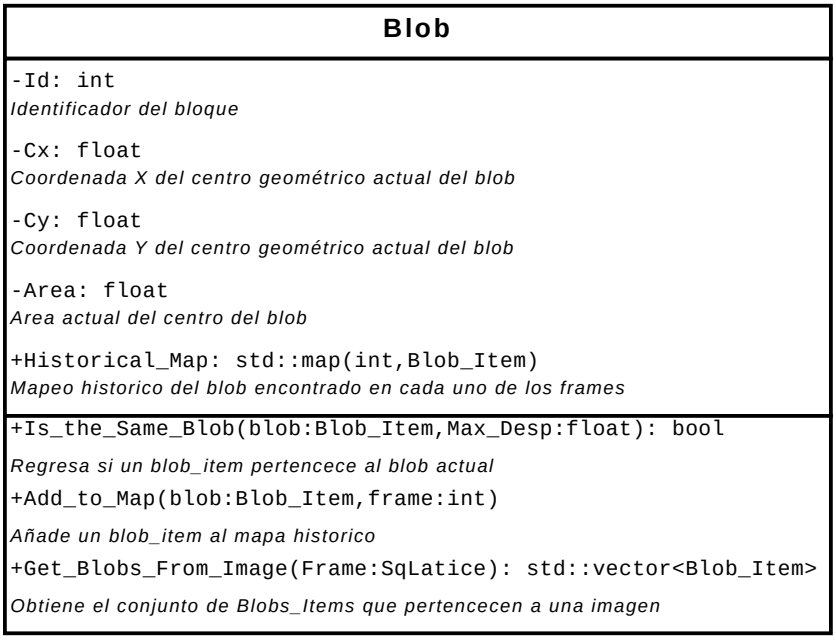


Figura 3.5: Diagrama UML de la clase *Blob*

(a) Posición 1 (b) Posición 2 (c) Combinando Posiciones

Figura 3.6: Encontrando lineas paralelas en el espacio 3D

3.3. Puntos y Lineas de Fuga

Como se explicó en la introducción a la geometría proyectiva (Sección 2.2), los puntos y lineas de fugas son conceptos que nos ayudaran a obtener una estimación de la estructura tridimensional de la escena.

3.3.1. Obtención de punto de fuga

Podemos encontrar un punto de fuga al proyectar dos lineas paralelas en la escena tridimensional y buscar su intersección en el espacio proyectivo, así nuestro problema se resuelve si podemos encontrar lineas paralelas.

Si consideramos las restricciones de que una persona camina generalmente en superficies planas y que al caminar su estatura no varia, entonces podemos considerar que la linea que unen los puntos superiores de la cabeza y la linea que une los pies de una misma persona en dos posiciones distintas son paralelas en la escena tridimensional pero que al proyectarse ya no lo son (ver Figura 3.6).

Como se explico en la sección 3.2, un objeto de la clase *Blob* nos representa a una persona en movimiento y además guarda su historial a lo largo del tiempo, así podemos obtener las combinaciones de pares de posiciones distintas tomadas de dos en dos y obtener sus puntos de fuga al hacer la intersección de las lineas que consideramos paralelas y obtener su punto de fuga de la siguiente manera: Sea (X_{C1}, Y_{C1}) y (X_{P1}, Y_{P1}) las coordenadas en la imagen del punto superior de la cabeza y del punto inferior de los

pies de una persona y (X_{C2}, Y_{C2}) y (X_{P2}, Y_{P2}) sus correspondientes en otra posición, entonces las lineas que unen los puntos de las cabezas y de los pies están dados por las ecuaciones 3.12 y 3.13 (Véase sección 2.2.2).

$$L_C = \begin{bmatrix} X_{C1} \\ Y_{C1} \\ 1 \end{bmatrix} X \begin{bmatrix} X_{C2} \\ Y_{C2} \\ 1 \end{bmatrix} \tag{3.12}$$

$$L_P = \begin{bmatrix} X_{P1} \\ Y_{P1} \\ 1 \end{bmatrix} \times \begin{bmatrix} X_{P2} \\ Y_{P2} \\ 1 \end{bmatrix} \tag{3.13}$$

Y el punto de fuga se calcula con la intersección de las lineas L_C y L_P (Véase la formula 3.14).

$$PF = (L_C) \times (L_P) \tag{3.14}$$

3.3.2. Obtención de la linea de Fuga

Habiendo obtenido los puntos de fuga, estos teóricamente deben ser colineales y la linea que los unen debe ser la linea de fuga, sin embargo debido a la discretización no se espera que sean exactamente colineales por lo cual para calcular la línea de fuga se puede hacer una estimación que se obtiene bajo un criterio de mínimos cuadrados y obtener una línea aproximada.

3.4. Extracción Robusta de la Linea de Fuga

Como se explico en la sección anterior (véase sección 3.3) antes de obtener la linea de fuga necesitamos obtener varios puntos de fuga y en base a ellos calcular la linea que los une. Al obtener los puntos de fuga no esperamos que queden perfectamente colineales, además esperamos que exista ruido por la propagación del error de las mediciones generando puntos atípicos (*outliers*) que perjudiquen el calculo de la linea de fuga; es por esto que necesitamos pensar en un método robusto para estimarla, un método que nos puede ayudar es el llamado RANSAC el cual describiremos a continuación.

RANSAC (RANdom SAmple Consensus, por sus siglas en ingles) es un método iterativo para estimar parámetros de un modelo matemático a partir de un conjunto de

datos observados los cuales pueden contener *outliers* . Para entender el funcionamiento del método RANSAC, lo describiremos en su forma más simple:

Dado un conjunto de puntos bidimensionales, se encontrará la línea que minimice la suma de los cuadrados de las distancias de los puntos a dicha línea sujeta a la condición de que los puntos participantes en la suma cuando mucho tengan una distancia t de la linea. En otras palabras se tendrá que resolver dos problemas: clasificación de puntos en *outliers* e *inliers* (puntos válidos) y el ajuste de linea para los puntos validos. En su forma más simple se tomarán aleatoriamente un par de puntos y se calculará la linea que pasa por ellos; a partir de esta linea se calculan los *inliers* y *outliers* para esta linea (tomando como criterio que los *outliers* están a una distancia mayor de t unidades de la linea). Este proceso se repite un número de veces y la linea que consiga más número de *inliers* se considerará como la solución al problema.

El ejemplo anterior es el caso más simple ya que para construir el modelo de la linea solo se uso un par de puntos aleatorios, para hacerlo más complejo se puede seleccionar un subconjunto de puntos aleatorios para calcular el modelo de la linea en cada iteración, el calculo de modelos podría hacerse con un ajuste de mínimos cuadrados. Además se puede calcular un ultimo ajuste con todos los *inliers* finales (usando también mínimos cuadrados u otro método) para tener una mejor solución. Un algoritmo más general y robusto, tomando estas consideraciones es el siguiente:

- Se selecciona una muestra aleatoria s del conjunto de puntos S e inicializamos el modelo con este subconjunto. En el caso del modelo de una recta se puede usar un ajuste por mínimos cuadrados.
- Determinar el conjunto de puntos S_i que se encuentran a una distancia no mayor a t del modelo (*inliers*).
- Si el tamaño de *inliers* es mayor a un umbral T se vuelva a calcular el modelo con todos los puntos S_i y con esto se finaliza.
- Si es menor el número de *inliers* al umbral T, se vuelve a calcular la muestra aleatoria s y se empieza de nuevo.
- Para evitar que se cicle infinitamente el algoritmo después de N intentos se puede

considerar el conjunto S_j que tenga más *inliers* y se vuelve a calcular el modelo con los puntos S_j.

Capítulo 4

Experimentos

El este capítulo se muestran los experimentos y resultados obtenidos al aplicar la metodología descrita en el capítulo 3, primero se describen los materiales utilizados para hacer los experimentos y posteriormente los experimentos y resultados obtenidos en cada una de las partes descritas en la metodología.

