Benemérita Universidad Autónoma de Puebla
Facultad de Ciencias de la Computación

SEGUIMIENTO Y RECONOCIMIENTO DE COMPORTAMIENTOS DE PERSONAS USANDO MODELOS OCULTOS DE MARKOV

Presenta
Julio César Pérez Sansalvador

Tesis sometida como
requisito para obtener el título de
Licenciado en Ciencias de la Computación

Asesor
Dra. Janeth Cruz Enríquez
Laboratorio de Visión
INAOE

Coasesor
Dr. Manuel Martín Ortiz
Facultad de Ciencias de la Computación
BUAP

Puebla, Pue. Marzo 2008

Resumen

La automatización de video-sistemas de vigilancia es un área importante de investigación para el sector comercial y privado, hoy en día es muy común encontrar cámaras de video-vigilancia en centros comerciales, escuelas, instituciones de gobierno, etc, en donde la mayoria de los eventos capturados por las cámaras son grabados en cintas, discos o algún otro medio de almacenamiento, estos son comúnmente archivados o sobrescritos y sólo en caso que ocurriese algún delito son analizados para obtener información de los posibles delincuentes, pero por supuesto para entonces es demasiado tarde.

Para evitar este problema se requiere un monitoreo constante, lo más común y económico es contratar personal el cual se encargue de realizar esta supervisión. Es probable que este personal no sea capaz de observar y captar todos los eventos debido a la cantidad de cámaras a su cargo, es por eso que existe la necesidad de mejorar y automatizar este proceso con el fin de lograr un monitoreo constante y solo en caso que se detecten actividades maliciosas, sospechosas o de emergencia se informe con alertas a las personas indicadas para que se actúe a tiempo.

Una solución es un sistema que informe de eventos sospechosos o de emergencia, una de las tareas importantes para lograr esta meta es enseñar al sistema como identificar estos comportamientos, aquí es donde toman lugar los métodos de aprendizaje para la interpretación de las secuencias de video; dos de los enfoques importantes y muy usados para este tipo de tareas son: Redes neuronales y Modelos gráficos probabilísticos.

En este trabajo se utilizará uno de los últimos mencionados debido al enfoque probabilístico, ya que este ofrece ventajas al manejar la incertidumbre de la tarea del procesamiento de video, segmentación incompleta, clasificación e interpretación.

Dos herramientas importantes dentro de este enfoque son las “Redes bayesianas” y los “Modelos ocultos de Markov”, estos últimos se han utilizado recientemente con resultados satisfactorios en el reconocimiento e interpretación de secuencias de video debido a que describen las relaciones probabilísticas entre atributos y actividades.

Agradecimientos

A mi mamá
Por ser mi mamá, por cuidarme, quererme, escucharme, ..., por todo lo que una mamá hace, te quiero mucho ma.

A mi papá
Por ser mi papá, gracias pa por tu cariño, cuidado y consejo, por enseñarnos que la vida es diversión , te quiero mucho pa.

A mi hermana
Por enseñarme que la vida es puro relajo y diversión, gracias por lavar mi ropa, jeje.

A mis abuelitos
Gracias a ustedes me tocaron esos padres tan buena onda que tengo, en especial a mi abuelita Eva, ya acabe abue!.

A Ket-zi
Porque me quieres y te quiero mucho corazón de melón, gracias por la clave final para terminar este trabajo.

A mi familia
Por el apoyo incondicional que siempre me ha dado, gracias a todos.

A mi asesora de tesis Janeth
Porque a pesar de tanto trabajo que tienes, siempre tuviste tiempo para mis dudas.

A mis amigos
Por enseñarme que no existen cosas inalcanzables.

Al Laboratorio de Visión del INAOE porque me brindaron el equipo necesario para realizar esta tesis, con ello fue posible evaluar los resultados con ambientes reales.

Tabla de Contenido

Índice de Figuras

Índice de Tablas

Lista de Algoritmos

Capítulo 1

Introducción

En este capítulo se da una breve introducción a la visión por computadora y se plantea el problema a resolver, en la ultima parte se específica la organización de la tesis.

1.1. Antecedentes y Motivación

1.1.1. Visión por computadora

El sistema visual de percepción humano es capaz de adquirir, procesar e interpretar toda la información visual a nuestro alrededor, al transmitir esas capacidades a una computadora para la toma de decisiones es necesario, primero entender los métodos de almacenamiento, procesamiento, clasificación y finalmente la interpretación de esos datos, es aquí donde la visión por computadora toma un papel importante.

La visión por computadora puede definirse como la disciplina encargada de la obtención de información por medio de algoritmos de análisis y manipulación de imágenes para la interpretación de estas y la toma de decisiones, además busca aplicar sus teorías en el desarrollo de sistemas de visión artificial.

Un sistema de visión por computadora esta formado básicamente de uno o varios sensores (cámaras) para la captura de imágenes y una unidad de procesamiento que se encarga de la obtención de información de los sensores y la manipulación de esta para la toma de decisiones.

La unidad de procesamiento se encuentra comúnmente constituida por las siguientes etapas.

- Captura

- Filtrado
- Segmentación
- Extracción de características
- Clasificación y análisis
- Interpretación de la imagen

En algunas ocasiones las imágenes capturadas por las cámaras presentan alteraciones, esto es conocido comúnmente como "ruido", este puede ser causado por diversos factores como aberración de la lente, elementos del propio escenario, es decir, nubes, lluvia y el movimiento de las ramas de los árboles producido por el viento, baja calidad de la imagen, etc.

En el caso específico de la detección de movimiento con cámara fija, si se produce algún movimiento de la cámara al momento de la captura de las imágenes, el desempeño del método de detección se verá afectado por estos movimientos.

Para todos estos casos es necesario un pre-procesamiento de las imágenes el cual consiste en la aplicación de técnicas que mejoren la calidad visual de la imagen y logren eliminar el ruido.

Una vez eliminado el ruido de las imágenes es común aplicar técnicas para segmentar la imagen, las técnicas de segmentación consisten en dividir la imagen en regiones homogéneas, cada píxel de la imagen es asignado a una región la cuál es etiquetada para un análisis posterior. La segmentación es una de las tareas más importantes en el análisis automático de imágenes ya que con este proceso se extraen los objetos de interés de la imagen para después obtener información de ellos en el proceso de análisis.

Después de aplicar la segmentación se extraen algunas características que permiten diferenciar entre las regiones, las más comunes se basan en el análisis de la forma, color o textura de la región.

Finalmente, tomando como base las características de cada región, se asigna una clase a cada una de estas regiones, para ello se utilizan diversos métodos de clasificación como árboles, basados en reglas, redes neuronales, redes bayesianas, etc, al obtener la clasificación de las regiones se puede dar una interpretación a la imagen.

1.1.2. Aplicaciones de la visión por computadora

Con el crecimiento de las técnicas de análisis de imágenes, también crece la demanda de las aplicaciones para la visión por computadora, estas van desde la automatización de la inspección visual de piezas, pasando por la interpretación de imágenes biomédicas hasta la integración en sistemas de exploración para el espacio.

Algunos ejemplos son:

- Sistemas automáticos de inspección visual para control de calidad en la manufactura de piezas.
- Interpretación de imágenes tomadas desde sensores remotos.
- Análisis de imágenes médicas.
- Sistemas de vigilancia.
- Sistemas automáticos de clasificación.
- Compresión y restauración de imágenes.
- Identificación de personas(huella dactilar, iris).

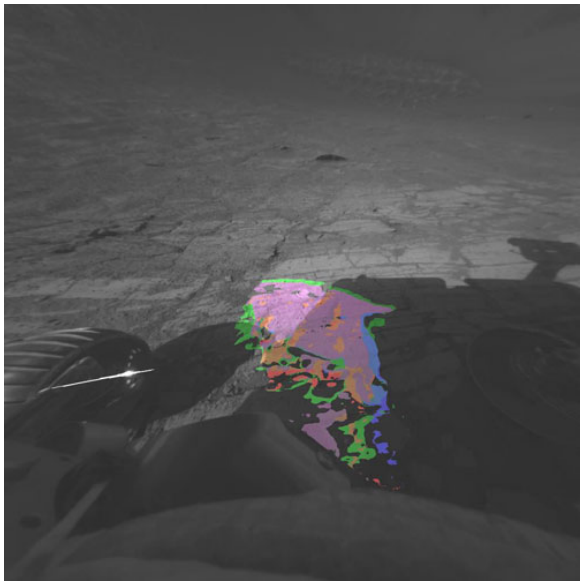

Figura 1.1: Mapa creado con el software Multimission Image Processing Lab (MIPL) en la NASA

En la figura 1.1 se muestra una aplicación de la visión por computadora, en este caso se presenta una imagen donde las regiones en color se encuentran al alcance del brazo del robot, estas distancias son calculadas haciendo uso de la visión estéreo.

1.2. Descripción del Problema

Hoy en día es muy común encontrar sistemas de vigilancia en escuelas, centros comerciales, bancos, casas, etc, que operan con una gran número de cámaras, toda la información obtenida por estos sensores es enviada al área de vigilancia donde se almacena en cintas, discos o algún otro medio, estos son archivados o sobrescritos, y solo en el caso de acontecer una actividad anormal o de emergencia tales como robo, peleas, etc., son revisadas para obtener mayor información de los responsables. Este método resulta ineficiente ya que es necesario actuar de inmediato en caso de algún delito o accidente.

Otra solución es contratar personas que se dediquen al monitoreo de estas cámaras, el problema en este caso es el gran número de datos o cámaras a revisar, el resultado son algunos eventos no identificados.

Es por eso que se propone una solución basada en visión por computadora la cuál realizará un monitoreo constante y sólo en caso de detectar actividades maliciosas o de emergencia se informará con alertas.

Una de las tareas importantes es la detección y clasificación de objetos en movimiento la cuál es un área importante de investigación en visión por computadora. El problema principal en esta tarea es la aplicación de técnicas de detección de movimiento en ambientes externos, ya que en un mundo dinámico como en el que vivimos, un gran número de objetos en movimiento deben ser clasificados como objetos que no son de interés, un buen ejemplo son las ramas de los árboles que se mueven por el viento.

Las tareas de segmentación y seguimiento de objetos es uno de los retos más grandes en visión por computadora ya que en este último se tiene gran dependencia al tipo de ambiente y de los objetos a seguir para la aplicación de los algoritmos utilizados para esta tareas.

Para llevar a cabo estas tareas es necesario desarrollar un conjunto de algoritmos que puedan integrarse a un sistema de vigilancia utilizado en ambientes externos, para este trabajo de tesis será un estacionamiento. El conjunto de estos algoritmos debe ser capaz de detectar objetos en movimiento, seguirlos, obtener sus características y clasificarlos como personas o autos para luego identificar sus comportamientos,

estos pueden ser: personas conversando, robo a personas (delincuencia) o pelea entre personas.

1.3. Objetivo

El reconocer el comportamiento de los humanos en diversos ambientes desde secuencias de video es un problema difícil de resolver debido a la naturaleza temporal de los comportamientos humanos. El objetivo general de este trabajo es:

- Desarrollo de un sistema para reconocimiento de algunos comportamientos de personas por medio del procesamiento y análisis de secuencias de imágenes capturadas mediante una cámara fija.

Objetivos Específicos

- Desarrollo de un algoritmo de detección de movimiento robusto a ambientes externos, es decir, detectar como ruido cambios en la intensidad de luz, movimiento de objetos causado por el viento.
- Implementación de un algoritmo para el seguimiento de los objetos en movimiento, este algoritmo se encargará de asignar etiquetas a los objetos detectados para lograr identificarlos a través del tiempo.
- Integración de un algoritmo de clasificación de los objetos en movimiento. Las clases a considerar son:
  - Personas
  - Automóviles
- Obtención de los modelos utilizados por el método "Modelos Ocultos de Markov.[ei]ntegración del algoritmo para la interpretación de las secuencias de video.

1.4. Contribuciones

Un sistema de video-vigilancia para exteriores que reconozca de manera eficiente algunos comportamientos de personas, y por medio de este tome decisiones para emitir alertas de seguridad.

El sistema hará uso de varios algoritmos como la detección de movimiento, seguimiento de objetos, clasificación y reconocimiento de comportamientos con ayuda de los "Modelos Ocultos de Markov", estos algoritmos se pueden tomar como base o ayuda para el desarrollo de otras aplicaciones.

1.5. Organización de la Tesis

En este capítulo se ha hablado de la visión por computadora y sus diferentes aplicaciones, además de plantear el problema a resolver en este trabajo, la parte complementaria de esta tesis se organiza de la siguiente manera.

En el capítulo dos se mencionan los conceptos básicos para la solución del problema planteado, se presenta el estado del arte en el desarrollo de sistemas de vigilancia, inicialmente se mencionarán los algoritmos de detección de movimiento y las características que se deben cumplir para la aplicación de estos métodos, después se describen los métodos de seguimiento y clasificación que se utilizarán para este trabajo y las restricciones de estos métodos, finalmente se explicará el funcionamiento de los "Modelos Ocultos de Markov"para reconocimiento de comportamientos.

En el capítulo tres se muestra la propuesta para la solución del problema presentado, este método se encuentra dividido en dos módulos principales, el módulo de visión y el de interpretación, en las secciones de este capítulo se explica con detalle cada uno de ellos.

En el capítulo cuatro se muestran los resultados obtenidos por cada uno de los módulos que conforman el sistema además de presentar los resultados de la conjunción de estos.

En el capítulo cinco se presentan las conclusiones de este trabajo y el trabajo futuro como algunas mejoras a los diferentes módulos que conforman el sistema.

Capítulo 2

Fundamentos

En este capítulo se mencionan los conceptos básicos, estos serán clave para entender mejor la solución propuesta y ayudarán a comprender algunos trabajos desarrollados en el área, en la parte final de este capítulo se presenta un análisis del estado del arte, el cuál cubre temas relacionados con este trabajo de tesis.

2.1. Introducción

El reconocimiento de comportamientos dentro de secuencias de video es una tarea difícil, esta entre otras cosas comprende el uso de varias técnicas de visión por computadora para lograr la interpretación de la secuencia de imágenes o de video.

2.2. Conceptos básicos

Para la mayoría de los sistemas que hacen uso de procesamiento de imágenes es conveniente caracterizar la imagen en forma matemática.

Una imagen pueden ser representada como una matriz A de n x m, donde:

- n = altura de la imagen.
- m = ancho de la imagen.

cada uno de los elementos de la matriz $A(x, y)$ representa un píxel de la imagen.

Cada uno de estos píxeles está asociado a un valor numérico que representa el valor de intensidad de luz de ese punto dentro de la imagen, estos valores se encuentran en el rango $[0, 255]$ donde 0 representa cero intensidad (oscuro), y el 255 la máxima

intensidad (claro), el uso de este rango de valores se debe a que comúnmente se utiliza un byte (8 bits) para el almacenamiento de cada píxel.

Una imagen digital en escala de grises o monocromática se representa por una sola matriz A, mientras que para una imagen digital a color se utilizan tres matrices: R, G, B, cada una representa una banda espectral(o canal) (R = Red, G = Green, B = Blue), a diferencia de la imagen en escala de grises, la imagen a color requiere de la combinación de las tres bandas para representar el color de un píxel.

Representación de un píxel negro en una imagen en escala de grises:

$$A(x, y) = 0 \tag{2.2.1}$$

Representación de un píxel negro en una imagen a color:

$$A(x, y) = (0, 0, 0) \tag{2.2.2}$$

2.2.1. Operaciones morfológicas

La *erosión* y la *dilatación* son las dos operaciones morfológicas básicas, la *morfología* se refiere al estudio de las formas y de la estructura.

La morfología matemática emplea la teoría de conjuntos para representar las formas de los objetos en una imagen. De este modo, las operaciones morfológicas se pueden describir simplemente añadiendo o eliminando píxeles de una imagen binaria.

Desde el punto de vista de visión por computadora se denomina *dilatación* al crecimiento de una región después de aplicar alguna máscara. La *erosión* es el proceso de aplicar algún tipo de máscara a una imagen con el fin de eliminar información que se encuentre aislada de posibles regiones.

Para poder definir la *dilatación* y la *erosión* es necesario recordar algunas operaciones básicas.

Definición 2.2.1. Sea A un conjunto con elementos en $\mathbb{Z}^2$, la *traslación* de A por un punto $x \in \mathbb{Z}^2$ se define como:

$$(A)_x = \{c | c = a + x, \forall a \in A\} \tag{2.2.3}$$

Definición 2.2.2. Sea A un conjunto con elementos en $\mathbb{Z}^2$, la *reflexión* de A se define como:

$$\hat{A} = \{x | x = -a, \forall a \in A\} \tag{2.2.4}$$

Haciendo uso de estas definiciones se enuncia la *dilatación* y la *erosión*.

Dilatación

Definición 2.2.3. Sean A y B dos conjuntos con elementos en $\mathbb{Z}^2$, la *dilatación* entre estos conjuntos se define como:

$$A \oplus B = \{p | p = a + b, a \in A \text{ y } b \in B\} \tag{2.2.5}$$
$$A \oplus B = \bigcup_{b \in B} (A)_b \tag{2.2.6}$$

Las expresiones 2.2.5 y 2.2.6 que definen a la dilatación son equivalentes.

Ejemplo de dilatación

$$A = \{(1,0),(1,1),(1,2),(2,2),(0,3),(0,4)\} \tag{2.2.7}$$
$$B = \{(0,0),(1,0)\} \tag{2.2.8}$$
$$A \oplus B = \{(0,1),(1,1),(1,2),(2,2),(0,3),(0,4), \tag{2.2.9}$$
$$(2,0),(2,1),(2,2),(3,2),(1,3),(1,4)\} \tag{2.2.10}$$

La figura 2.1 muestra gráficamente este ejemplo.

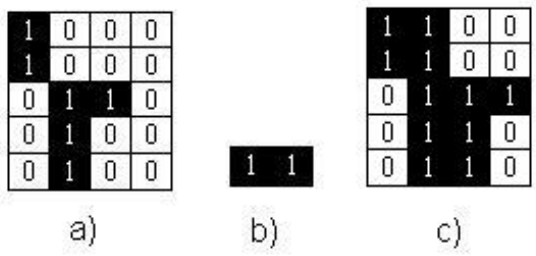

Figura 2.1: Para el ejemplo, el conjunto A se representa en inciso (a), el conjunto B se representa en el inciso (b) y $A \oplus B$ se representa en el inciso (c).

Erosión

Definición 2.2.4. Sean A y B dos conjuntos con elementos en $\mathbb{Z}^2$, la *erosión* entre estos conjuntos se define como:

$$A \ominus B = \{p | p - b \in A, \forall b \in \hat{B}\} \tag{2.2.11}$$

$$A \ominus B = \bigcap_{b \in \hat{B}} A_b \tag{2.2.12}$$

Al igual que en la dilatación, las expresiones 2.2.11 y 2.2.12 son equivalentes y definen a la erosión.

Ejemplo de erosión

$$A = \{(1,0),(1,1),(1,2),(0,3),(1,3),(2,3),(3,3),(1,4)\} \tag{2.2.13}$$

$$B = \{(0,0),(1,0)\} \tag{2.2.14}$$

$$A \ominus B = \{(0,3),(1,3),(2,3)\} \tag{2.2.15}$$

La figura 2.2 muestra gráficamente este ejemplo.

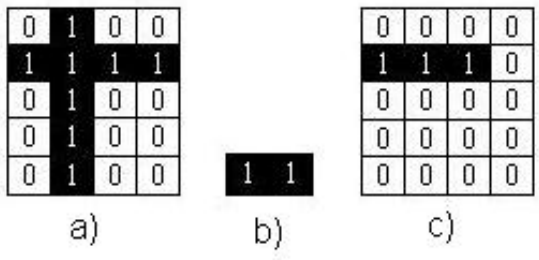

Figura 2.2: Para el ejemplo, el conjunto A se representa en inciso (a), el conjunto B se representa en el inciso (b) y $A \ominus B$ se representa en el inciso (c).

Apertura - Cerradura

Definición 2.2.5. La operación de *apertura* puede definirse como una composición de las funciones *erosión* y *dilatación*, si $A \oplus B$ representa a la operación de *dilatación* con el conjunto A y B, y $A \ominus B$ representan a la operación de *erosión* de los conjuntos A y B, entonces la *apertura* $A \circ B$ es:

$$A \circ B = (A \ominus B) \oplus B \tag{2.2.16}$$

Dado que la composición de funciones no es conmutativa, a la función de aplicar una *dilatación* y después una *erosión* se le conoce como operación de *cerradura*, esta se define para los conjuntos A y B como sigue:

$$A \bullet B = (A \oplus B) \ominus B \tag{2.2.17}$$

2.2.2. Detección de movimiento

La detección de movimiento en secuencias de video es conocida como una tarea difícil y un buen problema de investigación [1]. Primero es necesario separar el fondo de la imagen de los objetos de interés de una escena dinámica, este problema se vuelve más complejo debido al ambiente de trabajo. En este caso, un ambiente exterior, donde diversos factores como la iluminación, variaciones en la luz del sol, movimiento de objetos causado por el viento pueden llevar a confusiones al momento de la detección de objetos en movimiento.

A través de un algoritmo de segmentación se obtiene el fondo de la imagen y las regiones en movimiento también conocidas como "blobs", estos se vuelven el foco de atención para los algoritmos de seguimiento y reconocimiento de comportamientos. Con esta técnica se logra un mejor procesamiento al considerar sólo los píxeles en movimiento ya que se reduce considerablemente el espacio de trabajo.

Existen varias técnicas utilizadas para la detección de objetos en movimiento, tres de las más importantes son:

- Diferencia temporal
- Extracción del fondo adaptable
- Análisis del flujo óptico

La diferencia temporal consiste en realizar una resta absoluta entre una imagen en el tiempo t con otra en el tiempo $t-1$ y si el valor obtenido excede un umbral U, se marca como píxel en movimiento.

Definición 2.2.6. Sea la matriz M la que contiene la información de los píxeles en movimiento de la matriz A. Entonces:

$$M(x,y) = \begin{cases} 1 & \text{si } |A_t(x,y) - A_{t-1}(x,y)| > U \\ 0 & \text{si } |A_t(x,y) - A_{t-1}(x,y)| \leq U \end{cases} \tag{2.2.18}$$

Esta técnica es buena por su rápida adaptación a ambientes dinámicos, como en el caso de escenas de exteriores, el problema de esta técnica es que no obtiene todo el objeto en movimiento. Ver 2.3 (a).

El método de extracción de fondo, llamaremos fondo a la parte estática de la escena que no cambia con el tiempo, consiste en restar la imagen A en el tiempo t del fondo B, si el valor obtenido sobrepasa un umbral U se considera como píxel en movimiento.

Definición 2.2.7. Sea B el fondo extraído de la matriz A, entonces M que es la matriz de movimiento de A se define como:

$$M(x,y) = \begin{cases} 1 & \text{si } |A_t(x,y) - B(x,y)| > U \\ 0 & \text{si } |A_t(x,y) - B(x,y)| \leq U \end{cases} \tag{2.2.19}$$

La extracción del fondo consigue una mayor parte del objeto en movimiento pero es muy sensible a factores presentes en ambientes dinámicos como son los cambios de iluminación. Ver 2.3 (b).

Figura 2.3: Problemas en la detección de movimiento (a) Diferencia de imágenes (b) Extracción de fondo

El método del flujo óptico es usado en sistemas con cámara en movimiento, este es considerado bueno, pero la mayoría de los sistemas basados en él necesitan de sistemas con tiempos de respuesta muy pequeños, tiempo real, o de hardware dedicado.

El método de fondo adaptable es frecuentemente utilizado en ambientes externos debido a algunas ventajas que ofrece bajo ciertas condiciones como:

- Cambios de iluminación debido a nubes, niebla, lluvias, etc.

- Cambios en el fondo por movimiento de objetos debido a fenómenos naturales como viento o tormentas.
- Las luces artificiales como lámparas, al ser encendidas o apagadas modifican la iluminación de la escena y por tanto el fondo.
- Las sombras son también elementos importantes ya que pueden producir problemas al ser detectadas como parte del objeto en movimiento.

Todos estos factores producen cambios en el fondo el cual no es una entidad estática, es necesario el uso de un fondo adaptable cuando se trabaja con escenas de ambientes dinámicos. Si no son tomados en consideración los puntos mencionados y no se adapta continuamente el modelo del fondo, los errores en el modelo del fondo se irán acumulando y provocarán fallas en el método de detección de movimiento y en consecuencia se obtendrán malos resultados en los algoritmos posteriores (seguimiento, clasificación e interpretación).