4.1. Materiales

Para realizar los experimentos se utilizaron los siguientes materiales:

- Biblioteca de funciones *VisionLibs*: esta biblioteca fue desarrollada por el Dr. Arturo Espinosa Romero y se ha ido actualizando y mejorando con la colaboración de estudiantes de la Facultad de Matemáticas asociados a proyectos de investigación. *VisionLibs* se creo con el propósito de hacer un recopilado de funciones útiles en el procesamiento de imágenes y visión computacional, las clases desarrolladas en la presente tesis se anexan al conjunto de clases de la biblioteca.

- Cámaras fotográficas de la facultad de matemáticas: se utilizaron tres cámara fotográfica para capturar secuencias de fotografías de las escenas utilizadas para los experimentos.

  Dos de las cámaras son propiedad de la Facultad de Matemáticas, la primera es una cámara marca Sony modelo DFW-X710 con una resolución de 1024x768 pixeles y una velocidad de captura automática de 15 frames por segundo, la

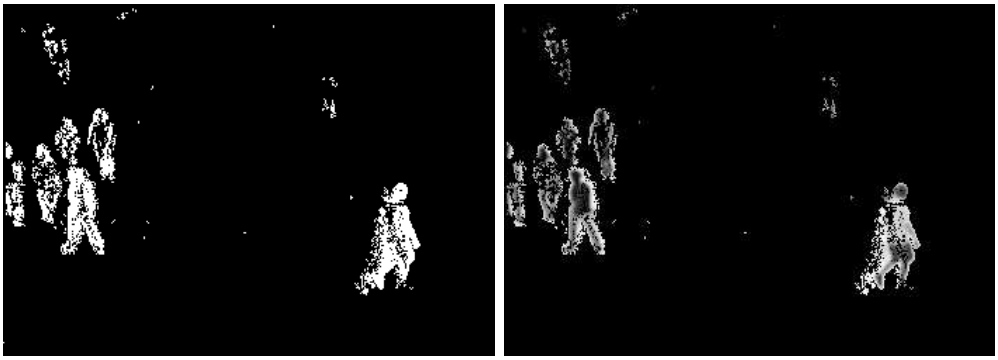
segunda es también de la marca Sony pero el modelo es el DFW-VL500 que tiene una resolución de 640x480 pixeles y una velocidad de captura automática de 30 frames por segundo. La tercera cámara utilizada es propiedad del Dr. Arturo Espinosa y es de la marca Sony Cybershot modelo DSC-F828 cuya resolución máxima es de 3264x2448 pixeles, esta última cámara no es automática sino que es manual.

- Cluster de Computadoras: Para el proceso de las secuencias de imágenes se utilizo un cluster de computadoras de la facultad de matemáticas disponible para investigación en el área de LICOVIR (Laboratorio de Instrumentación, COntrol, VIsión y Robótica). El cluster consta de cuatro computadoras cada una con dos procesadores de doble núcleo con una velocidad de 2 Ghz, dos gigabytes de memoria RAM y un disco duro de 250 gigabytes. Dicho cluster fue adquirido con apoyo del proyecto CONACyT SEP-2004-CO1-47893.

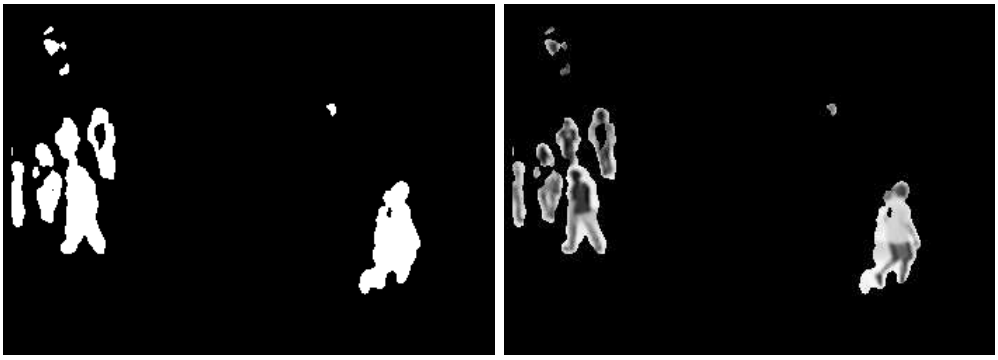
4.2. Adquisición de imágenes

Como se mencionó en la sección de materiales, se tomaron secuencias de imágenes para realizar experimentos, las secuencias fueron tomadas en la Facultad de Matemáticas con una velocidad de captura de aproximadamente ocho frames por segundo. Se seleccionaron escenas particulares para hacer los experimentos. Para seleccionar las secuencias de imágenes se tomaron en consideración los siguientes puntos:

- Que la escena contenga el piso o un plano paralelo a el. Pueden servir escenas que contengan varios planos paralelos.
- Sobre dichos planos hayan personas caminando puesto que son los objetos de análisis de la presente tesis.
- Que no existan tantas oclusiones entre las personas. Esto es debido a que los experimentos son para probar el objetivo de la tesis, la cual no comprende un manejo robusto de oclusiones.
- Que la cámara fotográfica este estática.



(a) Fondo sin filtrado Gaussiano (b) Fondo sin filtrado Gaussiano con texturas



(c) Fondo con filtrado Gaussiano (d) Fondo con filtrado Gaussiano y texturas

Figura 4.1: Resultados con y sin la aplicación de filtros Gaussianos en el proceso de extracción de sueldo

- Se escogieron posiciones estratégicas y una variación en la posición de la cámara de tal manera que tengamos resultados variados.

4.3. Extracción de Fondo

El primer paso de la metodología es la extracción de fondo (Véase el capítulo 3.1), sin embargo al hacer las primeras pruebas del algoritmo de extracción del fondo, se observó que usando el algoritmo se obtienen resultados con pixeles sin conexión y con muchos pequeños huecos en las regiones que no corresponden al fondo, por tal motivo se le añadió al algoritmo un filtrado Gaussiano utilizando una máscara Gaussiana con varianza igual a 2.0 pixeles (véase sección 2.3), los resultados con el filtrado y sin el filtrado se pueden observar en las imágenes de la figura 4.1.

Se efectuaron dos tipos de experimentos en el algoritmo de extracción de fondo.

Un experimento consistió en variar el número de Gaussianas en el algoritmo, el otro experimento fue el de variar los demás parámetros del algoritmo. El objetivo principal de estos experimentos fue el de probar el algoritmo y obtener los parámetros adecuados para el procesamiento de las imágenes.

4.3.1. Variación del número de Gaussianas

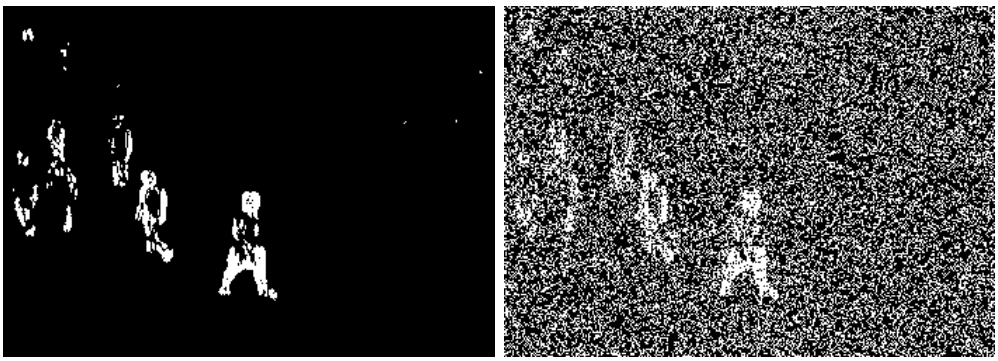
Un factor determinante en el desempeño del algoritmo de extracción de fondo es la elección adecuada del número de Gaussianas con el que se trabajará, este número está representado por la constante K en la ecuación 3.2. Si variamos el número de Gaussinas se pueden observar distintos resultados los cuales se aprecian en las figuras 4.2 y 4.3 y se explican a continuación.

En el experimento mostrado en la figura 4.2 se muestra el resultado de la extracción del fondo después de procesar las primera imagen de la secuencia, en la figura 4.2a se utilizó solamente una distribución Gaussiana para modelar el fondo mientras que en la figura 4.2b se utilizaron siete distribuciones Gaussianas. Se observa fácilmente que cuando se utiliza una Gaussiana rápidamente se obtienen resultados aceptables en las primeras iteraciones mientras que en el caso de muchas Gaussianas todavía no se tiene un modelo adecuado.