El fondo adaptable B se obtiene re-calculando los valores de B en las zonas donde hay movimiento.

Definición 2.2.8. Sea M la matriz de movimiento de la imagen A, entonces el fondo adaptable B queda definido por:

$$B_{t+1}(x,y) = \begin{cases} \alpha B_t + (1-\alpha)A_t(x,y) & \text{si } M_t(x,y) = 0 \\ B_t & \text{si } M_t(x,y) = 1 \end{cases} \tag{2.2.20}$$

Donde α es una constante en el tiempo, con valores en el rango $[0,1]$, e indica que tan rápido es actualizada la información.

2.2.3. Seguimiento de objetos

Una vez obtenidas las regiones de cada uno de los objetos en movimiento, el siguiente paso consiste en eliminar los objetos que no son de interés y mantener en observación a los objetos restantes, este proceso se realiza mediante el seguimiento de estos objetos mientras se mantengan en escena.

El seguimiento de objetos mejor conocido en inglés como "tracking" es el proceso de localizar uno o varios objetos en movimiento en una secuencia de video. En el área de visión por computadora el seguimiento de objetos es utilizado en varias aplicaciones como:

- Sistemas de vigilancia.

- Monitoreo de tráfico.
- Robótica.

El problema principal del seguimiento de objetos es la asociación de objetos detectados en un conjunto de imágenes consecutivas, este conjunto conforma una secuencia de video.

Un caso particular en el seguimiento de objetos es cuando los objetos se mueven demasiado rápido, en relación a la frecuencia de muestreo de la cámara, ya que al no presentarse en secciones continuas a donde se encontraban anteriormente, es difícil lograr un seguimiento adecuado.

El seguimiento de múltiples objetos comprende dos grandes tareas:

1. Representación y localización del objetivo.
2. Filtrado y asociación de datos. Algunos ejemplos son:
   - Filtrado
     - Filtro Kalman.
     - Filtro Kalman extendido.
     - Filtro Particula.
   - Asociación
     - Vecinos cercanos (Nearest Neighbor).
     - PDA (Probabilistic Data Association).
     - JPDA (Joint Probabilistic Data Association).

Los métodos de seguimiento pueden ser clasificados en 4 categorías, estas son:

1. Métodos basados en correlación de ventanas. Esta es de las técnicas más sencillas y más rápidas, es utilizada en diversas aplicaciones de visión por computadora, la idea es buscar el objeto en seguimiento dentro de un región delimitada por una ventana de búsqueda, y donde se tenga la mejor medida de correlación es en donde se localizará el objeto en seguimiento.
2. Métodos basados en minimización de energías locales. Con frecuencia los objetos se mueven siguiendo movimientos elementales como traslaciones, rotaciones, zoom, etc, los métodos que utilizan técnicas basadas en estas características se encuentran en esta clase.

3. Métodos basados en minimización de energías globales. Estos métodos calculan los movimientos en todos los puntos de la imagen, este tipo de métodos imponen condiciones de "regularidad" de tal manera que el movimiento de un punto no sea "demasiado" diferente al de sus vecinos.

4. Métodos que utilizan filtros Kalman. Cuando se conoce a "priori" el movimiento realizado por un objeto en escena es posible combinar esta información con los cálculos reales del movimiento del objeto, con ello se logrará mejorar la estimación de la trayectoria del objeto en seguimiento.

La eliminación de falsas alarmas se realiza con la ayuda de información como la cinemática de los objetos, si al ser analizada se encuentra que el objeto cambia de dirección constante y rápidamente se puede suponer que se trata de una *falsa alarma*, algunos objetos que cumplen con esta condición son las ramas de los árboles. Con el uso de esta información es posible pre-clasificar algunos objetos.

Uno de los métodos más sencillos para el seguimiento de objetos consiste en obtener una medida de similitud entre las regiones cercanas al objeto y el candidato con la mejor medida será el representante del objeto en la escena siguiente. Esta técnica es rápida, pero no es suficiente para resolver algunos problemas que se presentan en el seguimiento de objetos tales como la oclusión por objetos estáticos, es por ello que en ocasiones se hace uso de filtros de estimación o de un área de búsqueda más grande que resulta en mayor tiempo de procesamiento.

Una de las características importantes de los métodos para seguimiento de objetos es que deben ser computacionalmente rápidos para lograr su ejecución en tiempo real.

Algunos conceptos importantes utilizados en el seguimiento de objetos se presentan a continuación.

Para obtener el conjunto de regiones cercanas a otra se hace uso de las posiciones de cada región, cada una se encuentra limitada por un cuadro mínimo envolvente con coordenadas $(x1, y1), (x2, y2)$. Todas las regiones poseen un centro, ha este se le da el nombre de *centroide*, informalmente el centroide o baricentro es el promedio de los puntos que forman el cuadro mínimo envolvente.

Definición 2.2.9. Sean $(x1, y1), (x2, y2)$ las coordenadas del cuadro mínimo envolvente de la región R, entonces el centroide de esa región es $C(R) = (cx, cy)$, el cual será el punto central, los elementos cx y cy se obtiene de la siguiente manera:

$$cx = \frac{x1 + x2}{2} \qquad cy = \frac{y1 + y2}{2} \tag{2.2.21}$$

Definición 2.2.10. Sea RC el conjunto de las regiones cercanas a la región de interés R, entonces R_i es una región cercana si el centroide $C(R_i)$ o punto medio de la región se encuentra dentro del cuadro mínimo envolvente de R.

$$RC = \{R_i | x1 \leq cx_i \leq x2 \text{ y } y1 \leq cy_i \leq y2\} \tag{2.2.22}$$

Dentro del conjunto de las regiones cercanas solamente se toma una, esta debe cumplir con la mejor medida de similitud con el objeto en cuestión.

Definición 2.2.11. Sea $P(R_t) = (px, py)$ la posición de la región de interés en la imagen A en el tiempo t, entonces la nueva posición $P(R_{t+1})$ estará dada por la región R_i del conjunto RC, con la mejor medida de similitud $C(R_i) = c_i$.

$$P(R_{t+1}) = \{P(R_i) | max\{C(R_i)\}, R_i \in RC\} \tag{2.2.23}$$

2.2.4. Clasificación de objetos

Al terminar la fase de seguimiento, es posible conocer el tipo de objeto que se encuentra en escena, este proceso se conoce como clasificación de objetos, esto ayudará para que posteriormente, en la etapa de reconocimiento de comportamientos sea posible conocer la interacción entre objetos.

La inteligencia artificial se puede definir como la ciencia que se encarga de la creación de métodos o técnicas para lograr imitar el comportamiento y razonamiento humano.

Uno de los objetivos específicos de la inteligencia artificial es el desarrollo de métodos que permiten a las computadoras aprender, es decir, el desarrollo de programas que logren generalizar comportamientos con base en un conjunto de datos suministrados en forma de ejemplos (clasificación de datos para la toma de decisiones).

Dentro del ámbito de la inteligencia artificial(aprendizaje automático), clasificar se refiere a la generación de clases, grupos de elementos con características similares, que son utilizadas para la interpretación de los datos.

Definición 2.2.12. Una clase es un conjunto de elementos con características similares

Sea la clase C el conjunto de los elementos x_i que cumplen la propiedad P.

$$C = \{x_i | P(x_i = \text{Verdadero})\} \tag{2.2.24}$$

La clasificación se divide, por la forma de generar estas clases y desde el punto de vista de aprendizaje, en dos grandes campos:

- **Clasificación supervisada:** Se parte de un conjunto de elementos, con determinado número de variables o características, y de un conjunto de clases, además se conoce con certeza la pertenencia de cada elemento a las clases, la meta en este enfoque es encontrar una función o un patrón que identifique a cada clase, esta función se encarga de asignar una clase a los elementos a clasificar.
- **Clasificación no supervisada:** Se tiene un conjunto de elementos y se conoce la cantidad de grupos o clases que se desea generar, este tipo de técnica *agrupa* estos elementos conforme a sus características o variables.

Para este trabajo, se puede cuestionar el porqué del desarrollo de un método para clasificación de objetos, o mejor dicho, ¿porqué realizar un algoritmo de clasificación si ya se cuenta con un método adecuado para ello? (las personas logran clasificar mejor que las computadoras), algunas razones son:

- Al tener una persona o varias personas encargadas de clasificar, es posible obtener errores debido al agotamiento natural de las personas.
- La clasificación se vuelve más rápida y sólo se toma en cuenta la información suministrada, y el caso de las personas pueden ser persuadidas por información irrelevante.
- Un procedimiento mecánico de clasificación ayuda a las personas a concentrarse en otros problemas que suelen ser difíciles de resolver por una computadora.

Interpretación de la clasificación

Uno de los modos más sencillos para entender la clasificación cuando se basa en medidas cuantitativas es pensar en el conjunto de variables que caracterizan a los elementos y a las clases como un conjunto de ejes que definen un espacio de variables multidimensionales.

En la figura 2.4 se puede observar que el número de variables o características es igual al número de ejes en el espacio. Las clases son formadas por un conjunto de datos agrupados en cierta posición en el espacio.

La definición de las clases se puede lograr de dos formas:

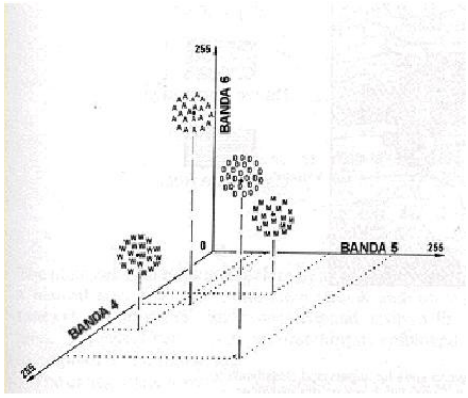

Figura 2.4: Interpretación de la clasificación en un espacio multidimensional.

1. Se divide el espacio de variables en regiones mediante el trazo de fronteras, las regiones resultantes son las clases obtenidas. Este método es utilizado comúnmente en algunos métodos de clasificación como los árboles de decisión y las redes neuronales.

2. A cada clase se le asigna un centroide, se obtiene una medida de asociación que será utilizada para establecer el grado de asociación entre los datos y la correspondencia con las clases. Este enfoque es, por ejemplo, utilizado por el método de mínima distancia y máxima probabilidad. Ver figura 2.5.

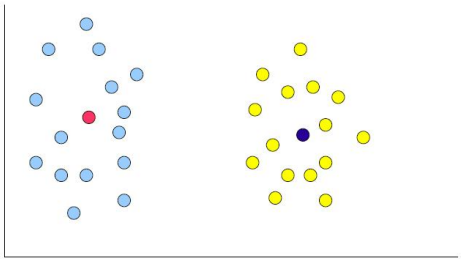

Figura 2.5: Centroide o elemento característico de la clase.

Dos de las medidas de asociación más utilizadas son la de similitud y disimilaridad, para el caso de la medida de similitud se calcula con base a la cantidad de características similares entre los elementos, entre más coincidencias existan, mayor será esta medida. Esta medida cumple con las siguientes propiedades.

Definición 2.2.13. Sea $Sim(D_i, D_j)$ la medida de similitud entre los elementos D_i y D_j, entonces las siguientes reglas normalmente se cumplen.

$$\text{(a)} \quad 0 \leq Sim(D_i, D_j) \leq 1 \tag{2.2.25}$$
$$\text{(b)} \quad Sim(D_i, D_i) = 1 \tag{2.2.26}$$
$$\text{(c)} \quad Sim(D_i, D_j) = Sim(D_j, D_i) \tag{2.2.27}$$

Algunas de las medidas de similitud más comunes son:

- Coeficiente de Dice
$$Sim(D_i, D_j) = \frac{2|D_i \cap D_j|}{|D_i + D_j|} \tag{2.2.28}$$
- Coeficiente de Jaccard
$$Sim(D_i, D_j) = \frac{D_i \cap D_j}{|D_i \cup D_j|} \tag{2.2.29}$$
- Distancia euclidiana
Si $D_1 = (x_1, y_1, ..., z_1)$ y $D_2 = (x_2, y_2, ..., z_2)$
$$Sim(D_i, D_j) = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2 + ... + (z_1 - z_2)^2} \tag{2.2.30}$$

Ejemplos de clasificadores

Existe un gran número de clasificadores, algunos suelen ser sencillos y otros demasiado complejos, la selección de estos clasificadores depende de las características que presentan los datos a clasificar.

Algunos ejemplos de clasificadores:

- Clasificación supervisada
  - Algoritmos de votación
  - Árboles de decisión (DT)
  - Algoritmos basados en reglas
  - K vecinos mas cercanos (K-NN)
  - Redes neuronales
  - Naïve Bayes (NB)

- Support Vector Machine (SVM)

- Clasificación no supervisada
  - K-Means
  - Algoritmo de pasada única
  - Métodos jerárquicos

Máquinas de vectores de soporte (SVM)

En este trabajo ha sido seleccionado el algoritmo de clasificación SVM, por lo que es necesario conocer su funcionamiento.

Las máquinas de vectores de soporte o SVM (por sus siglas en inglés) desarrolladas por Vapnik forman parte de los algoritmos utilizados en el aprendizaje automático, su sistema de aprendizaje se basa en el uso de un espacio de hipótesis de funciones lineales las cuales serán llevadas a un espacio de mayor dimensión, inducido por un Kernel, en caso de no lograr representar al conjunto de datos de interés.

Las máquinas de vectores de soporte buscan el *hiperplano* que generaliza mejor, el cual tiene mayor margen de separación entre las clases.

Las máquinas de vectores de soporte han resultado ser eficientes, tanto que para clasificación como para regresión se han encontrado muchas aplicaciones como clasificación de imágenes, reconocimiento de caracteres, detección de proteínas, clasificación de patrones, identificación de funciones, etc.

El kernel. Debido a las limitaciones computacionales de las máquinas de aprendizaje lineal estas no pueden ser utilizadas en la mayoría de las aplicaciones del mundo real. La representación por medio del Kernel ofrece una solución alternativa a este problema, proyectando la información a un espacio de características de mayor dimensión el cual aumenta la capacidad computacional de las máquinas de aprendizaje lineal. La forma más común en que las máquinas de aprendizaje lineales aprenden una función objetivo es cambiando la representación de la función, esto es similar a mapear el espacio de entradas X a un nuevo espacio de características N.

$$N = \{\phi(x)|x \in X\} \tag{2.2.31}$$

En la figura 2.6 se muestra un ejemplo de mapeo de valores al espacio de características, esto se debe a que en el primer espacio los datos no pueden ser

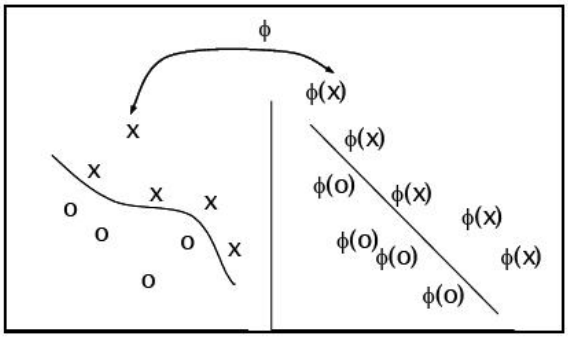

Figura 2.6: Mapeo de valores al espacio de características por medio de la función $\phi(x)$.

separados por una máquina lineal, y después de aplicarse la función $\phi(x)$ los datos son separados fácilmente.

Definición 2.2.14. Un kernel es una función, donde $\forall x, z \in X$

$$K(x,z) = \langle \phi(x) \cdot \phi(z) \rangle = \sum_{i=1}^{l} \phi_i^T(x)\phi_i(z) \tag{2.2.32}$$

La función Kernel debe satisfacer algunas condiciones, estas son descritas a detalle en [2], algunos de los Kernels que satisfacen esas propiedades son los siguientes:

Lineal. Este Kernel es una transformación lineal del tipo

$$K(x,z) = \langle Ax \cdot Az \rangle = x^T A^T A z = x^T B z \tag{2.2.33}$$

donde $B = A^T A$ es una matriz semidefinida positiva.

Polinomial. El mapeo polinomial es un método muy usado para modelar funciones no lineales

$$K(x,x) = (\langle x,x \rangle)^d \tag{2.2.34}$$
$$K(x,x) = (\langle x,x \rangle + c)^d \tag{2.2.35}$$

para el segundo caso $c \in R$. En la práctica es común utilizar el segundo kernel.

Funciones de base radial. Las funciones de base radial (RBF, por sus siglas en inglés) son también conocidas como funciones Gaussianas.

$$K(x,z) = \exp(\frac{-\|x-z\|^2}{\sigma^2}) \tag{2.2.36}$$

Estos son algunos de los kernels utilizados con mayor frecuencia con las máquinas de vectores de soporte.

El tema de las máquinas de vectores de soporte es demasiado extenso, si desea profundizar en el tema puede consultar [3] donde se explica a detalle el funcionamiento de estas.

2.2.5. Reconocimiento de comportamientos

Una vez que se ha identificado el tipo o los tipos de objetos que se encuentran en escena, es posible identificar si existe interacción entre ellos o que acciones realiza cada uno de ellos. Uno de los grandes problemas en visión por computadora e inteligencia artificial es la interpretación de las secuencias de video, el enfoque de investigación en esta área se enfoca en el desarrollo de métodos para el análisis visual que sean capaces de obtener y procesar información del ambiente que rodea a los objetos en escena [4].

Hablando del tema de los comportamientos humanos, es sabido que existe un gran número de ellos tales como: platicar, caminar, correr, robar, etc, y la capacidad de reconocerlos depende en parte del tipo de dispositivos utilizados para la adquisición de información, algunos comportamientos no solo son perceptibles por medio de dispositivos de visión, y por lo tanto no es posible reconocerlos todos; en este trabajo como ya se ha mencionado, el medio de adquisición de datos es una cámara.

Gran parte de las investigaciones se ocupa en el desarrollo de métodos para ambientes específicos, por lo que resulta difícil la reutilización de estos métodos en otros ambientes.

Debido a la gran variedad de comportamientos y de escenarios posibles, los investigadores han optado por dividir el problema de la comprensión automática de video en dos pasos:

1. Un módulo de visión encargado de obtener características relevantes de la escena y de eventos primitivos.

2. La información obtenida es utilizada por otro módulo que realiza una detección más compleja y el reconocimiento de patrones de comportamientos.

Al dividir el problema, es posible utilizar el conocimiento *a priori* de la escena en cada paso. El primero de ellos, por lo general hace uso de métodos estocásticos para el análisis de los datos, mientras que el segundo realiza un análisis estructural de los símbolos obtenidos.

Un comportamiento puede ser visto como una secuencia de acciones en el tiempo, [5] por ejemplo, el comportamiento "*Desayunar*"implica: (1) Ir a la cocina, (2) Abrir el refrigerador para tomar los alimentos (3) Cocinar los alimentos (4) Ir al comedor a desayunar (5) Encender la T.V., ver 2.7.

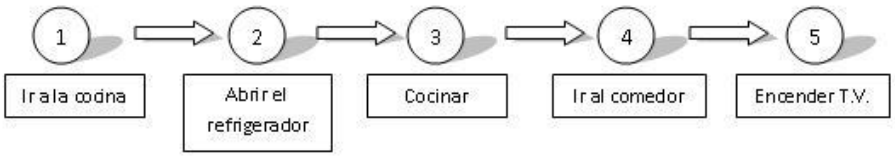

Figura 2.7: Secuencia de acciones para realizar el comportamiento *Desayunar*

Como se logra observar en la figura 2.7, un comportamiento es un conjunto de acciones en serie, por lo que se puede pensar inmediatamente en el uso de una máquina de estados para este modelo. Es posible entonces, hacer uso de algún tipo de modelo determinístico para el reconocimiento de comportamientos, pero como es sabido, al trabajar con secuencias de video es preferible el uso de métodos probabilísticos para la toma de decisiones, para este trabajo se hace uso de los Modelos Ocultos de Markov ya que en trabajos recientes como [6], [7] y [8] entre otros han demostrado buenos resultados, además de cumplir con ser un método basado en probabilidades para la toma de decisiones.

Modelo Ocultos de Markov

Las primeras publicaciones sobre los Modelos Ocultos de Markov fueron realizadas por Baum y sus colegas a finales de los 60's y principios de los 70's, sin embargo, no eran aplicados a situaciones tales como el reconocimiento de comportamientos por algunos problemas, el principal fue que estas publicaciones fueron realizadas en revistas dirigidas a un publico dedicado a la investigación en matemáticas, otro problema fue que la teoría presentada no era muy clara por lo que era difícil encontrar sus aplicaciones.

En la segunda mitad de la década de los 80's, los Modelos Ocultos de Markov comenzaron a ser aplicados al análisis de secuencias biológicas, en particular al ADN. Desde entonces, se han utilizado bastante en el campo de la bioinformática.

Definición 2.2.15. Un proceso de Markov o cadena de Markov, que recibe su nombre por el matemático ruso Andrei Markov, es una serie de eventos s en la cuál la probabilidad de que ocurra un evento en el tiempo t depende solamente de la probabilidad del evento inmediato anterior $t-1$.

Las cadenas o modelos de este recuerdan sólo el último evento y esto condiciona las posibilidades de la ocurrencia en eventos futuros.

Si denotamos al conjunto de N distintos estados como $S_1, S_2, ..., S_N$, y con q_t al estado actual en el tiempo t, entonces, para el caso de las cadenas de Markov, la probabilidad de pasar de un estado a otro solo depende del estado actual y del anterior.

$$P(q_t = S_j | q_{t-1} = S_i, q_{t-2} = S_h, \ldots) \tag{2.2.37}$$
$$= P(q_t = S_j | q_{t-1} = S_i) \tag{2.2.38}$$

Si este proceso es considerado independiente del tiempo, podemos generar una matriz de probabilidades de transición A de la forma:

$$a_{i,j} = P(q_t = S_j | q_{t-1} = S_i), \quad 1 \leq i, j \leq N \tag{2.2.39}$$

Esta matriz cumple con las siguientes propiedades:

$$a_{i,j} \geq 0 \tag{2.2.40}$$
$$\sum_{j=1}^{N} a_{i,j} = 1 \tag{2.2.41}$$

Para cualquier renglón i de la matriz a.

Considere el modelo de Markov que describe las probabilidades de cambios de clima de un día a otro, ver la figura 2.8.

Suponga que A, es la matriz de probabilidad de transiciones, se definen los estados del clima como: S_1 = Lloviendo, S_2 = Nublado, S_3 = Soleado.

La matriz de probabilidad de transiciones A se define de la siguiente manera:

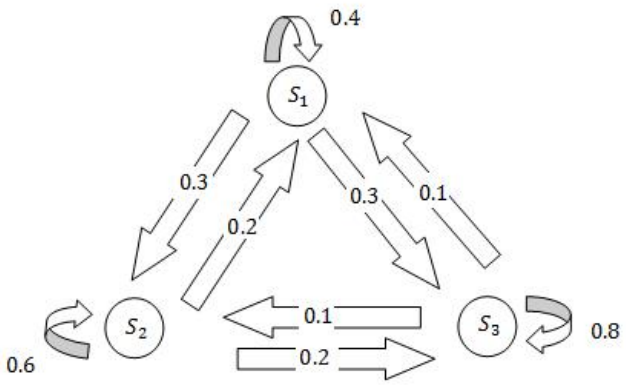

Figura 2.8: Modelo de Markov que considera los cambios de clima de un día a otro

$$A = \begin{array}{c} \\ S_1 \\ S_2 \\ S_3 \end{array} \begin{array}{c} \begin{array}{ccc} S_1 & S_2 & S_3 \end{array} \\ \begin{pmatrix} 0.4 & 0.3 & 0.3 \\ 0.2 & 0.6 & 0.2 \\ 0.1 & 0.1 & 0.8 \end{pmatrix} \end{array} \tag{2.2.42}$$

Dado que el clima en el día 1 es *nublado*, ¿cual es la probabilidad de que en los próximos 4 días, el clima presentado sea: *nublado - soleado - soleado -nublado - lloviendo*?, para responder a esta pregunta, es necesario definir el concepto de observación.

Definición 2.2.16. Se utiliza el símbolo π_i para denotar el estado inicial en S_i

Definición 2.2.17. Una observación O es una secuencia de estados que se presentan a lo largo del tiempo, $O = \{\underbrace{S_i}_{t=1}, \underbrace{S_j}_{t=2}, \underbrace{S_k}_{t=3}, \ldots, \underbrace{S_M}_{t=M}\}$. Donde $\underbrace{X}_{t=z}$, se refiere a la ocurrencia de X en el tiempo z.