En el experimento mostrado en la figura 4.3 se muestra el resultado de la extracción del fondo después de procesar 45 imágenes, a estos resultados se les añadió la textura para una mejor presentación, en la figura 4.3a se utilizaron tres distribuciones Gaussianas para modelar el fondo mientras que en la figura 4.3b se utilizaron siete. Se observa con claridad que no hay mucha diferencia en los resultados puesto que no se necesitan tantas Gaussianas para modelar el fondo. Sin embargo el costo computacional se incrementa pues se requiere de más memoria y se procesan más matrices.

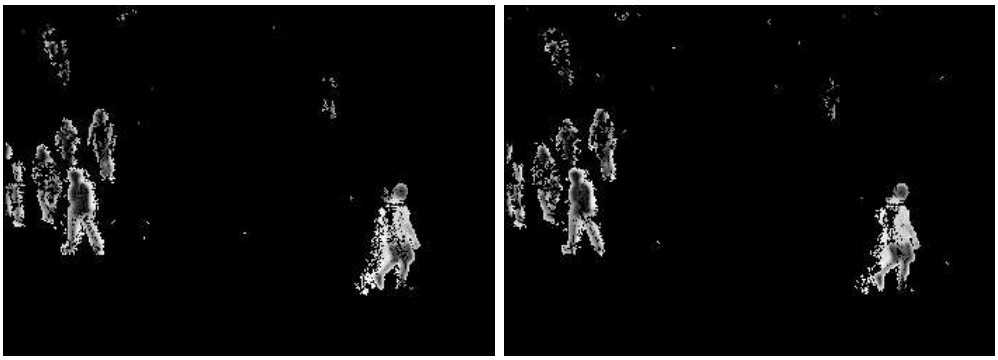
4.3.2. Variación de parámetros α y T

Para este experimento se analizaron los resultados al cambiar la constante de aprendizaje (α) y el parámetro que determina un umbral para que un pixel pertenezca al fondo (T) del algoritmo de extracción de fondo explicado en la sección 3.1, el número



(a) $K = 1$ (b) $K = 7$

Figura 4.2: Proceso de extracción de fondo utilizando varios valores para K en las primeras iteraciones



(a) $K = 3$ (b) $K = 7$

Figura 4.3: Proceso de extracción de fondo utilizando varios valores para K

de Gaussianas se mantuvo en tres.

Primero se cambió el parámetro α sin alterar los valores de los demás parámetros, esto arrojó resultados que teóricamente se esperaban: Si mantenemos fijos todos los parámetros y comparamos los resultados entre un valor α pequeño contra uno mayor, podemos apreciar que los resultados se diferencian en la parte de los pies de las personas y en la región contraria al movimiento de la persona.

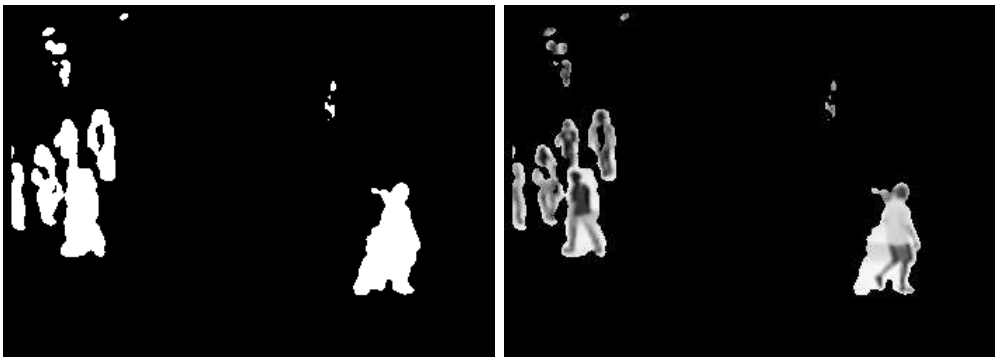
En las figura 4.4 se muestran resultados al ejecutar el algoritmo con distintos valores de α, se ejecutaron las pruebas con un valor constante de $T = 0.5$, y los valores α aplicados fueron 0.1, 0.3, 0.5 y 0.7. Si contrastamos el resultado de $\alpha = 0.1$ (Figura 4.4a)con el de $\alpha = 0.7$ (Figura 4.4d), se percibe claramente diferencias en la región de los pies de las personas y la región contraria a su movimiento las cuales se explicaran a continuación.

La diferencia entre los pies de las personas se explica fácilmente pues el factor α representa un factor de aprendizaje y si se tiene un mayor valor de este parámetro implica que en los pixeles que empiezan a dejar de moverse se conviertan en fondo rápidamente pues se le asigna un mayor peso (véase ecuación 3.7), así el pié pivote de las personas se definen mejor en valores con un α pequeño y con valores grandes este se trunca pues rápidamente se convierte en fondo (ver figura 4.4).

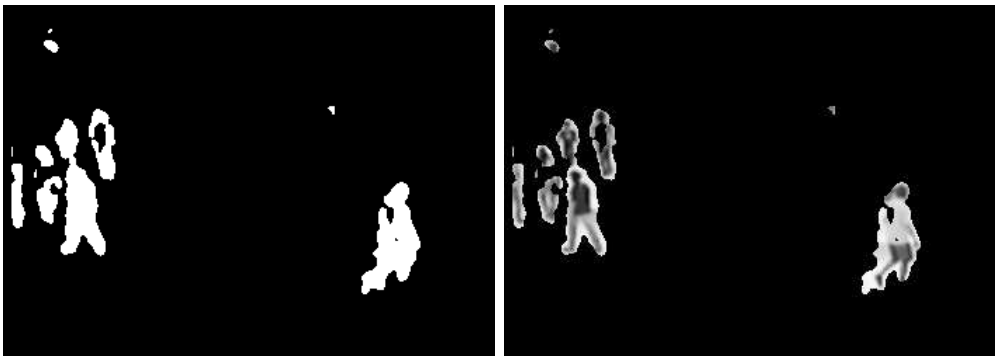
La región contraria al movimiento de la persona también tiene un efecto similar y es explicado por el mismo principio: los pixeles en los que la persona recientemente a pasado a lo largo de su trayectoria no se vuelven fondo de la imagen tan fácilmente con valores pequeños de α, mientras que con valores grandes de α casi no se observa este fenómeno (ver Figura 4.4).

Al parámetro T también se le asignaron distintos valores para probar cual nos afecta o nos perjudica: manteniendo un α constante con valor de 0.3, el parámetro T tomo los valores 0.1, 0.3, 0.5 y 0.7. Los resultados se aprecia en la figura 4.5.

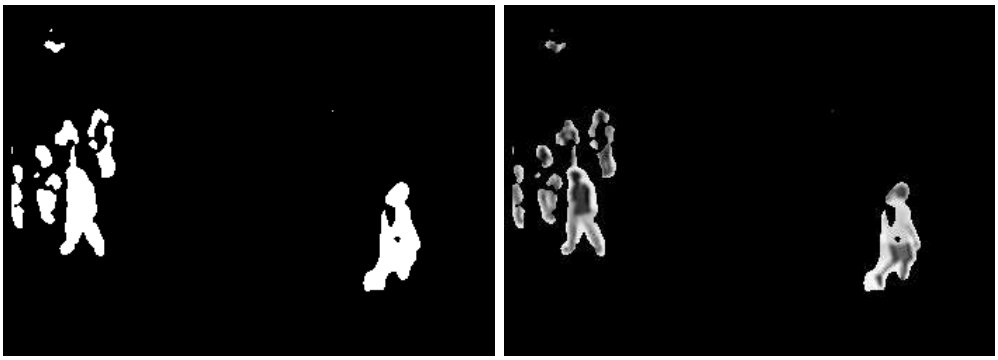
Al variar el parámetro T, se observa que al tener una T muy grande como la del valor 0.7, se toman muchos pixeles como fondo cuando no lo son, esto se explica fácilmente debido a que T representa una probabilidad acumulada mínima para seleccionar un conjunto de Gaussianas que nos modelen el fondo (ver formula 3.11) y al ser grande



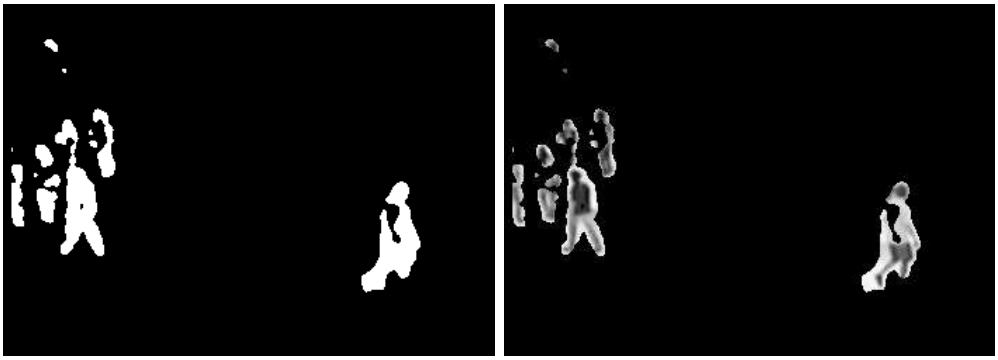
(a) $\alpha = 0.1$



(b) $\alpha = 0.3$



(c) $\alpha = 0.5$



(d) $\alpha = 0.7$

Figura 4.4: Proceso de extracción de fondo utilizando varios valores para α

Área Mínima	*Blobs* Encontrados	Regiones Eliminadas
10	1,2,3,4,5,8,9,10,13	11,12,14
30	1,2,3,4,5,8,9,10	11,12,13,14
50	1,2,3,4,5,8	9,10,11,12,13,14
100	1,2,3,4,5	8,9,10,11,12,13,14

Cuadro 4.1: *Blobs* encontrados en una escena

entonces tenemos mas probabilidad que incremente el numero de Gaussianas que modelen el fondo , incluyendo algunas Gaussianas que con otro parámetro mas pequeño de T no se incluyen, así más probabilidad de que un pixel pertenezca al fondo cuando pertenezca a una estas Gaussianas.

4.4. Extracción de *Blobs*

En esta sección se muestran los resultados obtenidos al extraer *blobs* después de extraer el fondo.

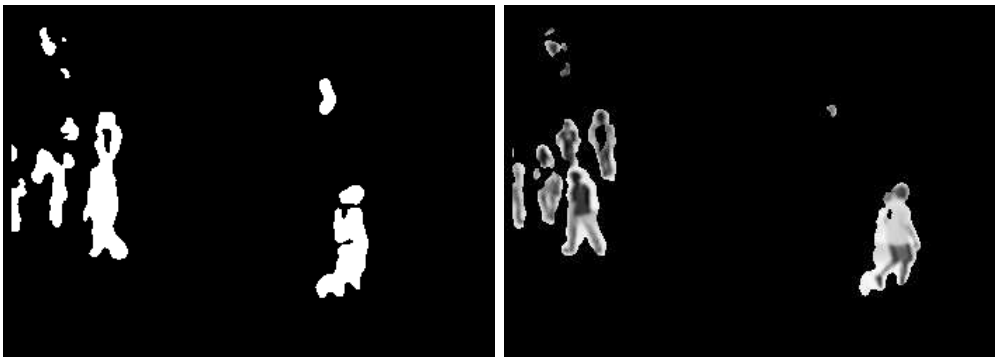
Un factor determinante para considerar que una región conectada se considere un *blob* es el área que ocupa, de esta manera un para que una región conectada pertenezca a un *blob* debe de cumplir con un área mínima.

En la taba 4.4 se muestran los resultados obtenidos al variar el área mínima que debe ocupar una región para considerarse un *blob*. En la primera columna se muestra el área, la segunda indica cuales *blobs* fueron encontrados haciendo referencia a las regiones marcadas en la figura 4.7.

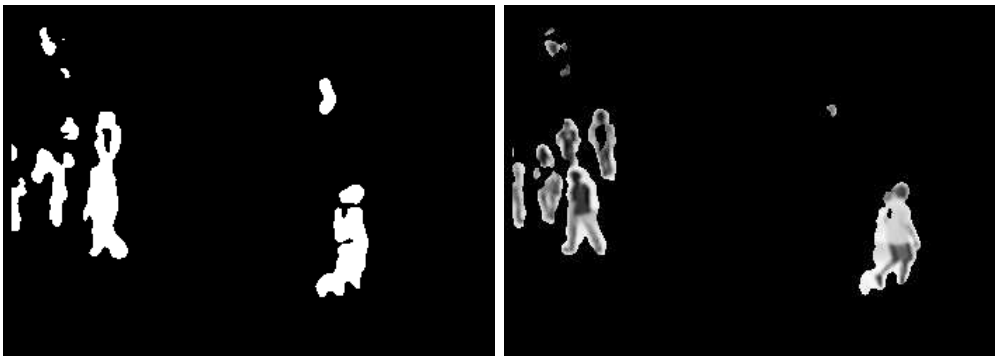
4.5. Características tridimensionales en la imagen

Como se mencionó en el capítulo 3, la principal característica tridimensional que vamos a obtener es la línea de fuga. Para lograrlo utilizamos las coordenadas que representan los pies de una persona y su cabeza, los cuales se encuentran en los *blobs* extraídos.

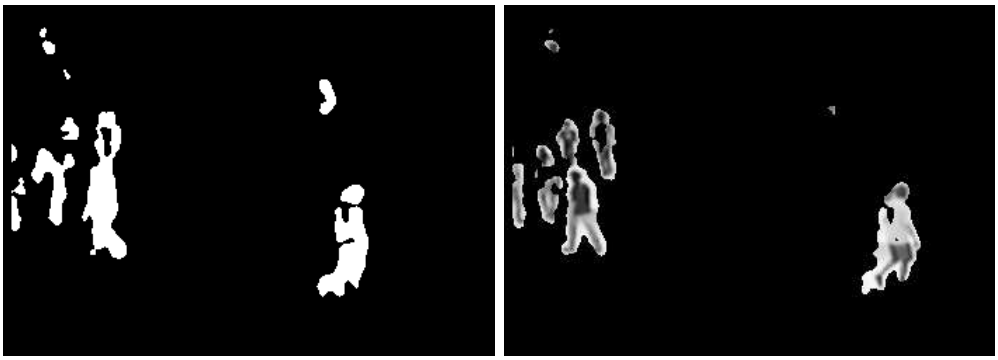
En la figura 4.8 se muestra la trayectoria obtenida al seguir los puntos que representan los pies y la cabeza de las personas. Se aprecia claramente que mientras se va alejando