Para el caso del modelo del clima, la pregunta anterior se puede plantear como: ¿Cuál es la probabilidad de O dado el modelo, donde $O = \{\underbrace{S_2}_{t=1}, \underbrace{S_3}_{t=2}, \underbrace{S_3}_{t=3}, \underbrace{S_2}_{t=4}, \underbrace{S_1}_{t=5}\}$.

$$\pi_i = P(q_1 = S_i) \qquad 1 \leq i \leq N \tag{2.2.43}$$

$$\begin{aligned} P(O|Modelo) = & \\ & = P(S_2, S_3, S_3, S_2, S_1|Modelo) \\ & = P(S_2) \cdot P(S_3|S_2) \cdot P(S_3|S_3) \cdot P(S_2|S_3) \cdot P(S_1|S_2) \\ & = \pi_2 \cdot a_{2,3} \cdot a_{3,3} \cdot a_{3,2} \cdot a_{2,1} \\ & = 1 \cdot (0.2) \cdot (0.8) \cdot (0.1) \cdot (0.2) \\ & = 3.2 \cdot 10^{-3} \end{aligned} \tag{2.2.44}$$

Con este modelo es posible responder a preguntas como: ¿Cuál es la probabilidad de que el clima se mantenga *nublado* durante n días?. Es posible contestar esta pregunta al evaluar la probabilidad de la observación

$$O = \{\underbrace{S_i}_{1}, \underbrace{S_i}_{2}, \underbrace{S_i}_{3}, \ldots, \underbrace{S_i}_{n}, \underbrace{S_j}_{n+1} \neq S_i\}$$

donde $S_i = S_2$, para el caso del modelo del clima:

$$P(O, Modelo, q_1 = S_i) = a_{i,i}^{d-1}(1 - a_{i,i}) = p_i(d) \tag{2.2.45}$$

El resultado $p_i(d)$ es la función de densidad de probabilidad de duración d en el estado i. Con base en este resultado, es posible obtener el número esperado de observaciones (o de duración) en un estado.

$$\overline{d_i} = \sum_{d=1}^{\infty} d p_i(d) \tag{2.2.46}$$

$$= \sum_{d=1}^{\infty} d(a_{i,i})^{d-1}(1 - a_{i,i}) = \frac{1}{1 - a_{i,i}} \tag{2.2.47}$$

Con este resultado, aplicado al modelo del clima es posible calcular el número probable de días consecutivos con cada clima.

$$\overline{d_1} = \frac{1}{1 - 0.4} = 1.6\text{días} \tag{2.2.48}$$

$$\overline{d_2} = \frac{1}{1 - 0.6} = 2.5\text{días} \tag{2.2.49}$$

$$\overline{d_3} = \frac{1}{1 - 0.8} = 5\text{días} \tag{2.2.50}$$

Un "Modelo Oculto de Markov" es un modelo estadístico donde se asume que el sistema a modelar es un proceso de Markov con parámetros desconocidos (de ahí el nombre de Modelos Ocultos). El objetivo es determinar la probabilidad de esos parámetros desconocidos a partir de las observaciones arrojadas por el modelo.

Un "Modelo Oculto de Markov" es también una máquina de estados donde al igual que en los modelos determinísticos, las transiciones dependen de la ocurrencia de algún símbolo, la diferencia es que a cada transición le corresponden dos salidas, una donde se describe la probabilidad de esa transición y la otra es una función de probabilidad (en el caso de los procesos de Markov, esta corresponde a las observaciones).

En general, el paradigma operacional según [9] es como sigue: seleccionar un estado inicial de acuerdo a alguna distribución de probabilidad, en cada paso, se genera un símbolo de salida por la definición del modelo, se cambia a otro estado de acuerdo a la matriz de probabilidades de transición.

Un Modelo Oculto de Markov se puede ver como un doble proceso estocástico, uno de ellos que se encuentra "*escondido*", y el otro que produce la secuencia de observaciones. el siguiente ejemplo mostrará mejor cómo es este proceso.

El problema de las urnas y las pelotas Imagine que se encuentra en un cuarto encerrado, fuera de él hay N urnas con M pelotas de colores. Una persona selecciona una urna por medio de una función de distribución de probabilidad y saca una pelota de la urna, el color de la pelota es agregado al conjunto de observaciones. Una nueva urna es seleccionada de acuerdo a la función de distribución de probabilidad de la urna actual y se repite el proceso. Al finalizar se contará con un conjunto finito de observaciones.

Para modelar este proceso se utilizará un estado para representar a cada urna, cada estado contará con N transiciones, que representan las transiciones a los otros estados (urnas) con base a una función de distribución de probabilidad, además cada estado tendrá M probabilidades que indican la posibilidad de obtener una pelota de cada color, la figura 2.9 muestra el Modelo Oculto de Markov para este problema.

Definición formal de los Modelos Ocultos de Markov

1. N, número de estados del modelo.
2. $S = \{S_1, S_2, \ldots, S_N\}$, conjunto de estados, el estado q en el tiempo t se define como q_t.

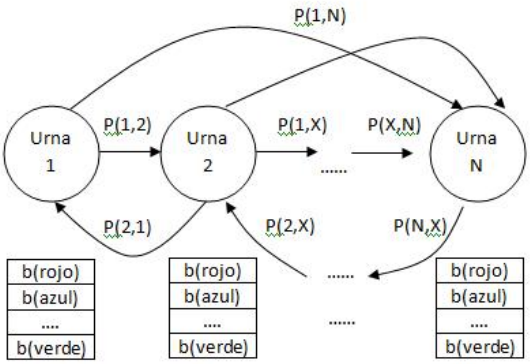

Figura 2.9: Modelo Oculto de Markov para el problema de las urnas.

3. M, número de observaciones por estado.

4. $V = \{V_1, V_2, \ldots, V_M\}$, el alfabeto utilizado para las observaciones.

5. $A = \{a_{i,j} | P(q_t = S_j | q_{t-1} = S_i)\}$, la matriz de probabilidad de transiciones.

6. $B = \{b_j(k)\}$ probabilidad de la observación V_k en el estado S_j.

$$b_j(k) = P(V_k \text{ en el tiempo t } | q_t = S_j), \qquad 1 \leq j \leq N \qquad 1 \leq k \leq M \tag{2.2.51}$$

7. $\pi = \{\pi_i\}$, probabilidad para S_i de ser el estado inicial.

$$\pi_i = P(q_1 = S_i), \qquad 1 \leq i \leq N \tag{2.2.52}$$

Dando los valores para cada uno de los elementos que definen un Modelo Oculto de Markov, es posible obtener un conjunto de observaciones $O = \{O_1, O_2, \ldots, O_T\}$ donde cada observación cumple $O_i = V_k$ y T es el número de observaciones en la secuencia.

La generación del conjunto O es como sigue:

1. Seleccionar un estado inicial $q_1 = S_i$ de acuerdo a π.

2. $t = 1$, inicializar el tiempo.

3. Seleccionar $O_t = V_k$ de acuerdo a la distribución de probabilidad $b_i(k)$ del estado S_i.

4. Pasar al siguiente estado $q_{t+1} = S_j$ con base en la distribución de probabilidad del estado $S_i = a_{i,j}$.

5. Incrementar el tiempo en pasos discretos de 1, $t = t + 1$, si $t < T$ regresar al paso 3, en otro caso terminar el proceso.

La forma estándar de describir un Modelo Oculto de Markov es:

$$\lambda = \{A, B, \pi\} \tag{2.2.53}$$

Problemas de los Modelos Ocultos de Markov Existen tres problemas que deben ser resueltos antes de poder utilizar los Modelos Ocultos de Markov en problemas reales.

1. Teniendo la secuencia de observaciones, $O = \{O_1, O_2, \ldots, O_T\}$ y el modelo $\lambda = \{A, B, \pi\}$, ¿Cómo calcular eficientemente $P(O, \lambda)$, la probabilidad de una observación en un modelo?.

2. Con la secuencia de observaciones $O = \{O_1, O_2, \ldots, O_T\}$ y el modelo λ, ¿Cómo obtener una secuencia de estados $Q = \{q_1, q_2, \ldots, q_T\}$ que sea óptima, o que mejore la observación O?.

3. ¿Cómo ajustar los parámetros de λ para maximizar $P(O, \lambda)$?.

La solución a cada uno de estos problemas tiene que ver con el rendimiento del modelo, para el caso del problema 3, esta es la forma en que se entrenará al modelo, la solución al problema 2 permite optimizar el modelo obtenido, y con la solución al problema 1 se obtendrá la probabilidad de pertenencia de un conjunto de observaciones a un modelo, para el caso de este trabajo esta medida se utilizará para reconocer comportamientos. Para referencias sobre la solución a cada uno de los problemas se puede consultar la bibliografía [10] y [9].

Uso de los Modelos Ocultos para reconocimiento de comportamientos En esta sección se explica de manera general como se pueden utilizar los Modelos Ocultos de Markov para el reconocimiento de comportamientos.

Primero, debe considerarse que se cuenta con un conjunto de secuencias, las cuales describen los comportamientos en escena y de cada una se conoce el tipo de comportamiento que representa. Se utilizan todas las secuencias de un comportamiento para entrenar el Modelo Oculto de Markov correspondiente. Se calculan las secuencias óptimas de estados mediante la solución al problema 2, el

modelo esta listo para recibir secuencias, cada secuencia será procesada por cada uno de los Modelos Ocultos de Markov y para el reconocimiento, se tomará el indicado por la máxima probabilidad obtenida.

2.3. Trabajo Relacionado

Existen varios trabajos enfocados al reconocimiento de comportamientos de personas que hacen uso de diversos métodos, algunos de ellos son mencionados en esta sección.

En [1] se hace uso de la técnica del fondo adaptable, obtienen la desviación estándar σ de cada píxel $I_{i,j}$, si $2\sigma \geq a_{i,j}$ el píxel se considera fuera del fondo, si después del tiempo $t = 3\gamma$, donde γ es el tiempo de muestreo de la cámara, los nuevos valores del píxel sustituyen a los anteriores, para la parte de seguimiento de objetos, utilizan la trayectoria y velocidad del objeto junto con una función de correlación, en la parte de clasificación, utilizan una red neuronal con tres capas y el método de "back-propagation" para aprendizaje, las clases identificadas son: humanos, automóviles y grupos de humanos, por otra parte, utilizan un método de "esqueletización en estrella" para el reconocimiento de comportamientos humanos.

En [11] utilizan como base un método de fondo adaptable combinado con la diferencia de imágenes, produciendo un conjunto de regiones etiquetadas como *en movimiento* o *sin movimiento*, esta combinación resulta en un método más robusto que el presentado en [1], el cuál no genera regiones de movimiento. Para la parte de seguimiento, cada objeto es asociado con un conjunto de datos, estos son:

1. p = coordenadas de posición en toda la imagen.
2. δp = incertidumbre en la coordenadas de posición.
3. $\vec{v}$ = Velocidad del objeto.
4. $\delta \vec{v}$ = Incertidumbre en la velocidad.
5. Coordenadas del cuadro mínimo envolvente.
6. Plantilla de intensidad del objeto.
7. Medida de confiabilidad.
8. Medida de permanencia.

Con estos datos es posible predecir la nueva ubicación del objeto p_{n+1} de la manera típica, donde Δt es un intervalo de tiempo entre dos imágenes:

$$p_{n+1} = p_n + \vec{v}_n \Delta t \tag{2.3.1}$$

y la incertidumbre en la posición, esto es δp_{n+1}, es la suma de la incertidumbre de la posición en el tiempo anterior δp_n con la incertidumbre en la velocidad $\delta \vec{v}_n$.

$$\delta p_{n+1} = \delta p_n + \delta \vec{v}_n \Delta t \tag{2.3.2}$$

Estos datos se utilizan para seleccionar al mejor candidato del conjunto de las regiones cercanas a la región de interés, y los datos antes calculados son actualizados para la nueva región.

La velocidad estimada del objeto $\tilde{v}_{n+1}$, que se obtiene en el calculo de la medida de similitud, es utilizada para deducir la velocidad actual del objeto en seguimiento:

$$\vec{v}_{n+1} = \alpha \tilde{v}_{n+1} + (1 - \alpha)\vec{v}_n \tag{2.3.3}$$

y de la misma forma, la incertidumbre en la velocidad es obtenida de la siguiente manera:

$$\Delta \vec{v}_{n+1} = \alpha \mid \vec{v}_{n+1} - \tilde{v}_{n+1} \mid + (1 - \alpha \vec{v}_n) \tag{2.3.4}$$

De acuerdo a los autores, el algoritmo es robusto a posibles divisiones y/o mezclas de los objetos, lo cual es conocido como "split and merge", al final de esta fase se hace la eliminación de objetos considerados como "falsas alarmas".

Para la parte de clasificación, utilizan dos métodos, el primero de ellos se ayuda de una red neuronal con tres capas, las clases identificadas son: humanos, vehículos y grupos de humanos, para el entrenamiento de esta red se utiliza el algoritmo de "back-propagation", el segundo método realiza un análisis de discriminación lineal, este método contiene dos módulos, uno para identificar formas y el otro para colores, cada módulo devuelve la máxima probabilidad de pertenencia a una clase utilizando el algoritmo K-NN.

En el reconocimiento también se cuenta con dos métodos, el primero se basa en una "esqueletización en estrella" para reconocer el comportamiento de distintos objetos por medio de correlaciones con plantillas que definen cada comportamiento, el segundo usa a los Modelos Ocultos de Markov para identificar la interacción entre varios objetos, las actividades reconocidas son: 1) Humano entrando a un vehículo,

2) Humano saliendo de un vehículo, 3) Humano saliendo de un edificio, 4) Humano entrando a un edificio, 5) Vehículo estacionado y 6) Encuentro de humanos.

Para el entrenamiento, se generan secuencias simuladas con estos comportamientos y las características utilizadas son la distancia y la velocidad de los objetos.

En [12] manejan la diferencia de imágenes para la detección de movimiento y posteriormente se agrupan los píxeles para formar regiones. Para la parte de clasificación se hace uso de dos elementos clave, el primero es un parámetro de clasificación utilizado para distinguir entre los diferentes tipos de objetos y el segundo es una medida de consistencia en el tiempo, esta medida es utilizada de acuerdo a lo siguiente: si un objeto se mantiene en escena durante un determinado tiempo, entonces se vuelve un candidato para ser clasificado, de lo contrario se tomará como parte del fondo. La clasificación se realiza en cada instante de tiempo y se va almacenando, después de un determinado tiempo se procede a tomar la clase que mayor frecuencia presente en el proceso, esto evita clasificaciones incorrectas a causa de oclusiones en los objetos. Una de las principales características utilizadas es la complejidad en la forma del objeto, ésta se obtiene de la siguiente manera:

$$D = \frac{\text{Perímetro}^2}{\text{Área}} \tag{2.3.5}$$

El seguimiento de objetos se realiza en pasos posteriores a la clasificación y consiste en lo siguiente: una vez obtenidas las plantillas de los objetos, son utilizadas para entrenamiento del algoritmo de seguimiento. Básicamente el algoritmo trabaja con una medida de similitud entre las plantillas almacenadas y las regiones obtenidas por el algoritmo de detección de movimiento. Se calcula la medida de correlación solo para los píxeles en movimiento para que el cálculo se realize más rápido, una posible mejora a este algoritmo consistiría en que sólo los píxeles en movimiento sean los guardados en la plantilla de cada objeto, con esto se lograría disminuir aún más el tiempo de procesamiento. Para lograr esto es necesario calcular la nueva posición del objeto haciendo uso de la velocidad del objeto, esta se calcula con base en la posición del centroide entre una imagen en el tiempo t y otro en el tiempo $t+1$ como se observa en la figura 2.10, la siguiente ecuación resume lo anterior:

$$\vec{v}_R = \frac{R_{t+n}(p) - R_t(p)}{(t+n) - t} \tag{2.3.6}$$

Donde $R_t(p)$ se refiere a la posición p de la región R en el tiempo t, $R_{t+n}(p)$ se refiere a la posición de la región R desde el tiempo base t más una constante de separación entre tiempos n, y $\vec{v}_R$ es la velocidad de la región R.

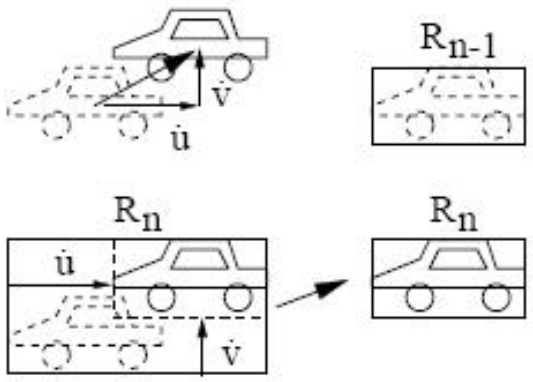

Figura 2.10: Eliminación de partes del fondo usando el conocimiento del desplazamiento del objeto.

En [4] se presenta el desarrollo de un sistema llamado VSIP (Video Surveillance Interpretaion Plataform) desarrollado por el grupo de investigación ORION en INRIA (Institut National de Recherche en Informatique et en Automatique) en Sophia Antipolis. VSIP es un ambiente genérico para combinar algoritmos utilizados en el procesamiento y análisis de videos que tiene la flexibilidad de combinar e intercambiar varios métodos en diferentes estados de la interpretación de secuencias de video. VSIP esta enfocado para ayudar a los desarrolladores a crear sus propios sistemas para ambientes específicos donde sea necesario el reconocimiento de comportamientos. El enfoque utilizado en este sistema es dividir el problema en dos módulos, el primero se encarga de extraer información y reconocer algunos comportamientos primitivos, el segundo hace uso de esta información para reconocer comportamientos más complejos. En la figura 2.11 se presenta de manera general el funcionamiento de este sistema.

El primer módulo se forma mediante tres tareas, la primera consiste en el detector de movimiento y el algoritmo encargado del seguimiento de los objetos, además de la generación de grafos de los objetos en movimiento para cada una de las cámaras. La siguiente tarea consiste en la fusión de los grafos. En la tercer tarea, el grafo obtenido es utilizado para el seguimiento de personas, vehículos y otros objetos que pueden estar en la escena. Para cada uno de los objetos identificados, el módulo de reconocimiento de comportamientos realiza tres niveles de razonamiento, que son: estados, eventos y escenarios.

En [7] se desarrollo un sistema para el reconocimiento de comportamientos humanos en secuencias de video, los comportamientos humanos son modelados como una secuencia de eventos estocásticos. Los eventos son descritos por un vector

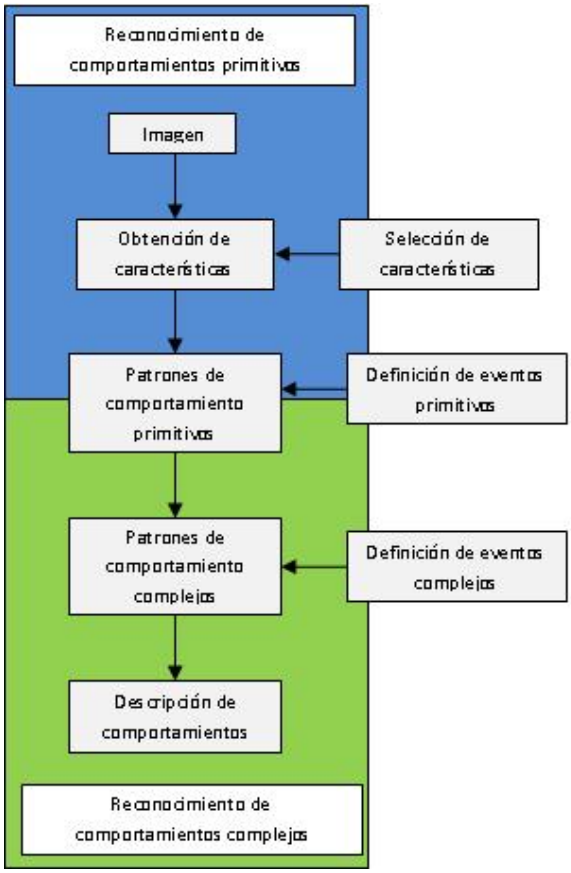

Figura 2.11: Flujo de la información para el reconocimiento.

característico que contiene información de la trayectoria de los objetos (posición y velocidad). El reconocimiento de estos eventos se logra por medio de la búsqueda en una base de datos de imágenes con características que representan cada uno de los eventos.

Este trabajo utiliza Modelos Ocultos de Markov, una parte de ellos ayuda a "suavizar" la secuencia de eventos y la otra se encarga de la interpretación final de las escenas por medio del cálculo de la máxima probabilidad del conjunto de Modelos Ocultos que contiene información para describir la secuencia de eventos en la escena. Este trabajo es probado en secuencias de video de cámaras de vigilancia y en videos de partidos de tennis.

En [8] se hace uso del conocimiento experto para el modelado de los comportamientos, el objetivo del trabajo es el reconocimiento automático de comportamientos en abejas, es este trabajo se hace uso de una combinación de los Modelos Ocultos de

Markov con un algoritmo de clasificación que utiliza una combinación de regresión de kernel. El sistema es capaz de realizar la detección de objetos en movimiento y la interpretación de la imagen al mismo tiempo, las fases del sistema son: seguimiento de objetos, obtención de las coordenadas del objeto, clasificación mediate el algoritmo de regresión (KR), se genera un Modelo Oculto para cada uno de los comportamientos. Los comportamientos considerados son: bailar, imitar un bailador, activa e inactiva.

En [6] se presenta un sistema de vigilancia distribuido para interiores, en este trabajo, se hace uso de los Modelos Ocultos de Markov Abstractos que son una extensión de los originales, la diferencia consiste en que en los modelos abstractos, las cadenas de Markov son reemplazadas por jerarquías en las políticas de Markov, para esta política de jerarquías cada comportamiento puede ser representado como una política en su nivel de abstracción correspondiente, el método para la creación de estas políticas es el siguiente: Se define una región con jerarquía y se define un conjunto de políticas para cada región definida, dependiendo de los comportamientos que se deseen reconocer en cada región, se generan las políticas necesarias para representarlo. Las políticas de más bajo nivel o más simples son definidas primero, luego las segundas que se generan por la combinación de varias políticas del nivel anterior y así hasta conformar los comportamientos que se quieren reconocer.

En [13] se presenta una alternativa para el algoritmo de aprendizaje para los Modelos Ocultos de Markov Abstractos, su enfoque se basa en el estudio del algoritmo de aprendizaje, logrando otro método para llevar a cabo la misma tarea.

En [14] se desarrollo un sistema de vigilancia para exteriores y para secuencias de video de deportes, las salidas de este programa es una descripción de lo que pasa en escena y las interacciones que existen entre los objetos. La entrada al sistema es un conjunto de datos obtenidos por otros algoritmos que procesan la imagen, las regiones de interés se encuentran bien definidas, una de las características que hace interesante a este trabajo es el procesamiento de la imagen para obtener la dirección de la mirada de cada persona con el fin de ayudar a reconocer mejor su comportamiento. En la figura 2.12 se puede apreciar esta característica.

En este trabajo se utilizan los Modelos Ocultos de Markov para el reconocimiento de comportamientos, se procesa una secuencia de observaciones para cada comportamiento que se tiene los cuales son representados por un Modelo Oculto, y se selecciona el comportamiento que corresponde al modelo con la máxima probabilidad obtenida. Este trabajo al igual que otros, reconoce comportamientos en personas, pero además de esto intenta dar una explicación del porqué de los comportamientos. Es necesario el conocimiento a priori de la escena para este caso.

En [15] se cuenta con un sistema que se encarga de verificar constantemente el estado de un automóvil, las entradas al sistema son: varias secuencias de video que

Figura 2.12: Una característica importante: ¿A donde esta mirando la persona?.

permitan obtener información sobre el ambiente, otro sensor que permita obtener el punto de vista del conductor y uno más que se encuentre en la cabeza del conductor, adquisición en tiempo real de los datos del automóvil tales como: velocidad, aceleración, reportes de frenos del carro, condiciones del motor, etc. En este trabajo se utilizaron 70 personas para entrenar el Modelo Oculto y su potencial extensión, después del entrenamiento, estos sistemas son capaces de predecir las condiciones hasta un segundo antes de que se presenten.

En [16] se sugiere que un algoritmo de seguimiento debe cumplir las siguientes ocho reglas:

1. **Fondo estático:** Cuando en la escena se encuentran multiples objetos, el fondo se considera como estático cuando la mayoría de los objetos se encuentran en movimiento.