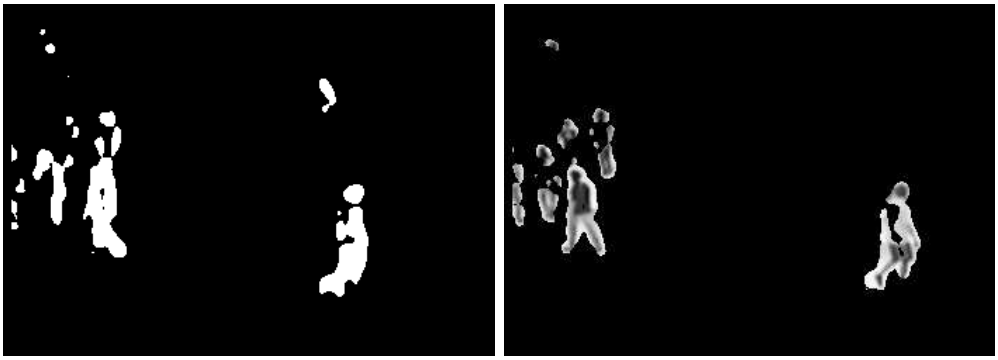
(a) $T = 0.1$



(b) $T = 0.3$



(c) $T = 0.5$



(d) $T = 0.7$

Figura 4.5: Proceso de extracción de fondo utilizando varios valores para T

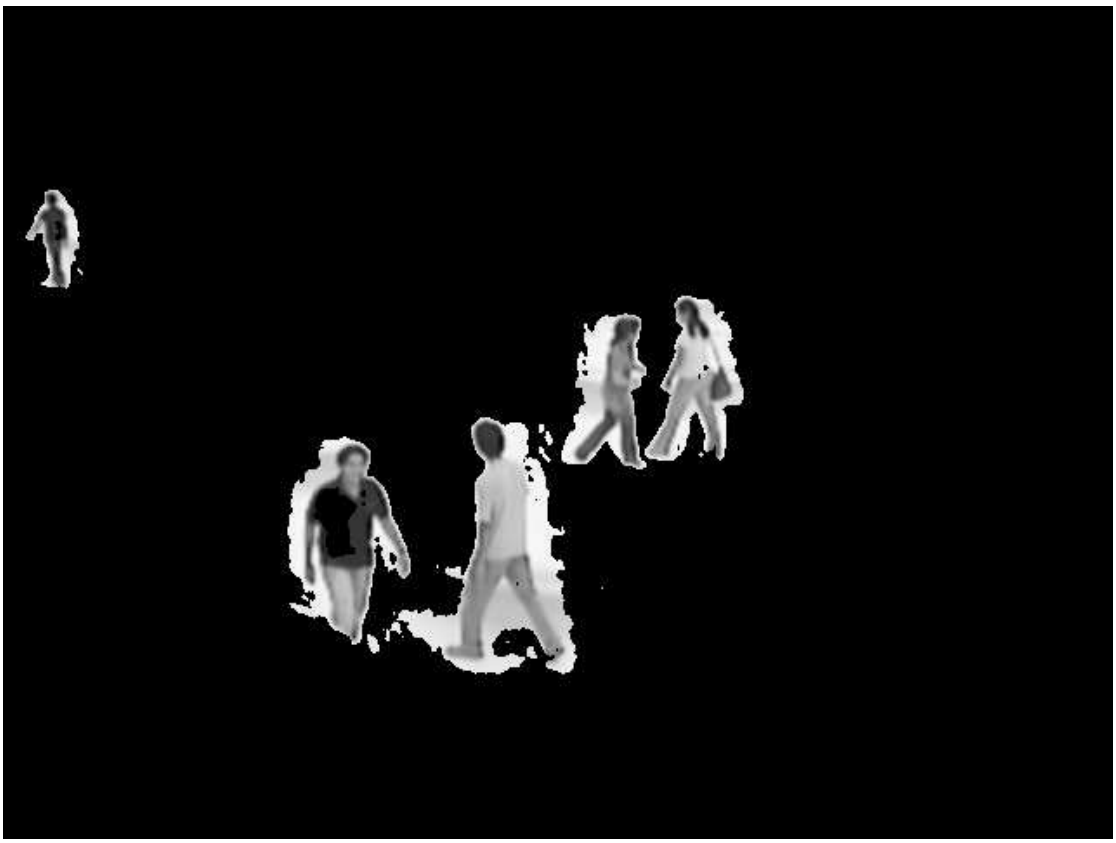


Figura 4.6: Resultado de la extracción del fondo con textura

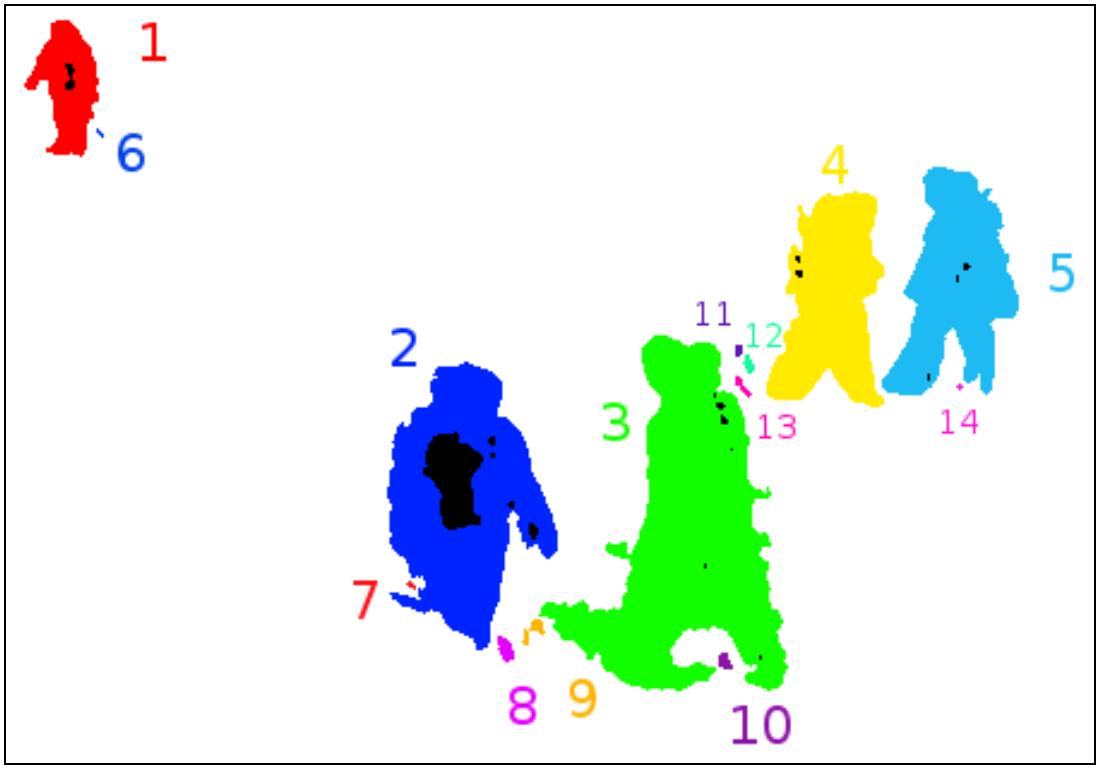


Figura 4.7: Regiones conectadas numeradas de la figura 4.6

de la cámara la distancia entre los dos puntos (cabeza y pie) se hacen cada vez menor.

En la figura 4.9 se aprecian los puntos de fuga calculados de la manera en que se describió en la metodología, claramente se aprecia mucho ruido pues los puntos de fuga teóricamente deben ser colineales, esto es debido a la propagación del error en las mediciones.

Para obtener la linea de fuga se utilizó el algoritmo de RANSAC descrito en la sección 3.4, la linea de fuga obtenida se aprecia en la figura 4.10. Se puede observar que nos da una muy buena aproximación a la linea de fuga de la imagen a pesar de que los puntos de fuga estén ruidosos. En la figura 4.11 se muestra la misma linea de fuga mostrando los inliers (puntos rojos) y outliers (azules) obtenidos con el algoritmo de RANSAC.

En las figuras 4.12 y 4.13, se muestran otros resultados del calculo de la línea de fuga los tienen una mayor precisión, esto es debido a que en lugar de trabajar con una imagen de 640x480 pixeles como se obtuvo el resultado mostrado en la figura 4.10 se trabajó con imágenes de 3264x2448 pixeles, de esta manera los errores en las mediciones son menos significativos.