2. **Variación en el tamaño del objetivo:** El tamaño del objetivo se reduce si se aleja de la cámara, un método de escalamiento de objetivos debe ser introducido en el algoritmo de seguimiento.

3. **Oclusión o perdida temporal de objetivo:** Cuando el objeto en seguimiento es perdido debido a que pasa por detrás de otro objeto, esto es conocido como oclusión. En este caso el algoritmo de seguimiento debe recuperar automáticamente al objeto perdido.

4. **Modelado del objetivo:** Un modelo del objetivo debe ser conocido por el sistema, para asegurar un mejor seguimiento.

5. **Detección automática de objetivos:** El algoritmo de seguimiento debe identificar automáticamente los objetos a candidatos a seguir.

6. **Tiempo real:** El algoritmo debe ser computacionalmente simple y debe estar optimizado para su ejecución en tiempo real.

7. **Trayectoria del objetivo:** El algoritmo debe ser capaz de mantener al objetivo aún si existen cambios de dirección (esto depende de los objetos a seguir).

8. **Conocer la velocidad del objetivo:** La velocidad puede cambiar bruscamente, puede también ser constante, incrementar o decrementar conforme pasa el tiempo.

Capítulo 3

Propuesta de Solución

En esta sección se presenta la metodología utilizada para la solución del problema planteado en este trabajo.

3.1. Esquema general de la solución propuesta

La meta en este trabajo de tesis es el desarrollo de un sistema que sea capaz de reconocer comportamientos humanos por medio del procesamiento de secuencias de imágenes, en los primeros pasos es necesario reconocer los objetos de interés para que posteriormente se lleven a un etapa de análisis y reconocimiento de los comportamientos que estos presenten.

El algoritmo utilizado en este sistema debe ser robusto al ruido que puedan presentar las imágenes, ya que como serán tomadas en ambientes externos es común encontrar varios factores que dificultan el procesamiento de las imágenes. La primera etapa consiste en la detección y extracción de los objetos en movimiento dentro de la escena, una vez obtenidos se pasa a la etapa de seguimiento donde se utiliza un algoritmo de seguimiento sencillo para obtener las trayectorias de los objetos, además de ayudar a eliminar los objetos considerados como *falsas alarmas*, la siguiente etapa realiza la clasificación de cada elemento en procesamiento por el algoritmo de seguimiento, las clases a la que los objetos pueden pertenecer son *Persona* o *Auto*. En la etapa final del proceso se realiza el reconocimiento de comportamientos en los objetos, utilizando la información obtenida de cada elemento en escena, detectando actividades sospechosas o maliciosas.

Una de las tareas de mayor dificultad será la manera en que el sistema identificará los comportamientos, en esta parte toman lugar los métodos de aprendizaje utilizados

más frecuentemente para la interpretación de secuencias de video, en este ámbito existen dos enfoques importantes:

- Redes neuronales
- Modelos gráficos probabilísticos

En este trabajo se hace uso de los modelos gráficos probabilísticos debido a las ventajas que proporciona al momento de manejar la incertidumbre en la tarea del procesamiento de video.

En este tipo de modelos existen varios ejemplares importantes, dos de las más importantes son:

- Redes bayesianas
- Modelos Ocultos de Markov

Los Modelos Ocultos de Markov han sido utilizados recientemente con resultados satisfactorios en el reconocimiento e interpretación de secuencias de video debido a que son capaces de describir las relaciones probabilísticas entre atributos y actividades.

El esquema general propuesto para el sistema comprende los siguientes dos módulos:

1. Módulo de visión. Este módulo será el encargado del procesamiento de las imágenes lo cual implica la representación de la escena en términos de objetos en movimiento, a grandes rasgos, este módulo realizará la tarea de segmentación, seguimiento y clasificación de los objetos en movimiento.
2. Módulo de interpretación. En este módulo se realizará la tarea del reconocimiento de comportamientos, las entradas a este módulo serán las características que conforman la escena, estas entradas serán procesadas por los Modelos Ocultos de Markov los cuales estarán entrenados para reconocer los comportamientos.

El diagrama general a bloques de este método se presenta en la figura 3.1.

Como se puede observar, cada módulo requiere del desarrollo de tareas específicas, la unión de estos módulos conformará la *unidad de procesamiento* (se mencionó en el capítulo 1), la cuál se encuentra en todos los sistemas de visión por computadora, las tareas a desarrollar para cada uno de estos módulos son las siguientes:

- Implementación y pruebas de métodos para la detección automática de objetos en movimiento (Etapa de filtrado y segmentación).

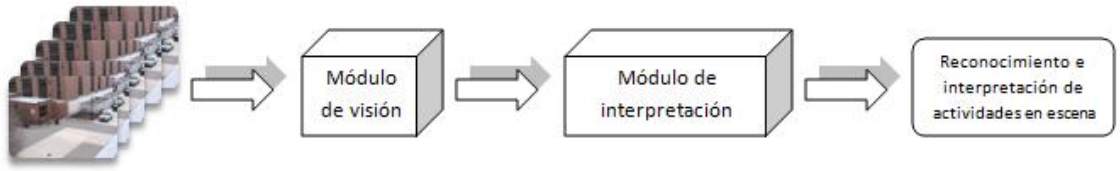

Figura 3.1: Diagrama general a bloques de solución.

- Seguimiento de objetos.
- Extracción de características de los objetos y clasificación (Etapa de clasificación).
- Reconocimiento de comportamientos con ayuda de los modelos ocultos de Markov (Etapa de interpretación).

3.2. Módulo de visión

En esta sección se describen a detalle las partes que lo conforman, estas son las siguientes:

1. Detección de objetos en movimiento.
2. Seguimiento de objetos.
3. Clasificación de objetos.

El diagrama general de este módulo se presenta en la figura 3.2.

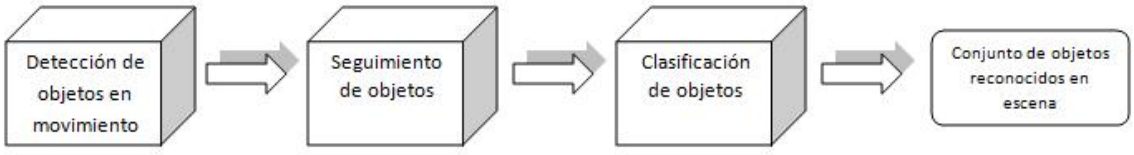

Figura 3.2: Diagrama a bloques del módulo de visión.

Cada una de las partes que conforman al módulo tiene una salida, la *detección de objetos en movimiento* genera un conjunto de regiones las cuales se utilizan como datos

de entrada para el algoritmo de *seguimiento de objetos* para obtener las trayectorias de cada región, la información de cada región es tomada por el algoritmo de clasificación para asignar una clase a cada una, esta información será utilizada por el módulo de interpretación.

3.2.1. Detección de objetos en movimiento

La primer tarea del módulo de visión consistirá en la detección de objetos en movimiento, en la sección 2.2.2 se mencionaron los enfoques más comunes para esta tarea, para lograr este objetivo se realizará una segmentación de la imagen, separando los objetos en movimiento del fondo.

El algoritmo de detección de este trabajo combina distintas técnicas, dos de las más importantes son:

- Extracción de fondo
- Diferencia temporal

Estos algoritmos trabajan bien en sistemas con cámaras fijas, pero tienen algunas restricciones, estas se describieron en la sección 2.2.2. Existen otros métodos, que son más complejos y con frecuencia requieren de hardware especializado para una respuesta en tiempo real, que son utilizados para cámaras en movimiento, un ejemplo es el *flujo óptico*. El método propuesto toma como base la extracción de fondo y la diferencia temporal.

Fondo adaptable

Lo que respecta a la extracción de fondo, el fondo utilizado no será un fondo estático, sino un fondo adaptable, un fondo adaptable es aquel que va cambiando con el tiempo, para lograr que el fondo se actualice constantemente es necesario tomar en cuenta lo siguiente:

1. ¿Cuales son los valores iniciales para el fondo adaptable?
2. ¿Cómo se actualizan los valores del fondo adaptable?

Las respuestas a estas preguntas conformarán la estrategia a utilizar para lograr el fondo adaptable (fa).

Los valores iniciales del fondo adaptable (fa) serán tomados de las primeras n imágenes de la secuencia de video, por lo que consistirá simplemente de un promedio de los valores presentados de cada píxel.

Para cada elemento $p(x) \in fa$, el valor inicial estará dado por:

$$p(x) = \frac{1}{n}\sum_{k=1}^{n} p_k(x) \tag{3.2.1}$$

Donde k indica el número de imagen desde el inicio de la secuencia.

Una vez obtenido el fondo adaptable inicial, es necesario el cálculo de una medida de tolerancia (umbral) para la identificación de movimiento entre el fondo adaptable y las imágenes siguientes, esta medida será la *desviación estándar*. La desviación estándar se obtiene de la siguiente manera para cada uno de los píxeles que conforman la imagen:

$$\sigma(x) = \sqrt{\frac{1}{n}\sum_{k=1}^{n}(x_k - \bar{x})^2} \tag{3.2.2}$$

Donde $\bar{x}$ representa el promedio de cada píxel, en este caso sería un elemento de fa, y x_k representa el valor del píxel en la imagen k.

Tanto el modelo de fondo adaptable (fa) y los umbrales para cada píxel (σ), representan propiedades estáticas del valor de intensidad de los píxeles observados en la secuencia de imágenes.

La manera en que se identificará movimiento con los datos obtenidos será por medio de una diferencia de imágenes entre el fondo adaptable (fa) y la imagen actual (I), si la diferencia es mayor al umbral (σ) para ese píxel, se manejara como *píxel en movimiento*.

$$Mfa(x) = \begin{cases} \text{En movimiento} & \text{si } \mid fa(x) - I(x) \mid > \sigma(x) \\ \text{Sin movimiento} & \text{En otro caso} \end{cases} \tag{3.2.3}$$

La matriz Mfa lleva el control de los objetos en movimiento en la escena.

Diferencia temporal

La diferencia temporal es uno de los métodos más rápidos para la detección de movimiento dentro de secuencias de imágenes, el problema de este método es que no logra detectar la totalidad de los objetos en movimiento, sólo sus contornos como se observó en la figura 2.3.

Es común utilizar una diferencia de imágenes entre 1 o 2 imágenes anteriores, en este trabajo se propone utilizar una diferencia de 5 imágenes para obtener mayor parte del objeto en movimiento.

La manera de detectar movimiento utilizando diferencia temporal de imágenes es: si la diferencia de *todas* las 5 imágenes anteriores excede un umbral, el umbral utilizado será la *desviación estándar* calculada para el fondo adaptable, se marcará al píxel analizado como *en movimiento*.

$$Md(x) = \begin{cases} \text{En movimiento} & \text{si } \mid I_k(x) - I(x) \mid > \sigma(x) \text{ para } 1 \leq k \leq 5 \\ \text{Sin movimiento} & \text{En otro caso} \end{cases} \tag{3.2.4}$$

La matriz Md lleva el control de las regiones en movimiento de la escena.

Detección de movimiento

Combinando estos dos métodos se obtendrán mejores resultados en la detección de movimiento, pero si se aplicaran de la forma mencionada se detectarían varios píxeles donde el valor de su intensidad no cambie demasiado sin embargo este caería fuera del umbral, en este caso la *desviación estándar*. Esto se debe a que gran parte de los datos se encuentran a no más de dos desviaciones estándar de la media como se aprecia en la figura 3.3.

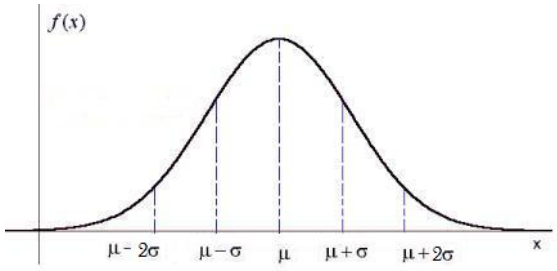

Figura 3.3: Distribución de Gauss, μ representa a la media y σ a la desviación estándar.

Tomando en cuenta esta observación los métodos mencionados se ajustarán de modo que los valores calculados para el umbral σ se encuentren a no más de dos desviaciones estándar, a este valor le llamaremos *sensibilidad*, en este caso se utilizará una sensibilidad de 1.5, por lo que los nuevos valores para los umbrales se verán afectados: $\sigma(x) = 1.5\sigma(x)$.

La detección del movimiento se realizará con la ayuda de los métodos mencionados y la modificación a los umbrales, por lo que se considerará que un píxel presenta movimiento si los dos métodos mencionados lo detectan.

$$M(x) = \begin{cases} \text{En movimiento} & \text{Si } Mfa(x) \text{ y } Md(x) \text{ presentan movimiento} \\ \text{Sin movimiento} & \text{En otro caso} \end{cases} \tag{3.2.5}$$

La matriz M representará la información de movimiento de la escena, los datos del fondo adaptable (fa) y los umbrales (σ) serán actualizados dependiendo de la presencia de movimiento en los píxeles.

La actualización para el fondo adaptable (fa) tiene el siguiente comportamiento:

$$Mfa_{n+1}(x) = \begin{cases} \alpha Mfa_n(x) + (1-\alpha)I_n(x) & \text{No hay movimiento} \\ Mfa_n(x), & \text{Existe movimiento.} \end{cases} \tag{3.2.6}$$

Y para los valores del umbral (σ):

$$\sigma_{n+1}(x) = \begin{cases} \alpha\sigma_n(x) + (1-\alpha)(5\times \mid I_n(x) - Mfa_n(x) \mid), & \text{No hay movimiento} \\ \sigma_n(x), & \text{Existe movimiento} \end{cases} \tag{3.2.7}$$

En ambos casos la variable α indica la frecuencia de actualización de los datos, $\alpha\epsilon[0,1]$, entre más cerca se encuentre de 1, más rápida será la actualización.

Análisis temporal de píxeles

Un método robusto para detección de movimiento debe ser capaz de reconocer en la escena cuando los objetos en movimiento se detienen (esta no es una tarea fácil ya que los métodos tradicionales no son capaces de identificar esta situación), para lograr esta tarea es necesario un análisis en el tiempo de los píxeles antes de determinar si se encuentra en movimiento o no. Este método es útil en la eliminación de "ruido" producido por nubes o cambios de luz en la escena.

Se ha observado que la gráfica que representa el valor de intensidad de un píxel cuando un objeto se mueve a través de él puede mostrar tres características relevantes, estas son:

1. En la gráfica se muestra un cambio de intensidad grande seguido de un periodo de inestabilidad, cuando el objeto pasa por completo la zona donde se encuentra el píxel en cuestión, los valores de intensidad regresan a los anteriores, cuando representaba el fondo, y se estabilizan. Esta situación se presenta en la figura 3.4(a) y se dice que el píxel representa una parte del objeto en movimiento.

2. En la gráfica se presenta un cambio de intensidad grande, seguido de un periodo de inestabilidad, y a diferencia del anterior los valores de intensidad del píxel se estabilizan sin regresar a los valores anteriores. En la figura 3.4(b) se presenta esta situación y se dice que el píxel representa una parte de un objeto en movimiento que se detuvo justo en la zona donde se encuentra el píxel analizado.

3. La gráfica muestra cambios pequeños sin periodos de inestabilidad. Algunos factores como son los cambios de luminosidad logran estos cambios pequeños. Esta situación se presenta en la figura 3.4(c) y se dice que el píxel sigue siendo parte del fondo.

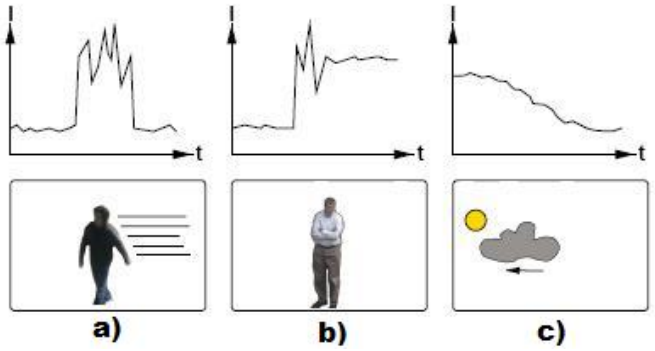

Figura 3.4: Posibles estados de un píxel en escena. (a) Píxel representando parte de un objeto en movimiento. (b) Píxel que representa a un objeto que se detuvo en la zona donde se encuentra el píxel analizado. (c) Cambios pequeños en los valores de intensidad comúnmente generados por cambios de luminosidad.

Para lograr identificar estas características es necesario determinar cuando existen cambios de intensidad significativos y cuales son los valores de intensidad en los que el píxel se mantiene estable después de pasar por un periodo de cambios constantes. Para identificar un cambio de intensidad significativo es necesario observar que la curva de valores de intensidad se re-establezca después de un periodo de tiempo, esto por supuesto, afectará el tiempo de respuesta del sistema. De hecho las decisiones tomadas corresponderán al análisis de k imágenes en el pasado.

Suponiendo que $I_t(x)$ es el valor de intensidad de algún píxel en el tiempo t, para representar el valor de intensidad de ese mismo píxel en k imágenes pasadas al tiempo t, se escribiría $I_{t-k}(x)$. Son necesarias dos medidas para identificar las características antes mencionadas: la primera servirá como una *medida de cambio*, y la segunda será una *medida de estabilidad*, estas medidas se calcularán desde k imágenes en el pasado.

La primer medida, la *medida de cambio* (mc), se calculará como el máximo de la diferencia absoluta de los valores de intensidad del píxel $I_t(x)$ con las k imágenes anteriores al tiempo t. Es decir:

$$mc = \mid I_t(x) - I_{t-i}(x) \mid, 1 \leq i \leq k = 5 \tag{3.2.8}$$

La medida de estabilidad (ms) representa la varianza del valor de intensidad de los píxeles desde k tiempos anteriores, esta medida se calculará de la siguiente manera:

$$ms = \frac{k \sum_{j=-k}^{0} (I_{t+j}(x))^2 - (\sum_{j=-k}^{0} I_{t+j}(x))^2}{k(k-1)} \tag{3.2.9}$$

Con estas medidas es posible establecer alguno de tres estados a cada píxel p, estos estados serán: píxel del *fondo*, píxel de objeto en *movimiento*, píxel de objeto *detenido*. El siguiente algoritmo presenta la forma de utilizar estas medidas para determinar el estado de cada píxel, este se representa con $E(p)$.

```
1  para cada p ∈ I hacer                                  /* Inicialización */
2      E(p) ← fondo
3  fin para cada
4  si E(p) = fondo o detenido Y mc > Umbral entonces
5      E(p) ← movimiento
6  sino
7      si E(p) = movimiento Y ms < Umbral entonces     /* Estabilizado */
8          si I(p) = Mfa(p) entonces
9              E(p) ← fondo
10         sino /* Estabilizado, no regreso a los valores anteriores */
11             E(p) ← detenido
12         fin si
13     fin si
14 fin si
```

Algoritmo 1: Asignación de estado a un píxel

Con esto se conocerá el estado de cada píxel, y al ser unidos para formar regiones se obtendrá el estado de los objetos, esto eso si se encuentran detenidos esperando algún evento o si solo esta pasando por la escena. Antes de unir a los píxeles es necesario eliminar pequeños puntos o "ruido" que aún se encuentre en la matriz de movimiento.

Aplicación de operaciones morfológicas

El resultado de la detección de movimiento seguido del análisis de píxeles, es decir la matriz M, contiene la información de los píxeles en movimiento, es posible que la matriz contenga información "basura" que pudo haberse filtrado en la operación de detección de movimiento, para eliminar esta información será necesario realizar una operación de *apertura* a la matriz M.

Los conjuntos utilizados para la operación de apertura en este trabajo son dos matrices de 3×3, en la figura 3.5 se muestran estas matrices.

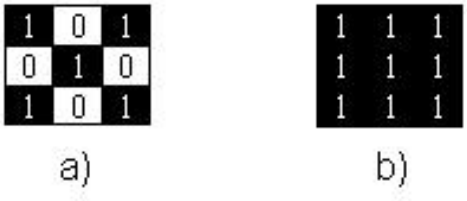

Figura 3.5: Matrices utilizadas para la operación de apertura. (a) para la erosión y (b) para la dilatación.

Generación de regiones

Debido a que para el método de seguimiento no es suficiente con conocer los píxeles que se encuentran en movimiento sino que necesita un conjunto de objetos para poder seguirlos es necesario aplicar un proceso más a la matriz que administra a los píxeles en movimiento M, a partir de está se generará un conjunto de objetos, estos objetos se obtienen conectando cada uno de los píxeles que lo conforman.

El método utilizado es el siguiente: para cada $p \in M$ marcado como *en movimiento* conectar con sus vecinos que también se encuentren marcados como *en movimiento* para formar una región. Las vecindades en las que se buscarán estos vecinos serán ocho. Ver figura 3.6.

Figura 3.6: Vecindades donde se buscará movimiento para el punto p.

Si los elementos vecinos de p, se encuentran marcados como *en movimiento* se agregan a la región que se esta generando y se procede del mismo modo para cada uno de ellos, este método se puede entender como una búsqueda en anchura sobre los píxeles en movimiento, la búsqueda en anchura requiere el uso de una estrucutra de datos tipo *cola*.

En el algoritmo 2 se detalla el método para la generación de regiones.

1 **para cada** $p_{i,j} \in I$ *Y* $p_{i,j}$ *esta en movimiento* **hacer** `/* Inicialización */`
2 $p_{i,j} \leftarrow$ No marcado
3 **fin para cada**
5 $p_{i,j} \leftarrow$ Marcado
6 AgregarCola($p_{i,j}$)
7 **mientras** $C \neq \emptyset$ **hacer**
8 $p \leftarrow$ QuitarCola()
9 checaVecino($p_{(i-s),(j-s)}$)
10 checaVecino($p_{(i-s),j}$)
11 checaVecino($p_{(i-s),(j+s)}$)
12 checaVecino($p_{i,(j-s)}$)
13 checaVecino($p_{i,(j+s)}$)
14 checaVecino($p_{(i+s),(j-s)}$)
15 checaVecino($p_{(i+s),j}$)
16 checaVecino($p_{(i+s),(j+s)}$)
17 **fin mientras**
18 **si** *Existen más puntos en movimiento* **entonces**
19 Ir al paso 5 con $p_{i,j}$ como el nuevo punto en movimiento
20 **fin si**

Algoritmo 2: Generación de regiones

Al hacer una búsqueda en los vecinos más cercanos al píxel en movimiento es posible no generar la región *completa* correspondiente al objeto debido a posibles *agujeros* que se encuentren en la región, por lo que es recomendable el uso de una variable que indique que tan cerca se deben de buscar los vecinos para generar las regiones, esta variable "s" se encuentra en los sub-índices que indican la posición del píxel cuando se realiza el llamado a la función *checaVecino*, esta variable indica que tan lejos se deben buscar vecinos para el píxel y ayuda a evitar la división de regiones por la presencia de *agujeros*.

El procedimiento *checaVecino* utilizado en el algoritmo 2 se encarga de agregar los píxeles marcados como *en movimiento* a la cola para posteriormente ser procesados. Este procedimiento se detalla en el algoritmo 3.

```
1 si punto_{i,j} = "en movimiento" Y "No marcado" entonces
2     punto_{i,j} ← Marcado
3     AgregarCola(punto_{i,j})
4     actualizaFronteras(punto, P_1(r), P_2(r))
5 fin si
```

Algoritmo 3: Procedimiento. checaVecino(*punto*)

Al mismo tiempo que se genera la región se obtendrán los puntos frontera de esta, estos serán identificados como:

- $P_1(r) = (x_1, y_1)$, que representará la coordenada superior izquierda de la región en la matriz que lleva el control del movimiento M.
- $P_2(r) = (x_2, y_2)$, que representará la coordenada inferior derecha de la región en la matriz que lleva el control del movimiento M.

Estos puntos serán calculados de la siguiente manera: Si existe un elemento vecino, digamos v, al punto p se compararán las coordenadas de los puntos frontera con las coordenadas del punto v, en el algoritmo 4 se presenta esta comparación, la cuál es realizada al hacer el llamado a la función *actualizaFronteras()* dentro del algoritmo 3.

```
1  si x_v < x_1 entonces
2      x_1 ← x_v;
3  fin si
4  si y_v < y_1 entonces
5      y_1 ← y_v;
6  fin si
7  si x_v > x_2 entonces
8      x_2 ← x_v;
9  fin si
10 si y_v > y_2 entonces
11     y_2 ← y_v;
12 fin si
```

Algoritmo 4: Procedimiento. actualizaFronteras(v, $P_1(r)$, $P_2(r)$)

En la figura 3.7 se muestra el diagrama de flujo para el algoritmo de detección de movimiento.

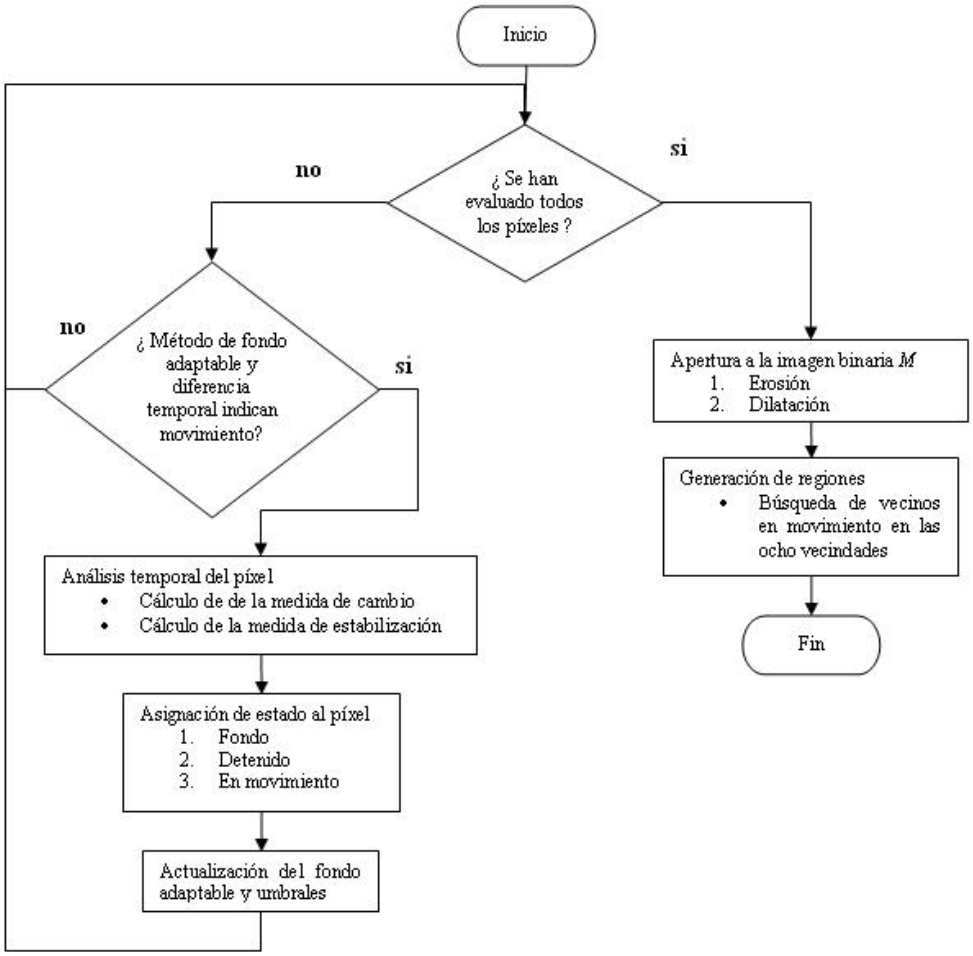

Figura 3.7: Diagrama de flujo del algoritmo de detección de movimiento.

3.2.2. Seguimiento de objetos

El proceso de seguimiento se lleva a cabo para cada una de las regiones que conforman a los objetos en movimiento, debido a que pueden existir varias regiones en movimiento al mismo tiempo en la escena es necesario utilizar algún medio de almacenamiento para su administración, se propone el uso de un estructura de datos tipo lista para poder compararlas con las regiones generadas en las imágenes siguientes.

Las tareas principales del algoritmo de seguimiento para este trabajo son:

1. Agregar nuevas regiones a la lista de objetos en seguimiento.

2. Asociar las regiones obtenidas por el método de detección de movimiento con las regiones en seguimiento.

3. Obtener la mejor similitud entre las regiones generadas por el método de detección de movimiento y las regiones en seguimiento.

4. Eliminar falsas alarmas.

Al termino del proceso de seguimiento, cada región será marcada con una etiqueta numérica para llevar seguimiento de ella, además se llevara un registro de algunas características que serán utilizadas por el método de *clasificación de objetos*.

Agregar regiones a la lista y asociar regiones para seguimiento

Al termino del método de *detección de objetos en movimiento* se obtiene un conjunto de regiones, las cuales deben ser análizadas para saber si se trata de una región de interés o si es algún objeto que puede ser considerado como *falsa alarma*.

En la primera etapa se aplica un filtro a las regiones generadas para validar el tamaño de su área, si él área de la región se encuentra dentro de un determinado tamaño entonces es posible que se trate de un objeto de interés y se agrega a la lista de seguimiento, de lo contrario se rechaza.

El área es el número total de píxeles que conforman al objeto, al conocer los puntos frontera de la región es posible realizar un barrido desde $P_1(r)$ hasta $P_2(r)$ e incrementar el área de la región al encontrar un punto que se encuentre en movimiento. Ver figura 3.8.

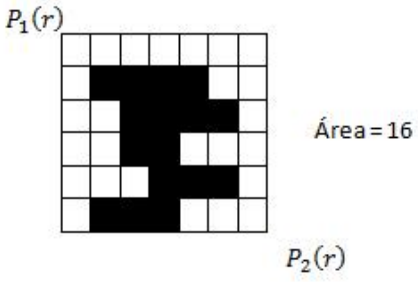

Figura 3.8: Cálculo del área de una región desde el punto $P_1(r)$ hasta $P_2(r)$.

Esta característica de selección es utilizada para no tener que procesar objetos que sean demasiado pequeños o demasiado grandes. Considerando que Lr es la lista

de objetos en seguimiento y que r es una región generada en la detección de objetos en movimiento, entonces:

$$Lr = \{r | \text{área mínima} \leq Area(r) \leq \text{área máxima}\} \tag{3.2.10}$$

El tamaño de estos objetos dependerá en gran medida de la distancia en que se encuentren a la cámara, los tamaños utilizados en este trabajo fueron calculados con base a observaciones realizadas en las secuencias de imágenes utilizadas para este trabajo.

Cada uno de los elementos de la lista de regiones llevará asociado un conjunto de datos, estos son los siguientes:

- Un identificador de región $ID(r)$ para etiquetar al objeto en seguimiento.
- Un identificador de región $Ele(r)$ para conocer el número que ocupa en la lista.
- Un punto(x, y) que será el *centroide* del objeto $Ce(r)$.
- La velocidad del objeto $V(r)$.
- Las coordenadas que indican la posición del cuadro mínimo envolvente $P_1(r), P_2(r)$.
- Una plantilla del objeto $Pla(r)$, la cual se utilizará para hacer comparaciones con otros objetos.
- Una plantilla de movimiento del objeto $PlaM(r)$, es una matriz binaria que contiene los puntos en movimiento del objeto, está se utilizará para agilizar los cálculos de similitud entre regiones.
- Un medida de similitud $C(r)$, la cual se utilizará para saber ¿qué tan parecido es este objeto con el que se comparó?.
- Una medida de permanencia $P(r)$, que será utilizada para conocer si el objeto se encuentra aún en escena.
- Varias medidas utilizadas para la clasificación de objetos que serán descritas en la parte de *clasificación de objetos*.

Existen varios casos que pueden presentarse al realizar el seguimiento de objetos, la idea es que en cada imagen de la secuencia de video se mantengan en seguimiento los objetos de interés, comparando los de la lista Lr con los detectados por el método de detección de movimiento. Los posible escenarios para la asociación de regiones son tratados a continuación:

1. El centroide de una región detectada por el método de detección de movimiento esta dentro del cuadro mínimo envolvente de una de las regiones de la lista de objetos en seguimiento. En este caso el ID del objeto en seguimiento es copiado a la nueva región y la medida de permanencia del objeto en seguimiento es incrementada. La nueva región se convierte en la única candidata para sustituir al objeto en seguimiento.

2. Varios centroides de regiones detectadas por el método de detección de movimiento están dentro del cuadro mínimo envolvente de una región en la lista de objetos en seguimiento, a cada una de estas regiones se les asigna el mismo ID del objeto en seguimiento, cada una de estas regiones será una región candidata para sustituir al objeto en seguimiento. La medida de permanencia del objeto en seguimiento es incrementada por cada región candidata.

3. Alguna región de la lista de objetos en seguimiento no presenta regiones candidatas, su medida de permanencia disminuye, si ha llegado a cero, el objeto en seguimiento se considerará como perdido y se borrará de la lista de objetos en seguimiento.

4. Si alguna región nueva no es propuesta como candidata de algún objeto en seguimiento, se agregará a la lista con un nuevo ID y con una medida de permanencia con un valor medio.

El algoritmo 5 muestra la implementación de estas posibilidades para el seguimiento de objetos.

Los elementos $Pla(o)$ y $PlaM(o)$ son llenados al momento de volverse un candidato para las regiones con la función *LlenaPlantillas(objeto)*.

Al termino de este proceso se conocerán todas las posibles regiones u objetos candidatos para cada objeto en seguimiento, cada una de las regiones será evaluada y se seleccionará la de mayor similitud al objeto en seguimiento.

Cálculo de la medida de similitud

Para cada una de las regiones candidatas de los objetos en seguimiento, se determina la mejor similitud entre regiones por medio de una medida de correlación, esta medida será la suma de las diferencias absolutas de cada uno de los píxeles de la imagen I en el tiempo n con los píxeles de cada una de las imágenes candidatas I en el tiempo $n+1$, la formula para esta medida es:

$$C(r) = \sum_{x \in R} \frac{W(x) \mid I_n(x) - I_{n+1}(x) \mid}{\mid\mid W \mid\mid} \tag{3.2.11}$$

```
1  para cada r ∈ Lr  Y para todo nuevo objeto o hacer
2      si P1(r) ≤ Ce(o) ≤ P2(r) entonces
3          ID(o) ← ID(r)
4          LlenaPlantillas(o)
5          P(r) ← P(r) + 1
6      sino
7          P(r) ← P(r) − 1
8          si P(r) = 0 entonces
9              Borrar(r)
10         fin si
11     fin si
12 fin para cada
13 para cada objeto nuevo o hacer
14     si ID(o) = ∅ entonces
15         NRegiones ← NRegiones + 1
16         ID(o) ← NRegiones
17         P(o) ← 2
18     fin si
19 fin para cada
```

Algoritmo 5: Asociación de regiones

La constante de normalización, representada por $|| W ||$ se calcula de la siguiente manera:

$$|| W ||= \sum_{x \in R} W(x) \tag{3.2.12}$$

La función $W(x)$ se calcula como sigue:

$$W(x) = \frac{1}{2} + \frac{1}{2}(1 - \frac{h(x)}{h_{max}}) \tag{3.2.13}$$

La función $h(x)$ estará definida como la distancia euclidiana entre el centroide $Ce(r_n)$ en el tiempo n y el punto x_{n+1} de la región en el tiempo $n+1$.

La diferencia entre los píxeles de mayor similitud será cercana a cero por lo que la región con la menor medida de similitud será la mejor candidata, esto eso:

$$r_{n+1} = \min_{r_{n+1} \in Candidatas} C(r_{n+1}) \tag{3.2.14}$$

Actualización de datos La velocidad del objeto será actualizada, y será simplemente la distancia euclidiana entre los centroides de las regiones en los tiempos n y $n+1$.

$$V(r) = \sqrt{(cx_n - cx_{n+1})^2 + (cy_n - cy_{n+1})^2} \tag{3.2.15}$$

Tiempo de procesamiento Un punto importante en este tipo de aplicaciones es la respuesta en tiempo real. Algunas medidas a tomar en cuenta para un procesamiento más rápido de los datos son:

- Para la medida de permanencia es necesario un límite debido a que irá incrementando con el simple hecho de encontrar candidatos para el objeto en seguimiento, si esto sucediera, al momento de salir de escena se mantendría por demasiado tiempo la búsqueda de regiones candidatas para el objeto, esto afectaría el tiempo de procesamiento de otros objetos por lo que se ha optado por un valor máximo de $k = 5$ para esta medida, de modo que se mantendrá en seguimiento a un objeto a los más en 5 imágenes consecutivas, de no encontrar regiones candidatas, el objeto será eliminado de la lista de objetos en seguimiento, siendo considerado como perdido o una falsa alarma.

- Para reducir el número de operaciones en la función de similitud solo se procesarán aquellos píxeles donde se presente movimiento, ya que en donde no existe movimiento no debe existir gran diferencia, para este fin se utilizará la *plantilla de movimiento* $PlaM(r)$ y donde no se presente movimiento, el valor de la función $W(x)$ será 0. $W(x)$ es modificada de la siguiente manera:

$$W(x) = \begin{cases} \frac{1}{2} + \frac{1}{2}(1 - \frac{h(x)}{h_{max}}) & \text{Si } PlaM(x) = \textit{en movimiento} \\ 0 & \text{en otro caso} \end{cases} \tag{3.2.16}$$

- Otra técnica utilizada con frecuencia es el sub-procesamiento de la imagen, esto quiere decir que solo parte de las plantillas de los objetos serán procesadas, esto se hace con el fin de tener un tiempo de procesamiento constante para todos los objetos, la idea es dividir a la imagen plantilla tantas veces sea necesario hasta obtener el tamaño deseado (*umbral*), y se harán *saltos* entre píxeles para simular que el objeto es del mismo tamaño que los demás, en la figura 3.9 se muestra la utilidad de esta técnica.

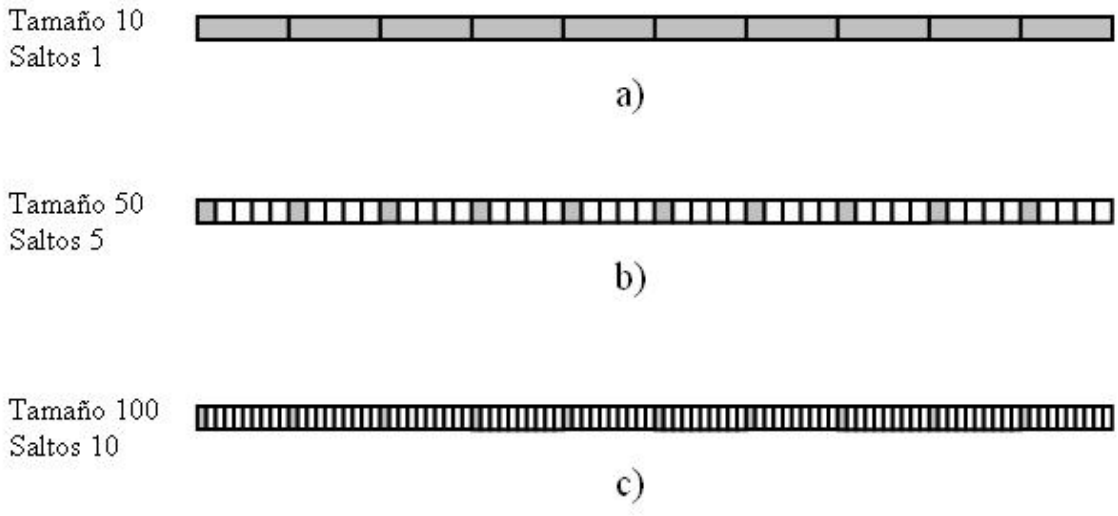

Figura 3.9: Procesamiento de una muestra de datos de diferentes tamaños en un tiempo constante. En (a) se tiene una muestra de diez datos, se utiliza un tiempo de procesador para cada dato, en (b) se tienen cinco veces más datos que en (a), se considera solo un conjunto de datos de la muestra de modo que el tiempo de procesamiento sea el mismo que para (a), en (c) se tienen el doble de datos que en (b), de igual manera se consideran solo ciertos datos de la muestra para procesar el mismo número que en las anteriores.

Eliminación de falsas alarmas

Las ramas de los árboles que se mueven por el viento o el ruido introducido por la señal de video son consideradas falsas alarmas en el contexto de seguimiento de objetos, un forma de distinguir entre los verdaderos objetos y las falsas alarmas es por la permanencia que muestran los verdaderos objetos en las imágenes, esta característica es tomada en cuenta al momento de asociar las nuevas regiones con las existentes en la lista de objetos en seguimiento, por ello el uso de la variable $P(r)$.

En la figura 3.10 se presenta el diagrama de flujo del algoritmo de seguimiento de objetos.

3.2.3. Clasificación de objetos

La ultima etapa antes del reconocimiento de comportamientos será la fase de clasificación de cada objeto.

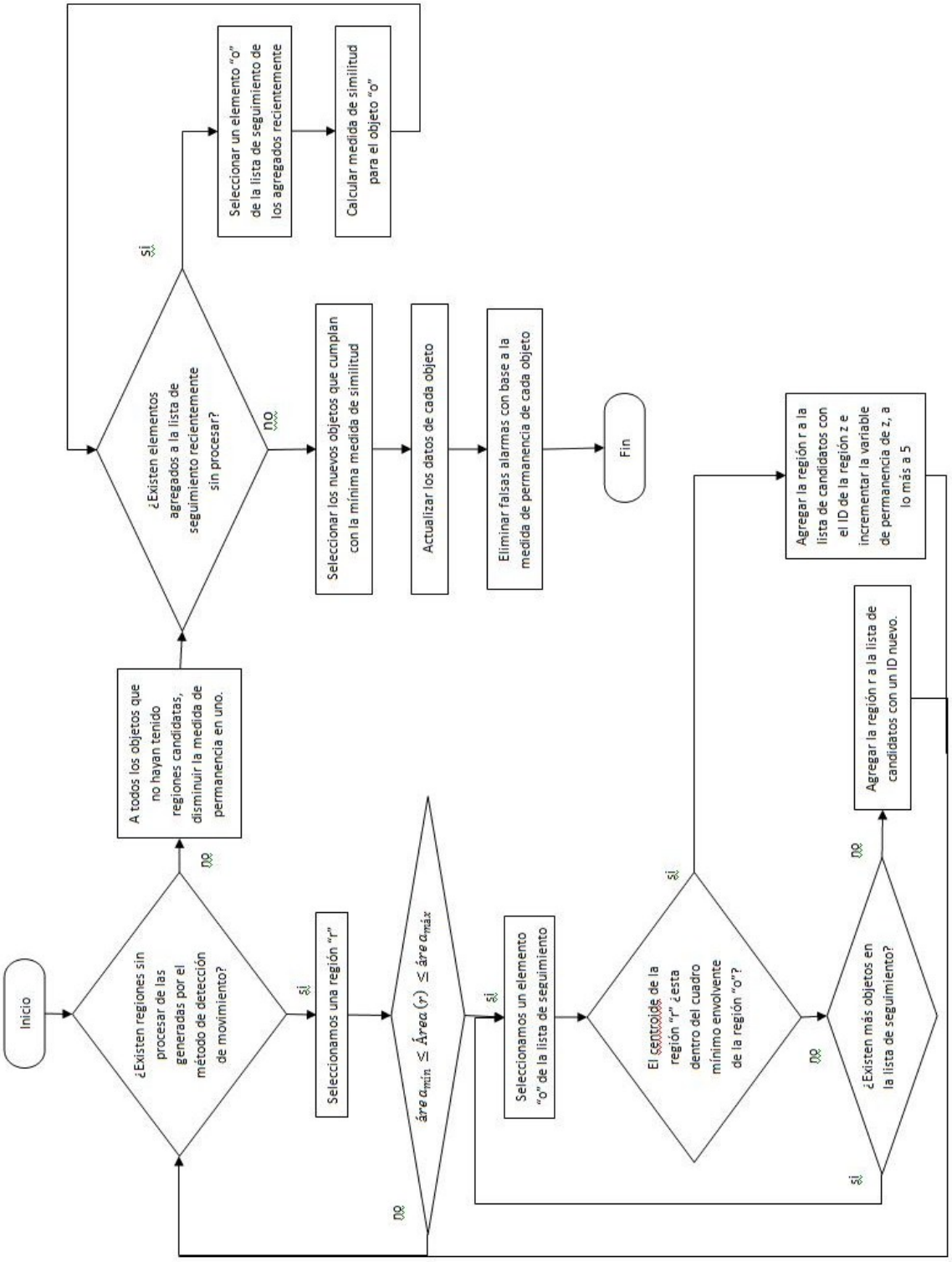

Figura 3.10: Diagrama de flujo del algoritmo de seguimiento de objetos.

En una tarea de clasificación se presentan dos etapas: la de entrenamiento y la de pruebas, cada una hace uso de un conjunto de datos obtenidos de los objetos de interés, estos datos son conocidos como *características* o atributos del objeto. En la etapa de entrenamiento cada conjunto de características se encuentra acompañado de una etiqueta o *clase*, la meta de la clasificación es asignar una *clase* a partir de sólo un conjunto de características dadas en la etapa de pruebas.

El método de seguimiento utilizado obtiene ciertas propiedades de los objetos, estas son de utilidad en esta etapa. La clasificación final de cada objeto se obtendrá con la participación de un método de clasificación, en este caso *SVM*, y un algoritmo de votación para asegurar consistencia en la clasificación de los objetos.

Las posibles clases a las que pertenecerá un objeto son dos, estas son las siguientes:

1. Personas: Esta clase es necesaria para ciertos comportamientos que se desean identificar, algunos de ellos como *platica entre personas* necesita de al menos dos objetos de esta clase.
2. Autos: Para el comportamiento *persona atropellada* es necesaria la presencia de un objeto de esta clase.

Los pasos necesarios para obtener la clasificación de un objeto son los siguientes:

- Etapa de entrenamiento
    1. Obtención de las características de los objetos para entrenamiento.
    2. Creación del modelo el cuál será utilizado por el método *SVM* para clasificación.
- Etapa de pruebas
    1. Extracción de características de los objetos.
    2. Procesamiento de los datos por medio de *SVM* utilizando el modelo obtenido en la etapa de entrenamiento.
    3. Selección de un clase con base al mayor número de votos.

Extracción de características

Las características que se utilizarán para la clasificación del objeto serán cinco, estas son las siguientes:

1. Ancho del objeto
2. Alto del objeto
3. Área del objeto
4. Relación de aspecto, ancho respecto al alto
5. Elongación

Algunas de estas características son calculadas al momento de generar regiones y otras al ser agregadas a la lista de regiones candidatas o de objetos en seguimiento.

Para calcular estas características es necesario conocer los puntos del cuadro mínimo envolvente del objeto, $P_1(r) = (x_1, y_1)$ y $P_2(r) = (x_2, y_2)$ los cuales fueron calculados en la sección 3.2.1, suponiendo que r es la región o el objeto de interés, entonces las características son calculadas de la siguiente manera.

Ancho y alto El ancho y el alto del objeto serán los mismos que el del cuadro mínimo envolvente, estas son características básicas de todos los objetos y necesarias para el cálculo de otras más.

$$Ancho(r) = x_2 - x_1 \tag{3.2.17}$$
$$Alto(r) = y_2 - y_1 \tag{3.2.18}$$

Área El área fue calculada en la sección 3.2.2. Al igual que el ancho y el alto es una característica básica y necesaria para el cálculo de otra característica.

Relación de aspecto Esta característica indicará la relación que guardan el ancho y el alto del objeto r. Cuando el ancho del objeto es mayor que el alto el valor resultante será mayor a uno, en caso contrario el resultado estará en el rango $[0, 1]$.

$$RA(r) = \frac{Ancho(r)}{Alto(r)} \tag{3.2.19}$$

Elongación Esta característica mide la similitud del objeto con un circulo, siendo el círculo la figura con menor elongación, $4\pi \approx 12.5$, y esta definida como:

$$\frac{\text{Perímetro}^2}{\text{Área}} \tag{3.2.20}$$

Un área menos elongada (más compacta) tiene un área mayor con mismo perímetro, ver figura 3.11.

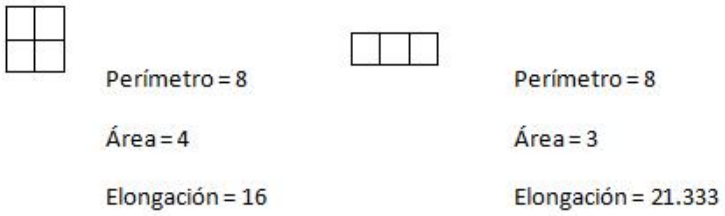

Figura 3.11: Elongación del círculo.

Maquinas de vectores de soporte, *SVM*

La meta de la *SVM* será crear un modelo a partir del cual sea posible asignar una etiqueta o *clase* a un objeto con un conjunto de características asociadas.