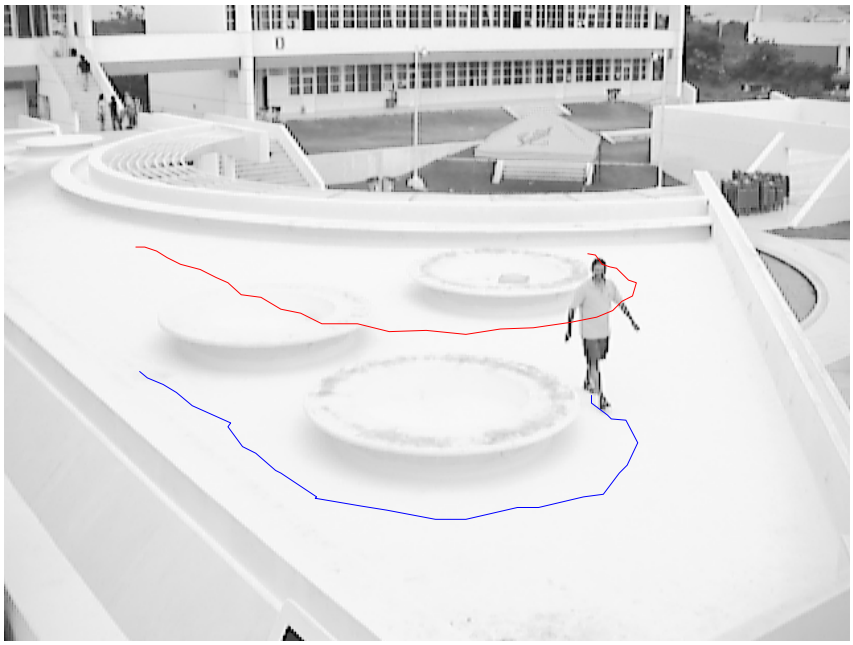


Figura 4.8: Trayectoria de una persona

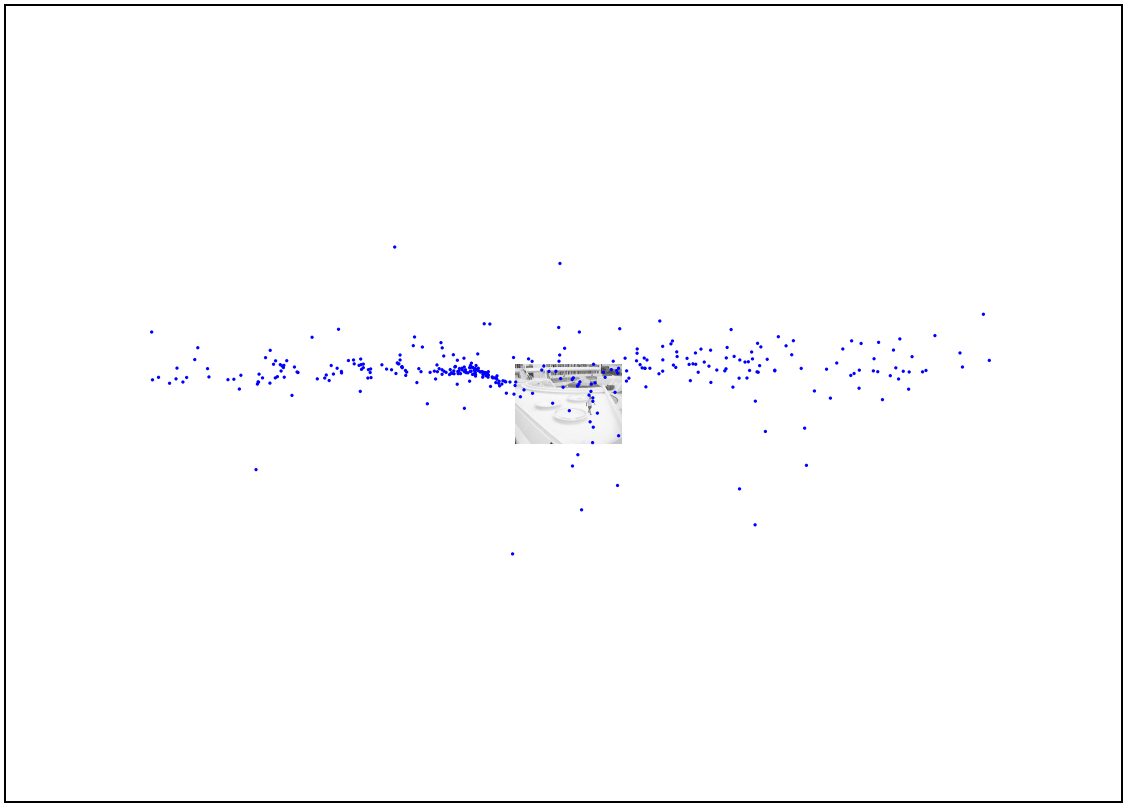


Figura 4.9: Puntos de fuga encontrados: en esta imagen se muestra la figura 4.8 con los punto de fuga calculados a partir de la trayectoria de la persona