Dado un conjunto de pares etiquetados para entrenamiento $(x_i, y_i), i = 1, \ldots, k$ donde $x_i \in R^n$ y $y \in \{1, -1\}^k$, la *SVM* requiere la solución de los siguientes problemas de optimización:

$$\min_{w,b,\xi} \frac{1}{2} w^T w + C \sum_{i=1}^{k} \xi_i \tag{3.2.21}$$

$$dado\ que\ y_i(w^T \phi(x_i) + b) \geq 1 - \xi_i \tag{3.2.22}$$

$$\xi_i \geq 0 \tag{3.2.23}$$

donde C es una variable de tolerancia y es un parámetro que debe ser seleccionado por el usuario.

El vector de entrenamiento en este caso es x_i el cuál es mapeado por la función ϕ a un espacio dimensional mayor donde es posible encontrar una hiperplano de separación lineal con el máximo margen de separación entre clases. La variable y_i representa la pertenencia a la clase, con 1 para indicar pertenencia y -1 en otro caso.

El kernel utilizado será el *RBF* o *funciones de base radial*, esto es debido a que con él no son necesarios muchos parámetros y es posible resolver problemas *no lineales*, este kernel esta definido de la siguiente manera:

$$K(x_i, x_j) = \exp(-\gamma \mid\mid x_i - x_j \mid\mid^2), \gamma \geq 0 \tag{3.2.24}$$

El conjunto de características obtenidas anteriormente serán representadas como un vector de números reales, estas características deben ser escaladas antes de obtener el modelo del *SVM* (etapa de entrenamiento) y al momento de realizar la clasificación de los objetos (etapa de pruebas), el valor de las características se encontrarán en el rango $[a, b]$ y el escalado se realiza de la siguiente manera:

1. Si se trabaja en la etapa de entrenamiento, entonces:
   - Obtener los valores máximos y mínimos de cada característica, estos es calcular: $max(ancho)$, $min(ancho)$, $max(alto)$ y $min(alto)$, y así para todas las características.
2. Si se trabaja en la etapa de pruebas, entonces:
   - Obtener los valores máximos y mínimos generados en la etapa de entrenamiento.
3. Suponiendo que Car es alguna característica y $max(Car)$ y $min(Car)$ son el valor máximo y mínimo respectivos entonces:
   - Si el valor de la características Car es igual al valor mínimo de esa característica, esto es $Car = min(Car)$, entonces $Car = a$.
   - Sino verificar si es igual al valor máximo, es decir $Car = max(Car)$, si es así entonces $Car = b$.
   - Si ninguno de los casos anteriores sucede, utilizar la siguiente formula:

$$Car = a + (b - a) * \frac{Car - a}{b - a} \tag{3.2.25}$$

Como se utiliza el kernel *RBF* es necesario el cálculo de dos parámetros, estos son C de la ecuación 3.2.21 y γ de la ecuación 3.2.24, la correcta selección de estos dos parámetros se reflejará en un alto porcentaje de objetos bien clasificados.

El método utilizado para obtener estos parámetros será dividiendo el conjunto de entrenamiento en dos partes, una parte se utilizará para entrenar la *SVM* y la otra para medir el desempeño de la selección de estos parámetros.

Para mejores resultados se dividirá el conjunto de datos de entrenamiento en k conjuntos y se utiliza el método *k-fold cross-validation*, esto quiere decir que se entrenará a la SVM con $k-1$ conjuntos y se evaluará el desempeño del modelo con el conjunto restante, al final el porcentaje de objetos correctamente clasificados será definitivo para la selección de los parámetros C y γ. Se recomienda hacer la selección de C y γ con incrementos en potencia de 2, esto es $C = 2^{-5}, 2^{-3}, \ldots, 2^{15}, \gamma = 2^{-15}, 2^{-13}, \ldots, 2^{3}$.

Clasificación de objetos

Una vez obtenida una clase por la SVM se agrega un voto a esa clase, y el objeto será clasificado como parte de la clase que cuente con mayor número de votos, este método tiene como objetivo mantener una clasificación del objeto y no cambiarla al presentarse cambios repentinos en la forma del objeto.

$$Clase(r) = \text{máx}\{votos_r(Persona), votos_r(Auto)\} \tag{3.2.26}$$

A cada objeto se le asigna una etiqueta indicando la clase a la que pertenece, esta etiqueta es utilizada en la siguiente etapa para identificar las relaciones e interacciones entre objetos.

En la figura 3.12 se muestra el diagrama de flujo del método de clasificación de objetos, se divide en: etapa de entrenamiento y etapa de evaluación o prueba.

3.3. Módulo de interpretación

En esta etapa se hace uso del conjunto de objetos clasificados para reconocer interacción entre ellos, una de las tareas más importantes de este trabajo es la identificación de comportamientos, para ello se utilizan Modelos Ocultos de Markov de los que se hablo generalmente en 2.2.5.

Para la identificación de los comportamientos propuestos en este trabajo es necesario que los objetos se encuentren en ciertos *estados*, por ejemplo, para reconocer el comportamientos *personas platicando* es necesario que las personas se encuentren en estado *detenidas* y a una cierta distancia, por lo que primero se hará la identificación de lo que se han llamado *comportamientos básicos*, posteriormente se obtiene el conjunto de observaciones que se presentan en escena, este conjunto es evaluado por todos los modelos, los cuales representan a cada uno de los comportamientos, al obtener el resultado de la evaluación se conocerá el comportamiento de los objetos.

Las partes que conforman a esta sección son las siguientes:

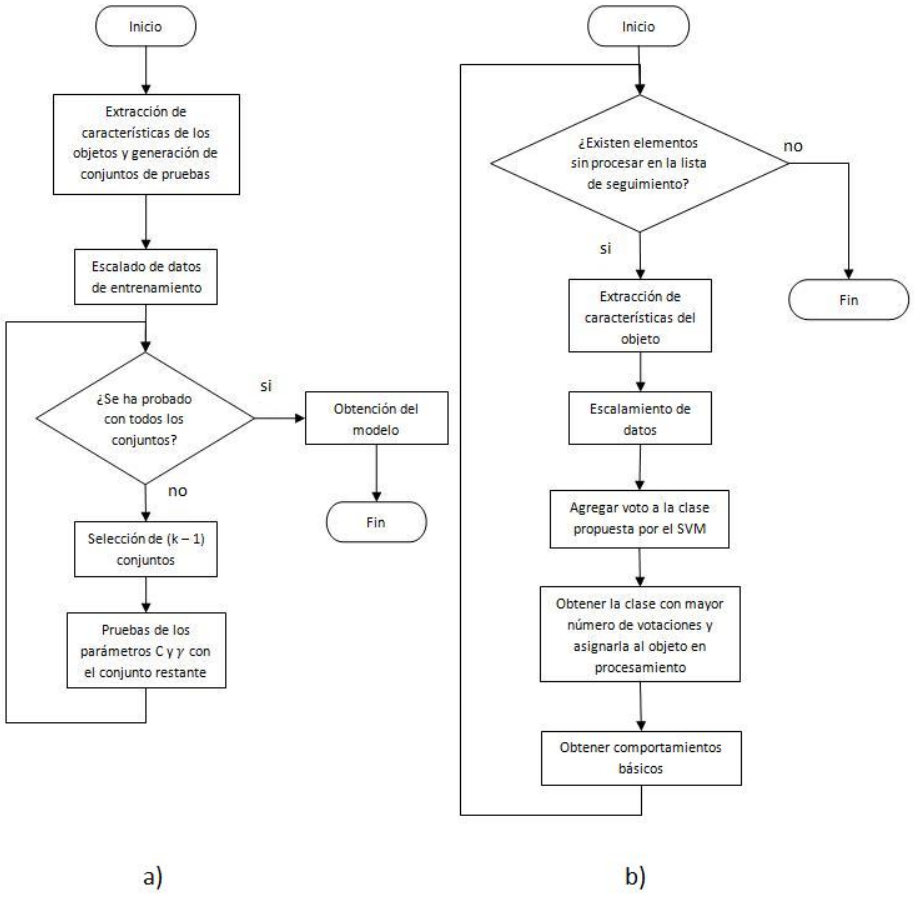

Figura 3.12: Diagrama de flujo del algoritmo de clasificación de objetos. En (a) se muestra la etapa de entrenamiento y en (b) la etapa de evaluación.

1. Reconocimiento de comportamientos básicos
2. Reconocimiento con Modelos Ocultos de Markov

El diagrama general de este módulo se presenta en la figura 3.13.

3.3.1. Reconocimiento de comportamientos básicos

Con la clasificación de los objetos es posible reconocer algunos comportamientos básicos de estos, por ejemplo saber si está detenido o en movimiento, para identificar este comportamiento se utiliza la velocidad del objeto, esta fue calculada en la sección 3.2.2, si esta sobrepasa un *Umbral*, entonces está en movimiento.

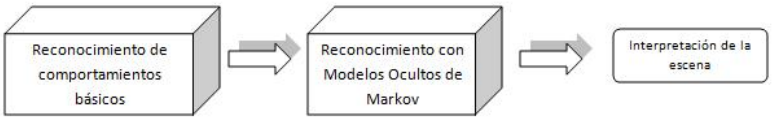

Figura 3.13: Diagrama a bloques del módulo de interpretación, el resultado final del procesamiento es la interpretación de la escena o la secuencia de imágenes.

Otro comportamiento basado también en velocidad es si se mueve rápido o lento, con esto podemos saber sí una persona *corre* o si está *caminando*, para saber esto se utilizará otro *UmbralCaminar* donde si $V(r) > UmbralCaminar$, entonces la persona esta corriendo, de lo contrario se encontrará caminando. El algoritmo 6 resume la identificación de estos comportamientos.

```
1  para cada p: objeto clasificado como persona hacer
2      si V(p) ≥ Umbral entonces
3          si V(p) ≥ UmbralCaminar entonces
4              p ← Corriendo
5          sino
6              p ← Caminando
7          fin si
8      sino
9          p ← Detenido
10     fin si
11 fin para cada
12 para cada a: objeto clasificado como auto hacer
13     si V(a) ≥ Umbral entonces
14         p ← Avanzando
15     sino
16         p ← Detenido
17     fin si
18 fin para cada
```

Algoritmo 6: Identificación de comportamientos básicos.

3.3.2. Reconocimiento con Modelos Ocultos de Markov

Para lograr el reconocimiento de comportamientos con Modelos Ocultos de Markov es necesario generar un modelo por cada comportamiento que se tiene, recordando

lo mencionado en 3.3.2 es necesario contar con un conjunto de observaciones, este conjunto será evaluado por cada uno de los modelos dando como resultado un conjunto de probabilidades, cada una asociada a un modelo, el comportamiento relacionado con el modelo de más alta probabilidad será el reconocido en la escena.

Se deben generar tres modelos, uno para cada comportamiento:

1. Personas platican
2. Persona atropellada
3. Robo a personas

Estos modelos tienen un conjunto de estados asociados que no son conocidos (ocultos), pero por cada transición entre estados deben generar observaciones, estas observaciones se utilizarán como base para construir los modelos de modo que en cada uno se maximize la probabilidad de la secuencia de observaciones asociada a cada comportamiento.

En las siguientes secciones se analizan las condiciones que debe cumplir la escena o la secuencia de imágenes para identificar los comportamientos.

Observaciones para el comportamiento *"Personas platican"*

Para que exista este comportamiento es necesario que existan almenos dos personas por lo que es necesario contar el número de personas que exísten en escena, como se cuenta con una lista de objetos clasificados esta tarea no presenta mayor problema.

Una vez identificadas a las "personas" en la lista de objetos se debe verificar que se encuentren "detenidas", es decir que la velocidad del objeto sea menor a un *Umbral*, este resultado fue obtenido en 3.3.1 por lo que sólo hay que verificar esta condición. Para agilizar el tiempo de ejecución es posible verificar las dos condiciones al mismo tiempo, es decir que el objeto este clasificado como "persona" y que este "detenida".

Al término de la identificación de "personas detenidas" se tomarán pares de objetos de esta lista y se comprobará si exíste interacción entre ellos. Para decir que dos objetos interactuan deben encontrarse a cierta distancia, esta distancia se mide desde los cuatro puntos mostrados en la figura 3.14. Si la distancia entre alguno de los puntos del primer y el segundo objeto se encuentra dentro de la *distancia de interacción*, se dice que los objetos interactuan, en este caso la primera y segunda persona están platicando.

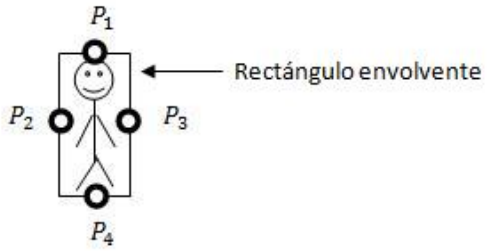

Figura 3.14: Puntos utilizados para identificar interacción entre objetos.

Para obtener la posición de cada punto se utilizan las coordenadas que definen al rectangulo envolvente del objeto, suponiendo que o es el objeto procesado, entonces:

$$\begin{aligned}
(x_1, y_1) &= P_1(o), \\
(x_2, y_2) &= P_2(o), \\
P_1 &= (\frac{x_1 + x_2}{2}, y_1), \\
P_2 &= (x_1, \frac{y_1 + y_2}{2}), \\
P_3 &= (x_2, \frac{y_1 + y_2}{2}), \\
P_4 &= (\frac{x_1 + x_2}{2}, y_2)
\end{aligned} \tag{3.3.1}$$

Como se puede observar es necesario que la escena cumpla con ciertas condiciones, a continuación se resumen las que debe presentar este comportamiento.

1. Es necesario que existan almenos dos personas
2. Almenos dos personas deben estar "detenidas"
3. La distancia entre las personas detenidas debe encontrarse en el rango de la *distancia de interacción*

Observaciones para el comportamiento *"Persona atropellada"*

Para este comportamiento se requiere de una personas y un automóvil en movimiento, del mismo modo que el comportamiento anterior estos objetos se buscan dentro de la lista de objetos clasificados.

La personas puede encontrarse "detenida", "caminando" o "corriendo" por lo que no es necesario verificar su velocidad, el automóvil necesita estar en movimiento, este dato se obtiene del resultado en 3.3.1.

El último paso es verificar la interacción entre un objeto tipo automóvil en movimiento y otro tipo persona, si exíste interacción es porque el automóvil esta atropellando a la persona.

Las condiciones que debe cumplir este comportamiento se resumen a continuación:

1. Debe existir almenos una personas sin importar si esta detenida o no
2. Debe existir un automóvil en movimiento
3. La distancia entre la persona y el automóvil en movimiento debe encontrarse dentro del rango de la *distancia de interacción*

Observaciones para el comportamiento *"Robo a personas"*

Para este comportamiento es necesario que almenos dos personas se encuentren cerca y en movimiento, este comportamiento debe presentarse durante un lapso de tiempo, la persona "delincuente" debe acercarse a la otra persona para robar de modo que "invada su espacio", este acto se verá reflejado como la desaparición de alguna de las dos personas, después el "delincuente" se alejará y aparecerá nuevamente la otra persona. En la figura 3.15 se muestran estas etapas.

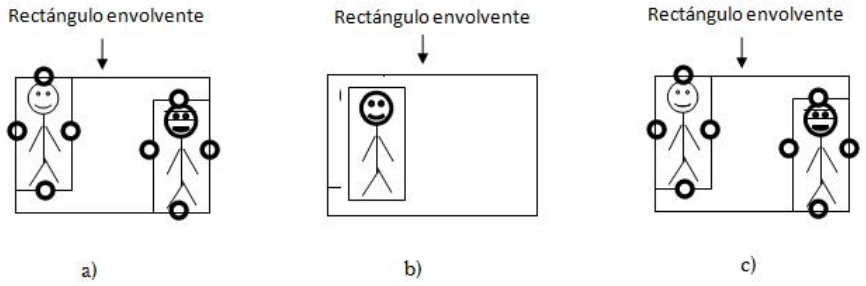

Figura 3.15: Etapas del comportamiento *Robo a personas*, en (a) se calcula el rectángulo que encierra a ambas personas y se espera por el evento (b) donde sólo una persona se encuentra en el rectángulo, en (c) se presenta el momento en que las personas se separan.

Este comportamiento necesita de un análisis en el tiempo por lo que se ha divido en tres etapas:

1. Identificación de interacción entre personas y cálculo del rectángulo que incluya a ambas: cambio a análisis en el tiempo
2. Una persona ha desaparecido dentro del rectángulo: en espera por que aparezca la otra persona
3. Las dos personas se encuentran dentro del rectángulo : comportamiento identificado

De la misma manera que en lo comportamientos anteriores se busca por objetos clasificados como "persoonas", una vez encontrados dos elementos se verifica que cumplan con la "distancia de interacción", si la cumplen se obtiene el rectángulo mínimo que encierra a ambas personas.

Las personas deben permanecer dentro del rectángulo calculado y en algún momento dentro de este sólo exístira una persona, este evento puede darse en distintos intervalos de tiempo por lo que se permanece en espera. Para identificar esto se verifican los objetos clasificados como personas en las imágenes siguientes, si el centroide de sólo uno de estos objetos se encuentra dentro del rectángulo entonces se pasa a la etapa de búsqueda de dos objetos persona dentro del rectángulo.

Para verificar la existencia de dos personas simplemente se cuenta el número de personas dentro del rectángulo.

A continuación se resumen las condiciones que debe cumplir este comportamiento:

1. Deben presentarse almenos dos personas
2. Las personas pueden encontrarse en cualquier estado pero deben estar interactuando
3. Al identificar interacción se realiza un análisis en el tiempo en donde se espera por la "desaparición" de una persona
4. Al terminar el análisis en el tiempo deben aparecer dos personas y deben interactuar

Observaciones utilizadas para los Modelos Ocultos de Markov

De las condiciones necesarias para los comportamientos es posible generar una lista de observaciones, cada observación es identificada por un número, las observaciones propuestas son las siguientes:

0, existe una persona en escena

1, existen al menos dos personas en escena

2, existe almenos un auto en escena

3, las personas A y B interactuan

4, el auto y la persona interactuan

5, la persona A está detenida

6, la persona B está detenida

7, el auto está en movimiento

8, el auto está detenido

9, es necesario un análisis en el tiempo

Para que los Modelos Ocultos de Markov identifiquen los comportamientos se debe presentar una secuencia de observaciones (condiciones), los comportamientos codificados como observaciones se muestran a continuación:

1. Personas platican: 0, 5, 1, 6, 3
2. Persona atropellada:
   a) Modo 1: 0, 2, 7, 4
   b) Modo 2: 2, 7, 0, 4
3. Robo personas: 0, 1, 3, 9, 3

Generación de los Modelos Oculto de Markov a partir de las observaciones

Para generar los modelos ocultos de Markov se utilizó la herramienta "Java HMM Pak v1.2" la cual se encuetra disponible en `http://www.public.asu.edu/~tmcdani/hmm.htm`, y el manual que acompaña a esta herramienta [17].

Con los conjuntos de observaciones para los comportamientos se generar un modelo oculto de Markov, para la generación de este modelo se utiliza el algoritmo de agrupamiento "k-means", los pasos se muestran a continuación:

1. Se genera un archivo con todos los comportamientos codificados en forma de observaciones, cada comportamiento se conocerá como un *símbolo*, por ejemplo para el comportamiento personas platican: 0, 5, 1, 6, 3, esta secuencia se considera como un *símbolo*, en el archivo se tiene el mismo número de secuencias o símbolos para cada comportamiento. Se seleccionan cuatro semillas iniciales, en este caso el *símbolo* que representa a cada comportamiento.

2. Se agrupan todos los símbolos de modo que la distancia euclidiana entre la semilla y los elementos de su grupo sea mínima.

3. Una vez asignado un grupo a cada símbolo se recalculan las semillas de cada uno de los grupos.

4. Se calcula el error entre la nueva semilla y los elementos de su grupo.

5. Si el error es mayor a un *Umbral* se repiten los pasos 2 al 5, de lo contrario ir al paso 6.

6. Se genera el Modelo Ocutlo de Markov utilizando el agrupamiento final.

La generación del Modelo Oculto de Markov se detalla en [17] y en el material que acompaña a la herramienta, al final del agrupamiento se cuenta con una matriz de clasificación C, de tamaño w x T, donde w es el número de *símbolos* o comportamientos, en esta caso cuatro y T el número máximo de elementos que conforman a los *símbolos*, es decir cinco.

1. N: número de estados, este valor es un parámetro de entrada, haciendo varias pruebas y observando las probabilidad obtenidas se decidio utilizar cinco estados.

2. S: el conjunto de estados $S = \{S_1, S_2, \ldots, S_N\}$

3. M: número de observaciones por estado, claramente diez, hay que notar que estas observaciones no entran "tal cual" a los modelos sino deben pasar por un método de codificación que utiliza el paquete, tanto en la etapa de entrenamiento como en pruebas.

4. V: el alfabeto de entrada esta compuestos por la secuencia de números del 0 al 9, al momento de la codificación se convertirán en otros datos.

5. A: la matriz de probabilidad de transiciones se calcula con ayuda de la matriz de clasificación obtenida, la idea general para obtener el valor para $A_{i,j}$ es la siguiente:

   a) Se calcula el número total de elementos con clasificación i, es decir $total(i) = \sum_{g=1}^{w} \sum_{h=1}^{T} C_{g,h}$, donde $C_{g,h} = i$.

   b) Se calcula el número de elementos que cambian de clasificación i a j, es decir $total(i,j) = \sum_{g=1}^{w} \sum_{h=1}^{T-1} C_{g,h}$, donde $C_{g,h} = i$ y $C_{g,h+1} = j$.

   c) Se obtiene el cociente $A_{i,j} = \frac{total(i,j)}{total(i)}$.

6. B: matriz de probabilidad de observaciones de tamaño N x M, para obtener los valores de estos vectores se utiliza la matriz de clasificación C y la matriz de datos de entrada O de igual tamaño que C, el cálculo del valor $B_{i,j}$ de manera general es de la siguiente manera:

   a) Contar el número total de observaciones j producidas en cualquier estado, es decir $total(j) = \sum_{g=1}^{p} \sum_{h=1}^{q} O_{g,h}$, donde $O_{g,h} = j$.

   b) Contar el número de observaciones j producidas desde el estado i, es decir $total(i,j) = \sum_{g=1}^{w} \sum_{h=1}^{T} C_{g,h}$ y $O_{g,h}$, donde $C_{g,h} = i$ y $O_{g,h} = j$.

   c) Calcular el valor obtenido para $B_{i,j} = \frac{total(i,j)}{total(j)}$.

7. π: vector de probabilidades iniciales, para obtener los valores para π_i, esto es la probabilidad de iniciar en el estado i se calcula el número total de transiciones de cualquier estado al estado i y se divide entre el número de *símbolos* o comportamientos w, es decir $\frac{A_{j,i}}{w}$.

El modelo generado se utilizará como base para generar los modelos correspondientes a cada comportamiento, este modelo tiene la misma probabilidad de generar o reconocer las observaciones de cualquier comportamiento por lo que es necesario maximizar las probabilidades para que reconozca sólo uno de ellos.

Como se recordará, en la sección se mencionó que uno de los problemas para trabajar con los modelos ocultos de Markov es la posibilidad de maximizar la probabilidad de una observación O dado un modelo λ, esto es: maximizar $P(O|\lambda)$, el algoritmo de *Baum-Welch* es utilizado para este fin, y esto es precisamente lo que se hace para generar los modelos de cada comportamiento.

Se toma el modelo generado con ayuda de "k-means", se generan cada una de las secuencias de observaciones de cada comportamiento, para cada comportamiento se ingresa el modelo y el conjunto de observaciones que lo definen, utilizando el algoritmo *Baum-Welch* se obtiene el modelo oculto de Markov maximizado para la observación dada, este proceso se repite para cada comportamiento, para detalles consultar [17].

Reconocimiento de comportamientos con modelo ocultos de Markov

Con los modelos generados se realiza un análisis de la escena buscando por los elementos indicados en las observaciones, si alguna se cumple se ingresa el número de observación a una lista, la lista de observaciones es procesada por cada uno de los modelos con ayuda del algoritmo *Forward*, el cuál resuelve el primer problema de los modelos ocultos de Markov, donde dado un conjunto de observaciones O y un modelo λ obtiene el valor para $P(O|\lambda)$.

Para decidir que comportamiento se presenta en escena se obtiene la probabilidad máxima de las generadas por los modelos, si la máxima probabilidad sobrepasa un *Umbral* entonces se marca como reconocido.

Capítulo 4

Resultados

En este capítulo se presentan los experimentos y resultados obtenidos con la implementación de las técnicas mencionadas en el capítulo anterior. El sistema fue desarrollo en *C++ Builder 6*, para la captura de video se utilizó una cámara *EVI-D100P Pal PTZ* marca *SONY* y un Frame Grabber *Arvoo Picasso 2SQ* marca *Picasso*. El tamaño de las imágenes utilizadas es de 320 x 240, las secuencias de imágenes contienen en promedio 4000 imágenes.

4.1. Detección de movimiento

Para la detección de movimiento se utilizaron algunas secuencias donde se presentan objetos que entran y salen de escena, algunos de ellos se detienen y después vuelven a moverse.

Los primeros resultados corresponden a detección de movimiento sin eliminación de "ruido", después se agregaron las operaciones morfológicas para reducirlo y a continuación se presentan los resultados que corresponden a objetos que se detienen en escena por cierto tiempo y después vuelven a moverse.

Para obtener el fondo que se adaptará con el tiempo se utilizan las primeras 25 imágenes, este fondo esta formado por el valor promedio de los píxeles del total de las imágenes, algunos fondos obtenidos se presentan en la figura 4.1.

Utilizando el fondo adaptable se realiza la detección de movimiento, el primer escenario para la evaluación no contiene objetos que puedan generar "ruido", el segundo escenario presenta árboles donde se mueven sus ramas debido a la intervención del viento, estos movimientos son detectados como "ruido", el valor utilizado para *sensibilidad* es 1.5, en la parte superior se presentan los resultados *sin* la operación de apertura, la parte izquierda de la figura 4.2 muestra la imagen

Figura 4.1: En (a) se presenta una imagen de la secuencia y en (b) el fondo adaptable obtenido de la secuencia.

original de la secuencia, en el centro el movimiento detectado en forma de manchas y a la derecha se resalta el movimiento detectado sobre la imagen.

Para eliminar el "ruido" detectado como movimiento se utiliza la apertura, en la parte inferior de la figura 4.2 se muestran los resultados *con* apertura los cuales son visiblemente mejores, además de elimiar el "ruido" esta operación ayuda a obtener figuras o manchas de objetos mejor definidas y se logra conectar regiones que corresponden a un mismo objeto.

Los resultados obtenidos con objetos que se detienen en escena se presentan en la figura 4.3, donde los objetos o partes del objeto que se mueven se presentan en color verde o azul, el verde representa las secciones del objeto que no se han detenido y el azul corresponde a los fragmentos que se han detenido, los fragmentos azules o en

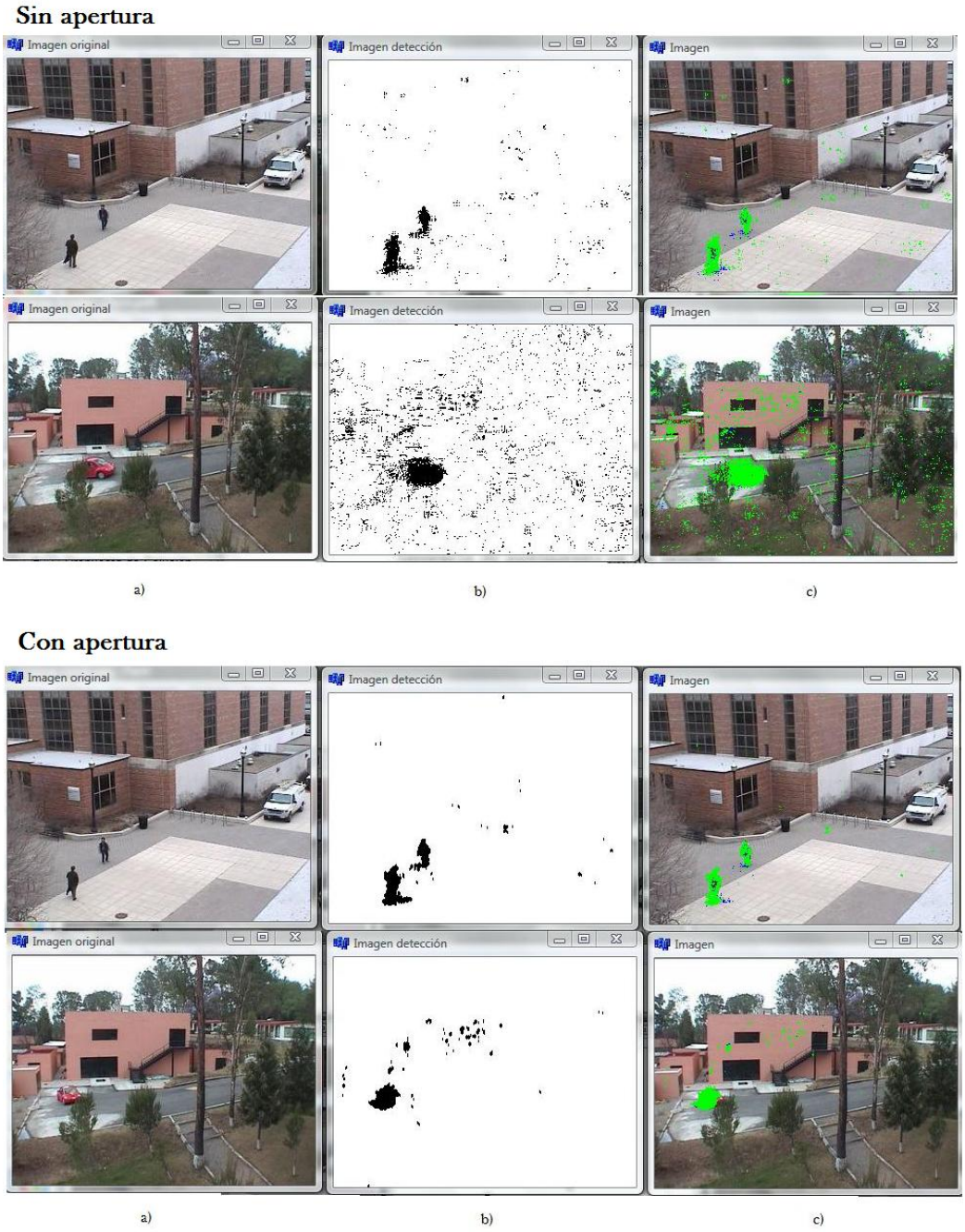

Figura 4.2: Resultados de detección de movimiento *sin* y *con* la operación morfológica de apertura, la columna (a) presenta la imagen original de la secuencia, en (b) se muestra el movimiento detectado en forma de manchas, en (c) se colocan las manchas en la imagen original mostrando el movimiento en color verde para objetos que no se detienen y en azul para los que se detienen en escena.

verdes no son considerados en la actualización del fondo adaptable, este se muestra en la misma figura (4.3).

4.2. Seguimiento de objetos

Como se mencionó en el capítulo anterior los resultados obtenidos por el algoritmo de detección de movimiento, las manchas de movimiento o regiones, son utilizados por el método de seguimiento de objetos, para evaluar el desempeño de este se procedio de la siguiente manera:

1. El método de seguimiento asigna una etiqueta numérica a cada objeto detectado, esta etiqueta debería conservarla durante toda su permanencia en escena por lo que se toma nota de la etiqueta asignada por primera vez a algunos objetos utilizados como referencia.

2. Se cuenta el número de veces que cambia la etiqueta.

3. Los objetos utilizados como referencia deben ser los de interés para el reconocimiento de comportamientos, esto es:

   a) Autos en movimiento

   b) Autos que se han estacionado

   c) Personas en movimiento

   d) Personas que se han detenido

Para la primera evaluación se toma una secuencia donde se presentan dos personas que se acercan y platican, después de un tiempo se separan y salen de escena, algunos momentos obtenidos de la secuencia se presentan en la figura 4.4.

En esta secuencia se observa que una de las personas fue "perdida" por el algoritmo, esto se debe al cambio de forma del objeto, el algoritmo trabaja con una medida de correlación, si el objeto cambia de forma, es decir, se voltea o pasa detrás de algún objeto el algoritmo de seguimiento lo perderá. Sin embargo es posible reconocer los comportamientos ya que otra de las tareas del algoritmo de seguimiento es obtener características importantes de los objetos las cuales son utilizadas para la clasificación y el reconocimiento.

En la siguiente secuencia utilizada para evaluación entran dos autos a escena, las personas que los manejan bajan de ellos y salen de escena, algunas personas caminan por el área y después de un tiempo uno de los autos sale de escena.

Figura 4.3: Resultados para objetos que se detienen en escena, en la columna (a) se presenta la imagen original, en la columna (b) se presenta el fondo adaptable para cada secuencia, los objetos de color azul o verde no son utilizados para la actualización del fondo, en (c) se presentan los objetos en movimiento, en color verde los que no se detienen y en azul los que se han detenido en escena.

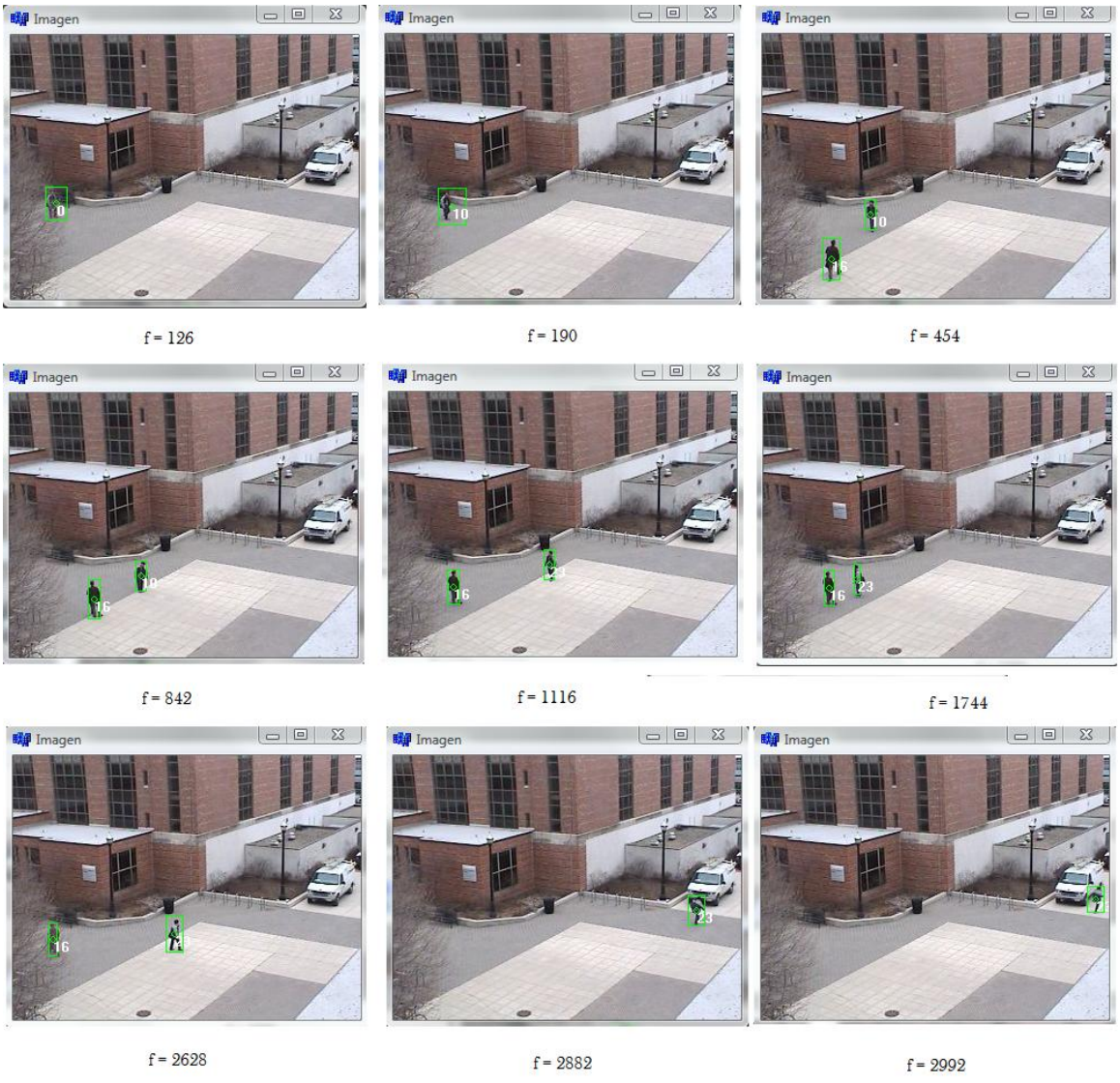

Figura 4.4: Seguimiento de personas, las imágenes muestran que sólo una de las personas detectadas cambia de etiqueta un total de dos veces mientras que la etiqueta de la otra persona no cambia.

Como se puede ver en las imágenes de la figura 4.5, el auto rojo presenta seis cambios de etiqueta, las primeras tres, del frame 90 al 152, se deben a que paso detrás de otros objetos, en este caso árboles, las tres restantes se deben a que mientras el auto se encontraba estacionado algunas personas obstruyeron la visión de él, para el auto azul también se presentaron seis cambios, las primeras dos son por los árboles de las escena, cuando el auto llega pasa por detrás de algunos árboles los cuales no permiten ver el auto completamente, los tres cambios siguientes son debido a que algunas personas pasaron frente a él y obstruyeron la visión de la cámara, el ultimo cambio se debe nuevamente a los árboles.

4.3. Clasificación de objetos

La clasificación de objetos se divide en dos etapas, la de entrenamiento y la de pruebas, para la primera etapa se utilizaron 130 objetos cada uno con su correspondiente clasificación, personas o auto, se utilizó el método *k-fold cross-validation* de modo que se crearon 13 conjuntos de datos, se crea un modelo con $k-1$ conjuntos y se utiliza el conjunto sobrante para pruebas.

A continuación se presenta la manipulación de uno de los trece conjuntos generados, los datos de entrenamiento sin procesar son los siguientes:

```
0 1:12 2:34 3:343 4:0.352941 5:32.758018
0 1:10 2:26 3:207 4:0.384615 5:21.043478
0 1:8 2:20 3:142 4:0.400000 5:13.021127
0 1:10 2:20 3:151 4:0.500000 5:16.556292
0 1:18 2:44 3:458 4:0.409091 5:26.901747
0 1:16 2:32 3:347 4:0.500000 5:28.244957
0 1:16 2:40 3:390 4:0.400000 5:36.923077
0 1:16 2:36 3:348 4:0.444444 5:29.896551
0 1:8 2:24 3:140 4:0.333333 5:27.457144
0 1:12 2:28 3:235 4:0.428571 5:25.889362
1 1:60 2:16 3:583 4:3.750000 5:44.461407
1 1:38 2:10 3:276 4:3.800000 5:28.057972
1 1:50 2:38 3:1332 4:1.315789 5:19.946697
1 1:20 2:12 3:162 4:1.666667 5:27.709877
1 1:100 2:42 3:2756 4:2.380952 5:30.726051
1 1:24 2:10 3:187 4:2.400000 5:30.887701
1 1:66 2:54 3:2298 4:1.222222 5:56.083984
1 1:28 2:16 3:261 4:1.750000 5:61.796936
1 1:52 2:30 3:992 4:1.733333 5:64.525200
```

Figura 4.5: Seguimiento de autos, las imágenes muestran los cambios de etiqueta que presentaron los autos que se encuentran en escena, seis cambios para cada uno.

```
1 1:52 2:44 3:1492 4:1.181818 5:84.943703
```

La primera columa indica la clase a la que pertenece el objeto, cero indica personas y uno es auto, cada una de las siguientes columnas tiene el siguiente formato:

Número de característica:Valor

Las características utilizadas se mencionaron en el capítulo anterior, las etiquetas corresponden con ellas de la siguiente manera:

1. Ancho
2. Alto
3. Área
4. Relación de aspecto = $(Ancho/Alto)$
5. Circularidad = $(Perimetro^2/Area)$

Para los resultados se utilizó el paquete *SVMLib* [18] para la generación de modelos y pruebas, en [19] se presenta una guía para el uso de esta librería, en ella se muestran las mejoreas obtenidas al escalar los datos por lo que se optó por el escalamiento, los nuevos datos se muestran a continuación:

```
0 1:-0.87234 2:0.037037 3:-0.826244 4:-0.931286 5:-0.57753
0 1:-0.914894 2:-0.259259 3:-0.924297 4:-0.915589 5:-0.828281
0 1:-0.957447 2:-0.481481 3:-0.971161 4:-0.907965 5:-1
0 1:-0.914894 2:-0.481481 3:-0.964672 4:-0.858407 5:-0.924329
0 1:-0.744681 2:0.407407 3:-0.743331 4:-0.903459 5:-0.702884
0 1:-0.787234 2:-0.037037 3:-0.82336 4:-0.858407 5:-0.674133
0 1:-0.787234 2:0.259259 3:-0.792358 4:-0.907965 5:-0.488377
0 1:-0.787234 2:0.111111 3:-0.822639 4:-0.885939 5:-0.63878
0 1:-0.957447 2:-0.333333 3:-0.972603 4:-0.941003 5:-0.690996
0 1:-0.87234 2:-0.185185 3:-0.90411 4:-0.893806 5:-0.724554
1 1:0.148936 2:-0.62963 3:-0.653208 4:0.752212 5:-0.327018
1 1:-0.319149 2:-0.851852 3:-0.874549 4:0.776991 5:-0.678135
1 1:-0.0638298 2:0.185185 3:-0.113194 4:-0.454122 5:-0.851758
1 1:-0.702128 2:-0.777778 3:-0.956741 4:-0.280236 5:-0.685586
1 1:1 2:0.333333 3:0.913482 4:0.0737461 5:-0.621025
1 1:-0.617021 2:-0.851852 3:-0.938717 4:0.0831858 5:-0.617564
1 1:0.276596 2:0.777778 3:0.583273 4:-0.500492 5:-0.0782358
1 1:-0.531915 2:-0.62963 3:-0.885364 4:-0.238938 5:0.0440504
1 1:-0.0212766 2:-0.111111 3:-0.358327 4:-0.247198 5:0.102449
1 1:-0.0212766 2:0.407407 3:0.00216294 4:-0.520515 5:0.539509
```

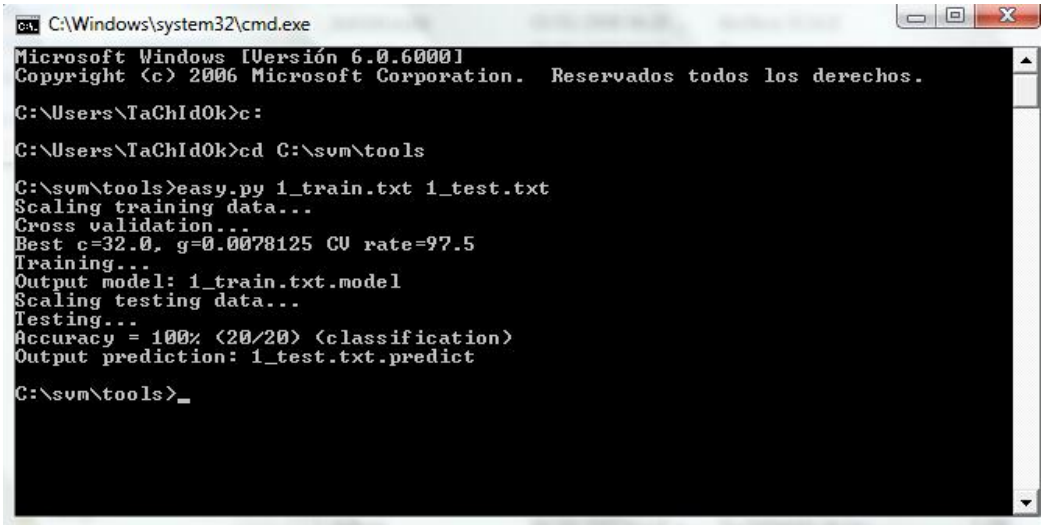

Figura 4.6: Salida del programa de *SVMLib* utilizado para generar el modelo para clasificación y la evaluación con un conjunto de prueba.

Con este conjunto y los $k-1$ restantes se genera un modelo el cual es utilizado para asignar una clase a los objetos del conjunto restante, esta fase es la de prueba, el *SVMLib* cuenta con una herramienta para obtener estos resultados, la salida de este programa se presenta en la figura 4.6.

En la tabla 4.1 se muestran los parámetros para el SVM obtenidos con ayuda de *SVMLib* así como la precisión promedio del modelo usando *k-fold cross-validation*.

En la tabla 4.2 se muestra la matríz de confusión del clasificador, con estos datos se puede calcular la tasa de acierto y de error, estos se obtienen de la siguiente manera; sumar verdaderos positivos, verdaderos negativos, falsos positivos y falsos negativos, es decir $N = VP + VN + FP + FN$, para este clasificador se tiene:

$$N = 125 + 127 + 5 + 3 = 260 \tag{4.3.1}$$

La tasa de acierto y de error se calculan de la siguiente manera:

- Tasa de acierto, $s = \frac{VP+VN}{N} = \frac{125+127}{260} = 0.9692$
- Tasa de error, $\epsilon = 1 - s = 1 - 0.9692 = 0.0308$

Conjunto utilizado para pruebas	Parámetros SVM		Precisión del modelo
1	$C = 32.0$	$\gamma = 0.0078125$	97.5 %
2	$C = 32.0$	$\gamma = 0.0078125$	97.5 %
3	$C = 32.0$	$\gamma = 0.0078125$	97.9167 %
4	$C = 32.0$	$\gamma = 0.0078125$	98.3333 %
5	$C = 32.0$	$\gamma = 0.0078125$	97.5 %
6	$C = 32.0$	$\gamma = 0.0078125$	97.5 %
7	$C = 2048.0$	$\gamma = 0.0078125$	98.3333 %
8	$C = 32.0$	$\gamma = 0.0078125$	97.5 %
9	$C = 2048.0$	$\gamma = 0.03125$	98.75 %
10	$C = 32.0$	$\gamma = 0.0078125$	97.9167 %
11	$C = 0.5$	$\gamma = 0.5$	97.5 %
12	$C = 32.0$	$\gamma = 0.0078125$	97.5 %
13	$C = 32.0$	$\gamma = 0.0078125$	97.5 %
Total	No aplica		97.7884 %

Tabla 4.1: Parámetros para el SVM obtenidos con *SVMLib* y precisión del modelo usando *k-fold cross-validation*.

	Persona	Auto
Persona	125	5
Auto	3	127

Tabla 4.2: Matríz de confusión para el clasificador.

En la figura 4.8 se presentan algunas imágenes donde se encuentran clasificados los objetos en seguimiento, para mostrar la clase de cada objeto se utilizan colores, el color azul claro es para personas y para autos el color violeta, algunos objetos presentan etiquetas donde se muestra algún comportamiento básico identificado.