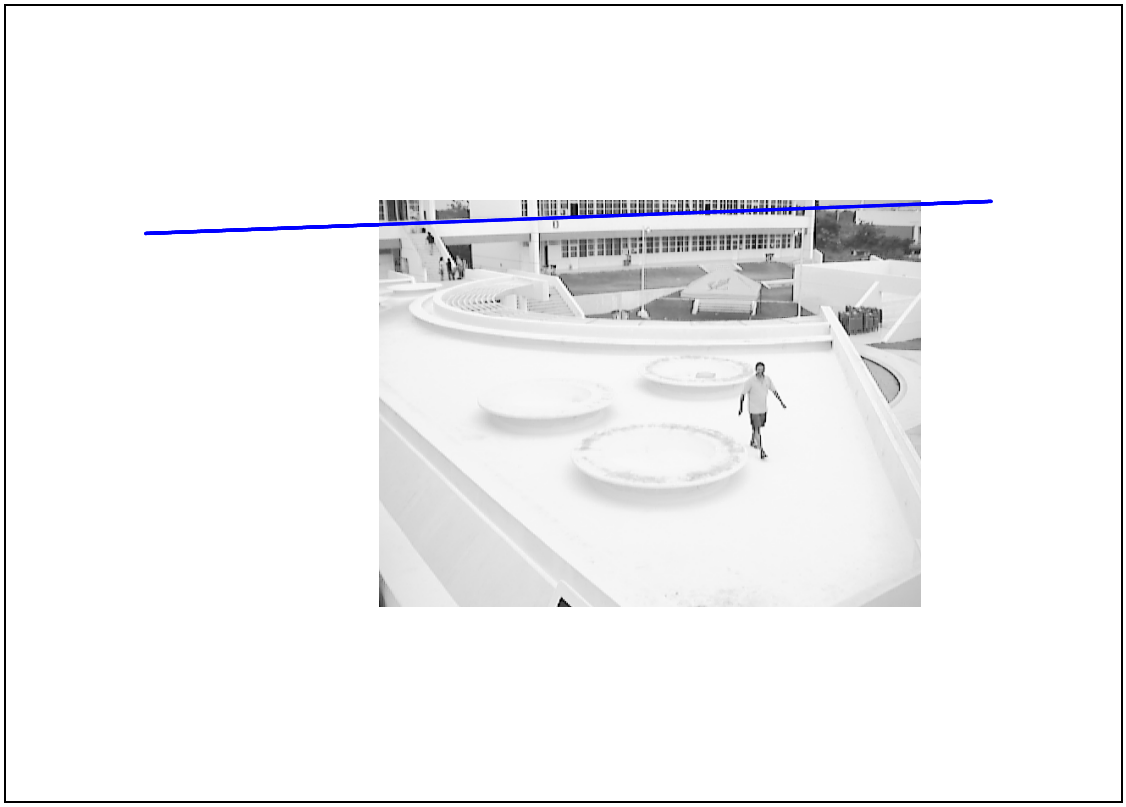


Figura 4.10: Linea de fuga calculada

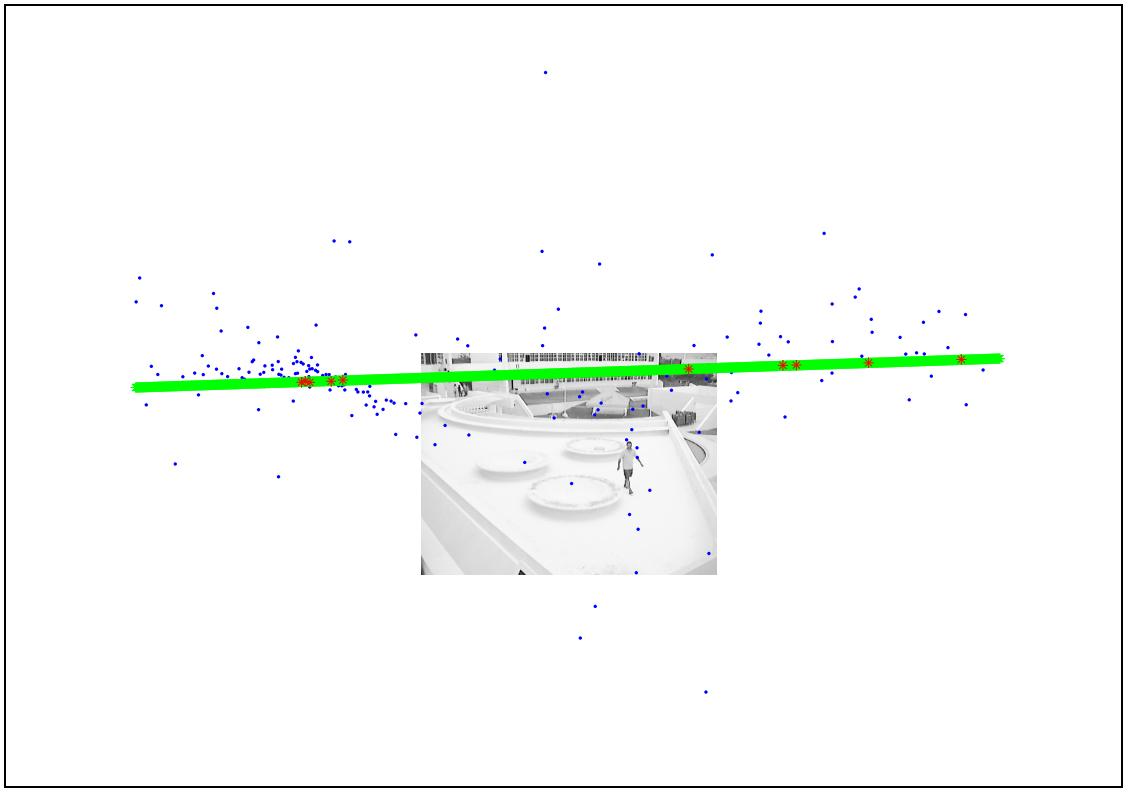


Figura 4.11: Linea de fuga calculada con inliers y outliers

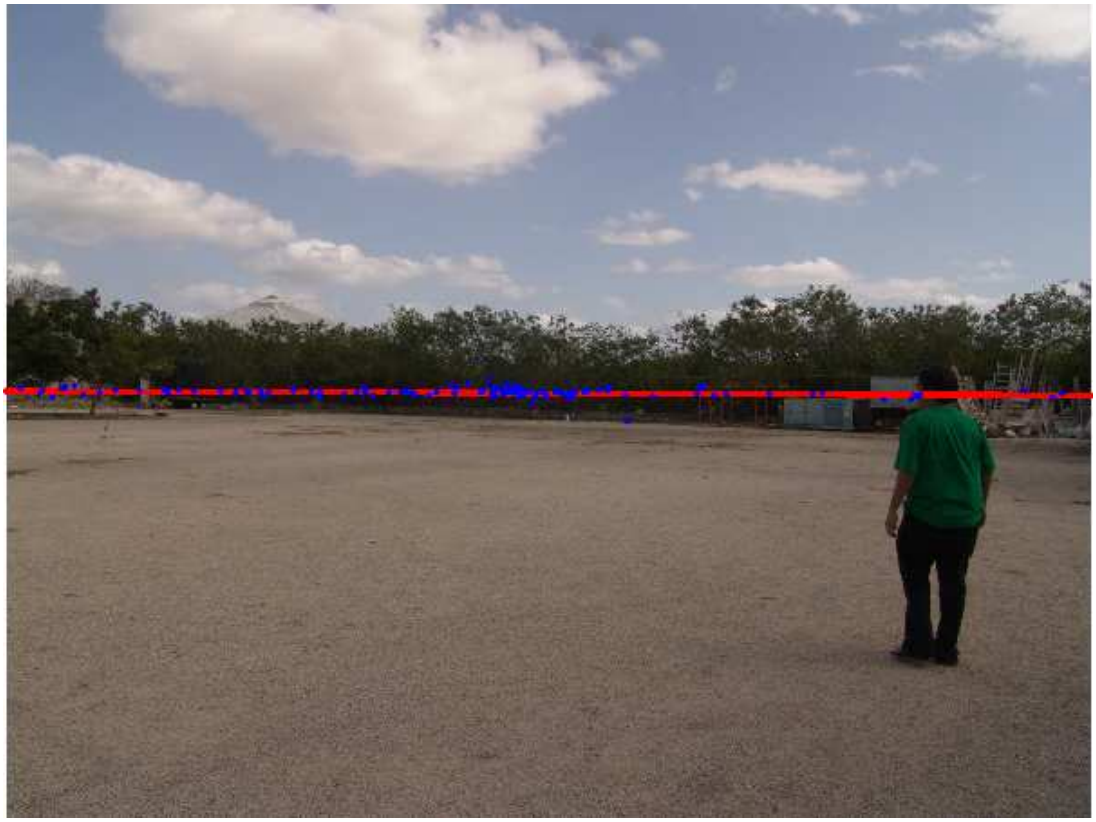


Figura 4.12: Linea de fuga calculada

Figura 4.13: Linea de fuga calculada

Capítulo 5

Conclusión

En este capítulo se presentan las conclusiones obtenidas al realizar la presente tesis, describiendo previamente el trabajo realizado y se proponen mejoras para trabajos futuros.

5.1. Trabajo Realizado

Para la elaboración de la presente tesis se realizo lo siguiente:

- Revisión bibliográfica:

  Se analizaron las distintas fuentes bibliográficas citadas en la presente tesis para evaluar trabajos similares realizados y escoger los métodos adecuados para la elaboración de la presente tesis.

- Extracción del fondo de la escena:

  Se implemento un algoritmo adaptivo para la extracción de fondo, de tal manera que sea robusto y soporte cambios de iluminación naturales presentados en la escena.

- Reconocimiento de personas en movimiento:

  Se implemento un método para reconocer personas en movimiento a partir del fondo extraído.

- Seguimiento de trayectorias:

Se implemento un mecanismo para almacenar dinámicamente las trayectorias de los cuerpos en movimiento de tal forma que se pueda utilizar en procesos posteriores de una manera fácil.

- Definición de las características tridimensionales en la imagen:

  Se establecieron referencias en la escena tridimensional para poder establecer características tridimensionales, estas referencias son los puntos y la linea de fuga, ya que a partir de ellos sabemos la orientación de la cámara utilizada para tomar imágenes y por las características de la geometría proyectiva sabemos que todos los planos paralelos al piso se interceptan en esta linea de fuga.

- Captura de secuencias de imágenes:

  Se tomaron fotografías de escenas que contengan personas caminando para realizar los experimentos.

- Evaluación de parámetros:

  Se evaluó el proceso de extracción de fondo y se analizaron los efectos de los parámetros del algoritmo para escoger aquellos valores que nos muestren mejores resultados.

- Evaluación de reconocimiento de personas y el seguimiento de trayectorias:

  Se estudió el proceso de extracción de personas caminando de la escena y se analizaron los efectos de sus parámetros para obtener mejores resultados en el seguimiento de sus trayectorias.

- Evaluación de la extracción de características tridimensionales:

  Se evaluó el proceso de obtención de los puntos y linea de fuga.

- Estimación robusta de la extracción de la linea de fuga:

  Se utilizó el algoritmo RANSAC para obtener la linea de fuga a partir de los puntos de fuga calculados de una manera robusta.

5.2. Conclusiones de los resultados

Después de analizar los resultados obtenidos en los experimentos de la presente tesis se obtuvieron las siguientes conclusiones:

- Es posible hacer la extracción de fondo de una manera robusta, y se encontró que con una adecuada selección de parámetros de los algoritmos nos arroja buenos resultados.
- Se pudo hacer un seguimiento de los objetos en movimiento, sin embargo al seguir una persona funciona bien si utilizamos su cabeza pero al seguir los pies genera mucha variación debido al rápido movimiento de los pies.
- Al seguir objetos con diferentes velocidades (por ejemplo escenas donde exista movimiento vehicular y de personas) se concluyo que es necesario hacer una adaptación al algoritmo de extracción de fondo para que sea más robusto ya que si se adaptan los parámetros para seguir a las personas el seguimiento de los autos no genera muy buenos resultados debido a que los factores de aprendizaje están configurados a la velocidad de la persona. Una solución podría ser utilizar distintos factores de aprendizaje.
- A pesar del ruido en los puntos de fuga se puede encontrar con gran exactitud la linea de fuga gracias a la robustez del algoritmo RANSAC que se utilizó.
- La linea de fuga es una buena característica tridimensional encontrada ya que de acuerdo a su pendiente y su posición con respecto al centro de la imagen podemos encontrar la orientación de la cámara con que fueron tomadas las fotografías.
- Al seguir la trayectoria de una persona podemos encontrar planos tridimensionales proyectados en la imagen, debido a la restricción de que la altura de una persona permanece constante.
- Debido a que se utiliza la restricción de que una persona camina sobre un plano, el trabajo de la presente tesis solo aplica en espacios arquitectónicos creados por el hombre, ya que si se aplica en espacios naturales es difícil que el terreno sea plano. Para poder utilizar este trabajo en espacios naturales se tendría que utilizar otra

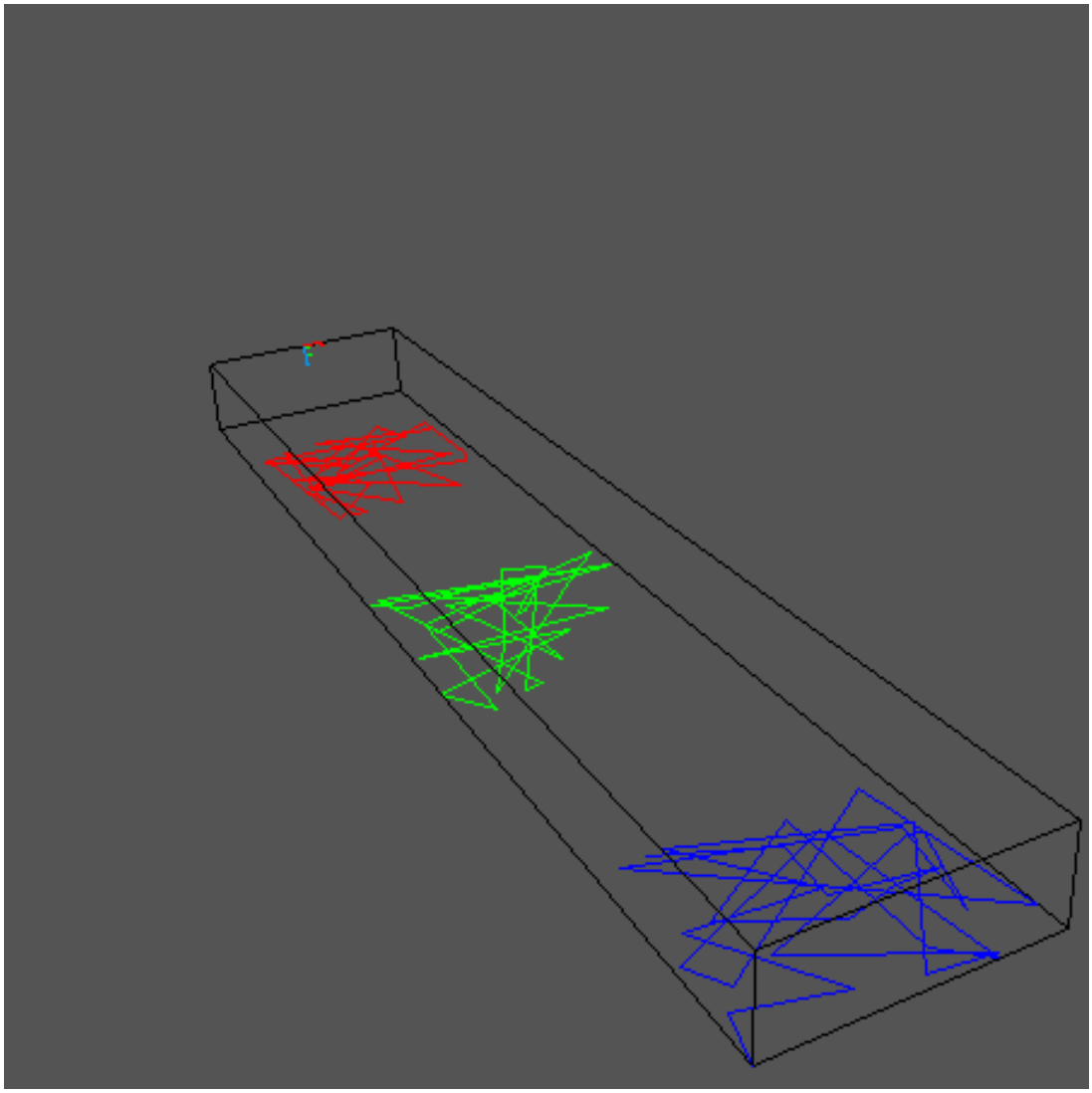


Figura 5.1: Variación de la dispersión de las trayectorias de los cuerpos

restricción, por ejemplo de que a pesar de que una persona caminando siempre camina erguida, es decir aunque este en una colina siempre conserva la vertical.

- Si observamos un grupo de personas caminando en un plano que en la imagen proyectada se encuentra cerca de la linea de fuga vemos que sus trayectorias se aglomeran mientras si observamos personas mas alejadas de la linea de fuga, sus trayectorias se observan mas dispersas, ver figura 5.1. Esto podría ser utilizado para obtener mas información tridimensional en la imagen.

5.3. Trabajo a futuro

El trabajo realizado en la presente tesis puede ser mejorado o ampliado en trabajos futuros, entre los trabajos propuestos se encuentran los siguientes:

Incluir en el algoritmo de extracción de fondo una solución para que funcione con objetos a diversas velocidades, la solución podría ser incluir varios factores de aprendizaje.

Mejorar el seguidor de personas añadiendo una solución a los problemas presentados al obtener los puntos de apoyo de los pies, pues debido al rápido movimiento de los pies existe mucho ruido. Una solución podría ser utilizar la información de la textura. Otra mejora podría ser incorporar detección de oclusiones y obstáculos, o utilizar modelos de siluetas de personas como las utilizadas por Adan Michael Baumberg (1995)

Obtener más información tridimensional de la escena y representarla en la imagen, como solución se propone utilizar los planos completos que describen las trayectorias de los cuerpos en movimiento para obtener lineas de fuga, y utilizar la observación de la dispersión de trayectorias conforme se acercan o se alejan de la linea de fuga. También se puede representar obstáculos y planos perpendiculares al piso.

Se puede utilizar el trabajo de esta tesis en otros problemas, principalmente en problemas de robótica e inteligencia artificial.

Bibliografía

Adan Michael Baumberg (1995). *Learning Deformable Models for Tracking Human Motion*. PhD thesis, University of Leeds.

Bruce, V., Green, P. R., and Georgeson, M. A. (1996). *Visual Perception: Physiology, Psychology, and Ecology*. Psychology Press, 3^{rd} edition.

Chirs Stauffer and W.E.L Grimsom (1999). Adaptive background mixture models for real-time traking. In *Proceedings of the Conference on Computer Vision and Pattern Recognition*, volume 2. IEEE Computer Society. ISBN: 0-7695-0149-4.

Christopher M. Bishop (1995). *Neural Networks for Pattern Recognition*. Oxford University Press.

David A. Forsyth and Jean Ponce (2003). *Computer Vision a Modern Approach*. Prentice Hall.

Hannah Dee and David Hogg (2004). Detecting inexplicable behavior. In *British Machine Vision Conference*, pages 477–486.

Milan Sonka, Vaclav Hlavac, and Royer Boyle (1993). *Image Processing, Analysis and Machine Vision*. Chapman & Hall.

Pascual J. Figueroa, Neucimar J. Leite, and Ricardo M.L Barros (2006). Tracking soccer players aiming their kinematical motion analysis. *Computer Vision and Image Understanding*, 101:122–135.

Rafael C. Gonzalez and Richard E. Woods (2002). *Digital Image Processing*. Prentice Hall, 2^{nd} edition.

Richard Hartley and Andrew Zisserman (2000). *Multiple View Geometry in Computer Vision*. Cambridge University Press.