4.4. Reconocimiento de comportamientos

Como se mencionó en el capítulo anterior el reconocimiento de comportamientos se divide en dos partes, la primera el *reconocimiento de comportamiento básicos* y la segunda *reconocimiento con Modelos Ocultos de Markov*, los resultados de la primera parte se muestran en la figura 4.8, utilizada para mostrar resultados de la parte de

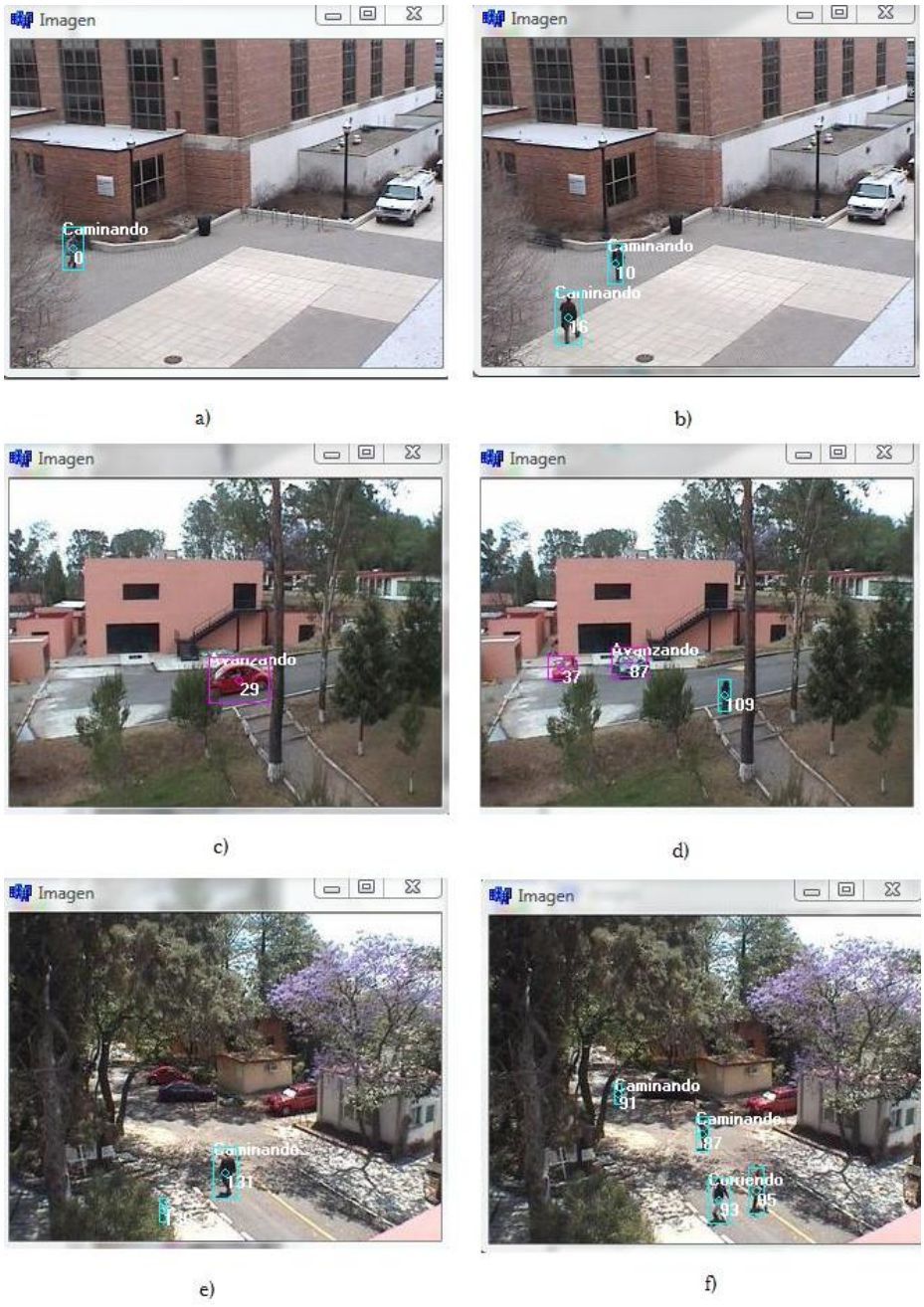

Figura 4.7: Algunas imágenes donde se presentan los objetos en seguimiento clasificados, en (a) se presenta una persona caminando, en (b) dos personas que se acercan para platicar, las dos siguen en movimiento, en (c) se presenta un auto que entra en escena, en (d) un auto estacionado otro estacionandose y una persona detenida, en (e) se muestra una persona caminando además de una falsa alarma clasificada como persona, en (f) se muestran varias personas en movimiento.

clasificación, los elementos que no presentan la etiqueta de comportamiento es porque se encuentran detenidos, los *umbrales* utilizados se muestran en la tabla 4.3.

Tipo de umbral	Valor, (número de píxeles)
Caminar o avanzar	1
Correr	5

Tabla 4.3: Umbrales utilizados para la identificación de comportamientos básicos.

Para la segunda parte, *reconocimiento con Modelo Ocultos de Markov* se creó un archivo con las observaciones que representan a cada uno de los comportamientos a identificar, el archivo se muestra a continuación:

```
1 5 4
0 5 1 6 3
0 2 7 4 4
2 7 0 4 4
0 1 3 9 3
```

La primera fila indica: 1, número de dimensiones de cada elementos de las observaciones, es decir, cuantos elementos constituyen un elemento de la observación; 5, número de elementos de una observación; 4, número de observaciones de entrada para la generación del modelo, las filas restantes son las observaciones que representan a cada comportamiento, el modelo obtenido se presenta en la figura 4.8.

El paquete HMMPak realiza una codificación interna para las observaciones, esta se presenta en la tabla 4.4.

Comportamiento	Original	HMMPak
Personas platican	0 5 1 6 3	0 1 2 3 4
Persona atropellada (modo 1)	0 2 7 4 4	0 5 6 7 7
Persona atropellada (modo 2)	2 7 0 4 4	5 6 0 7 7
Robo a persona	0 1 3 9 3	0 2 4 8 4

Tabla 4.4: Codificación utilizada por HMMPak para las observaciones que representan a los comportamientos.

La descripción de la interfáz desarrollada se presenta en la figura 4.9.

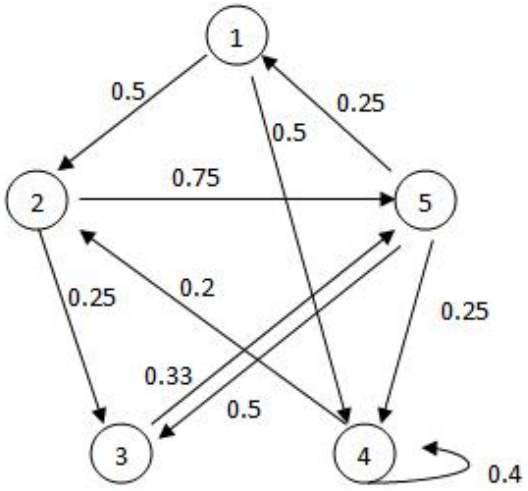

Figura 4.8: Modelo obtenido para la identificación de cada comportamiento.

En el sistema utilizado una vez identificado un comportamiento se enmarcan los objetos involucrados y se presenta la probabilidad obtenida por cada modelo, a continuación se presentan los resultados obtenidos en algunas secuencias de imágenes.

Para el comportamiento *personas platican* se utilizó una secuencia donde sólo se presenta este comportamiento, algunos momentos de la secuencia y las probabilidades obtenidas por los modelos se presentan en las figuras 4.10 y 4.11

Tanto para el comportamiento *persona atropellada* como *robo a persona* se utilizaron secuencias que presentan varios comportamientos, los resultados de detección para estos comportamientos se presentan en las figuras 4.12 y 4.13 para *persona atropellada* y en 4.14 y 4.15 para *robo a persona*.

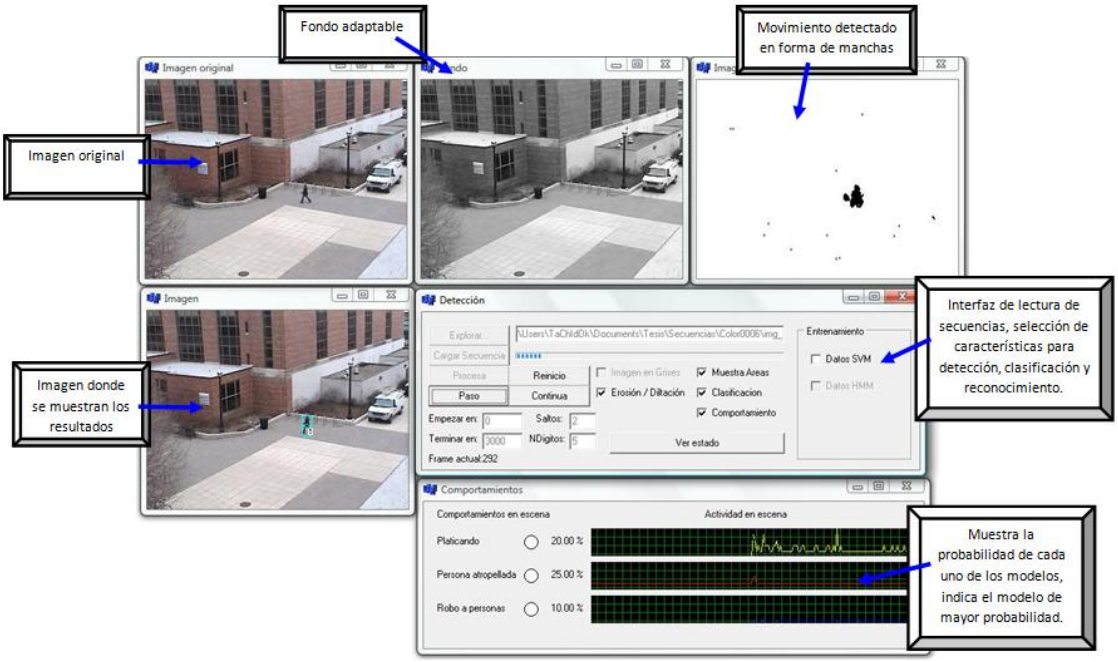

Figura 4.9: Reconocimiento del comportamiento *personas platican*, 1.

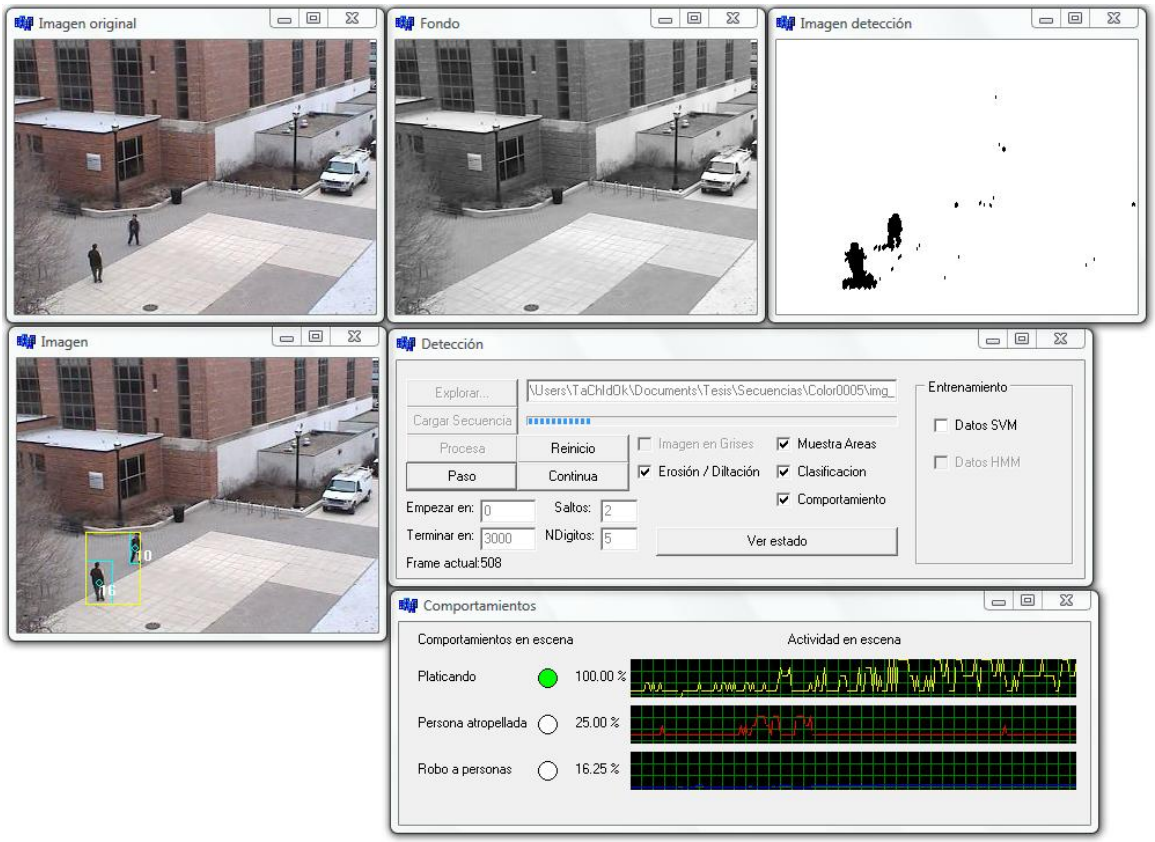

Figura 4.10: Reconocimiento del comportamiento *personas platican* 1.

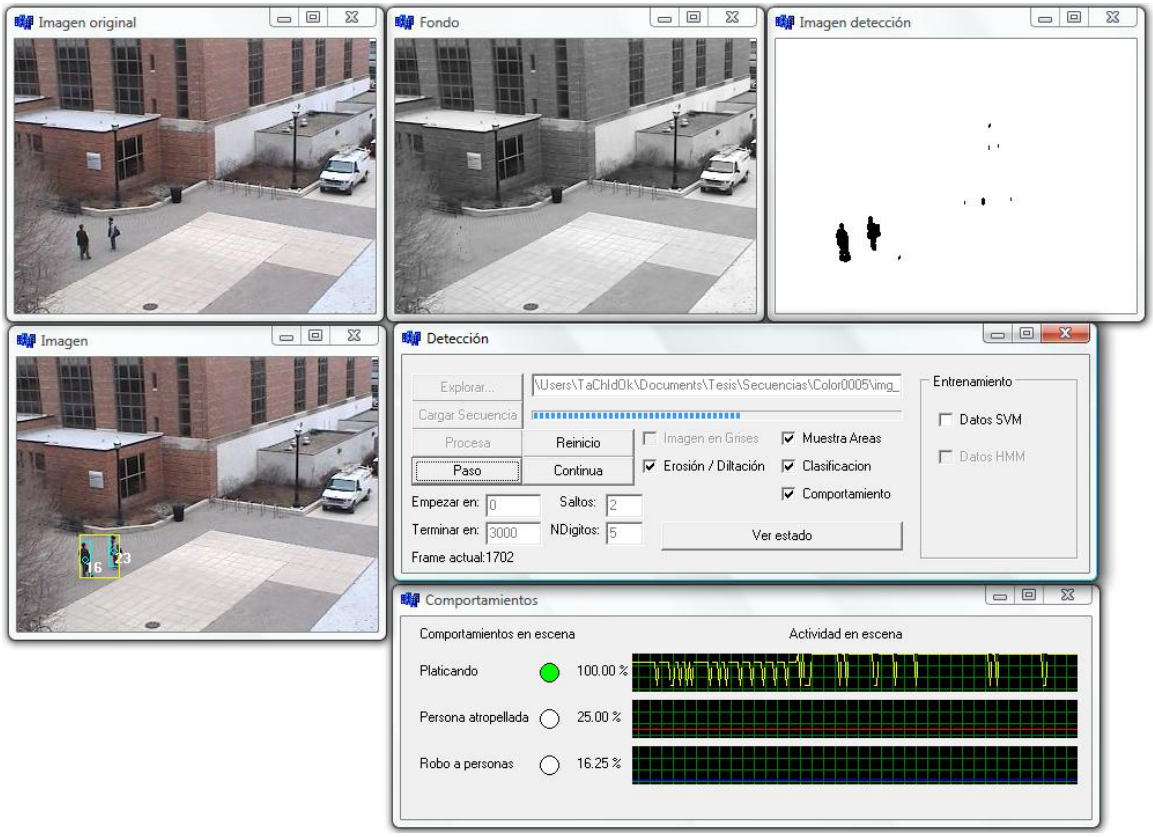

Figura 4.11: Reconocimiento del comportamiento *personas platican* 2.

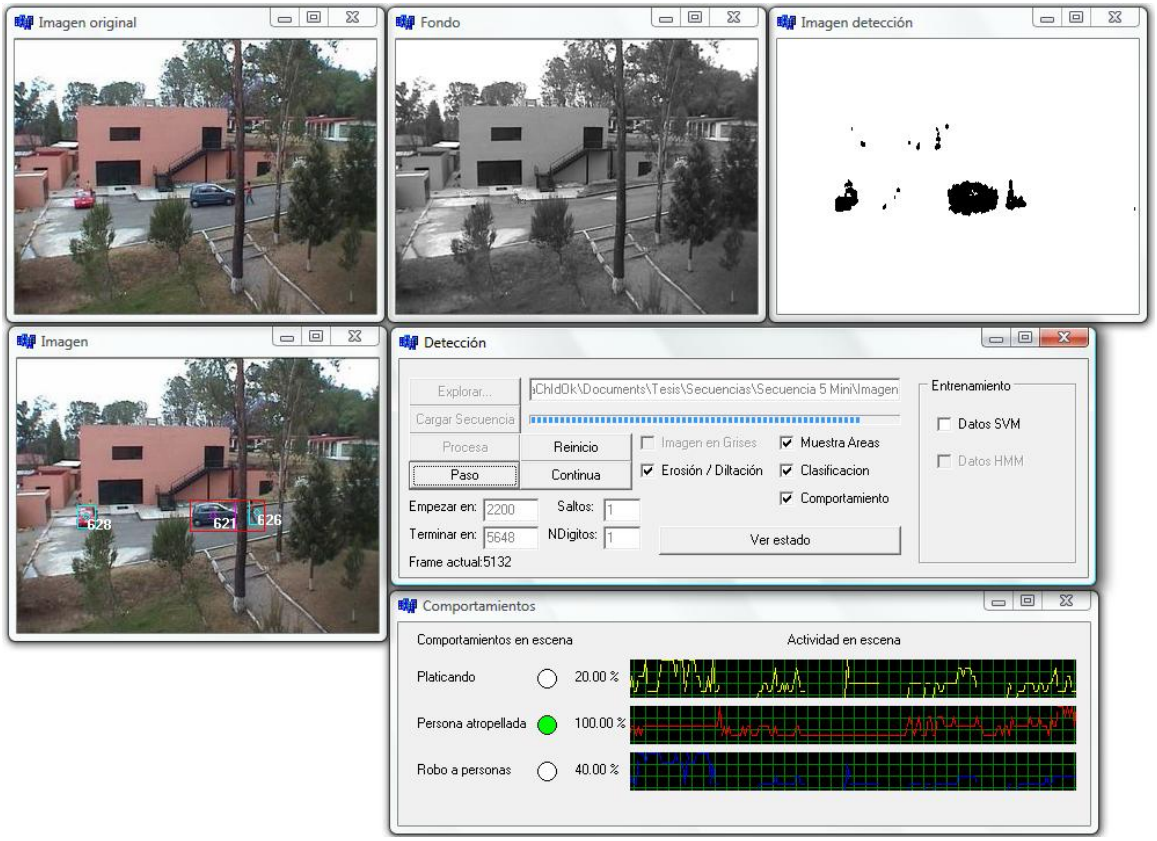

Figura 4.12: Reconocimiento del comportamiento *persona atropellada* 1.

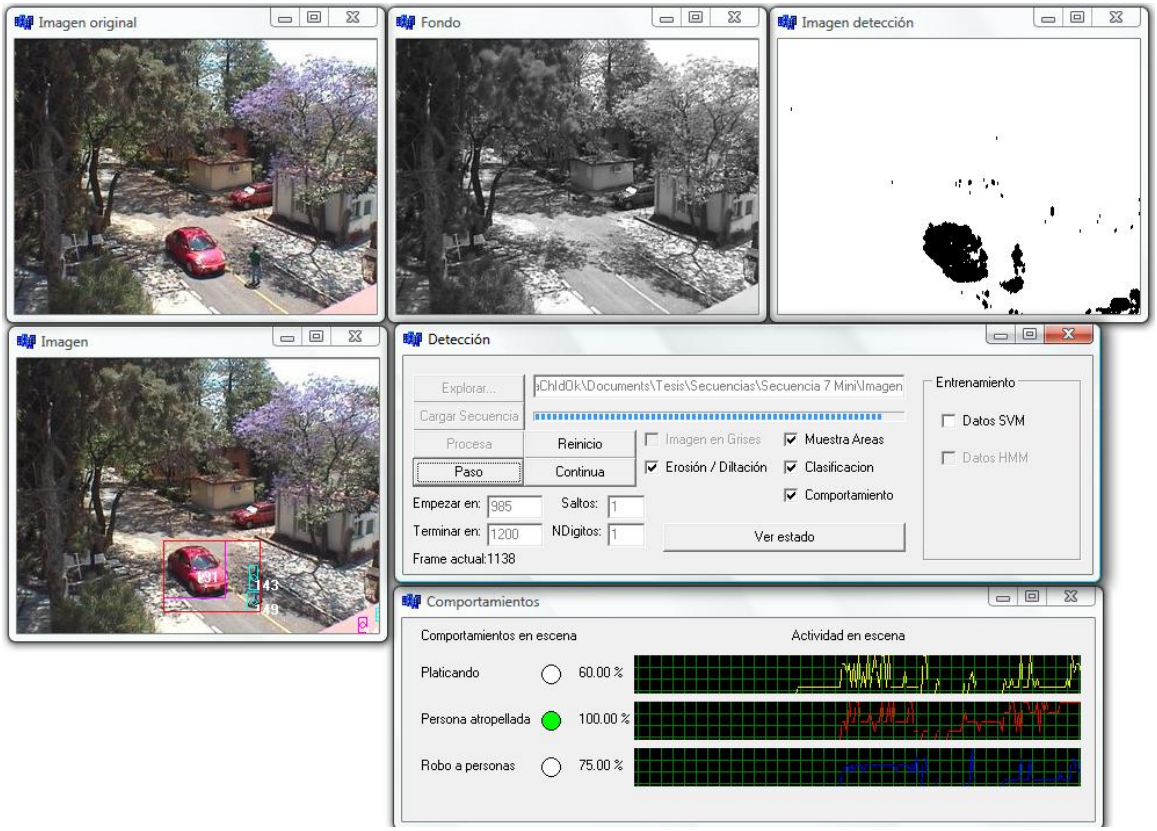

Figura 4.13: Reconocimiento del comportamiento *persona atropellada* 2.

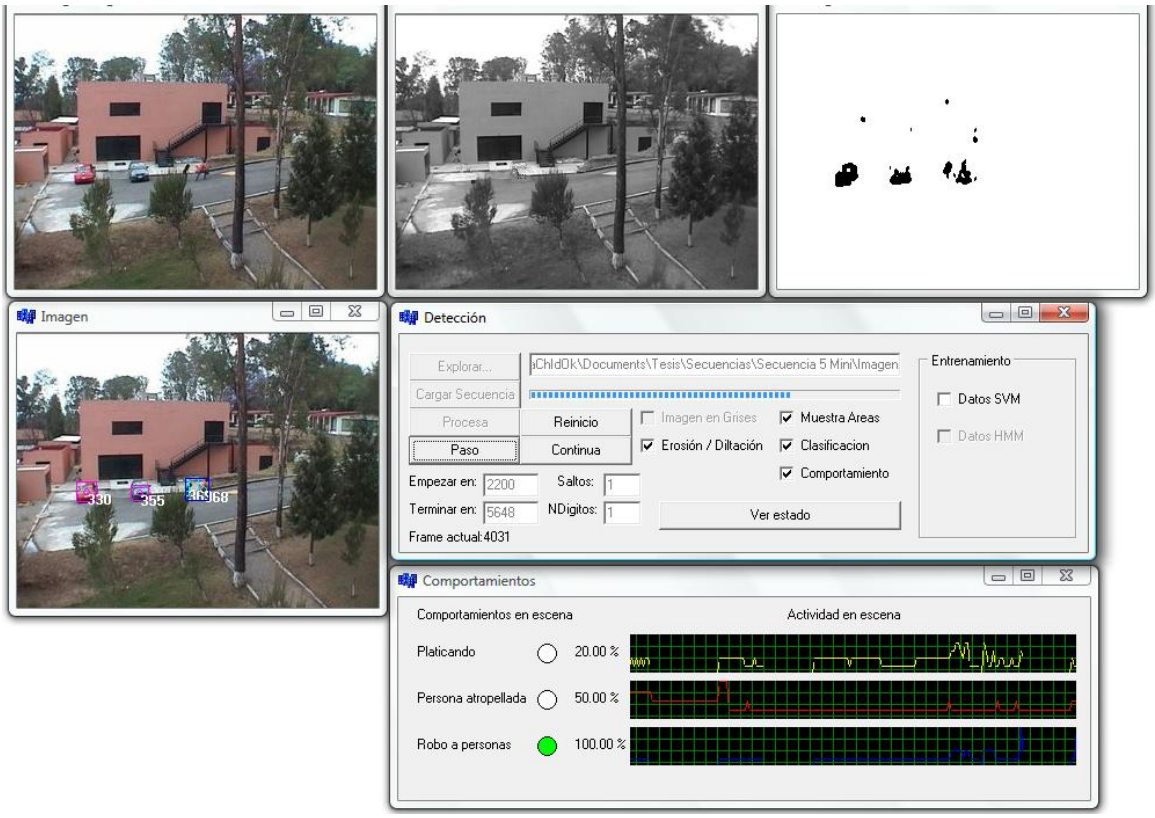

Figura 4.14: Reconocimiento del comportamiento *robo a persona* 1.

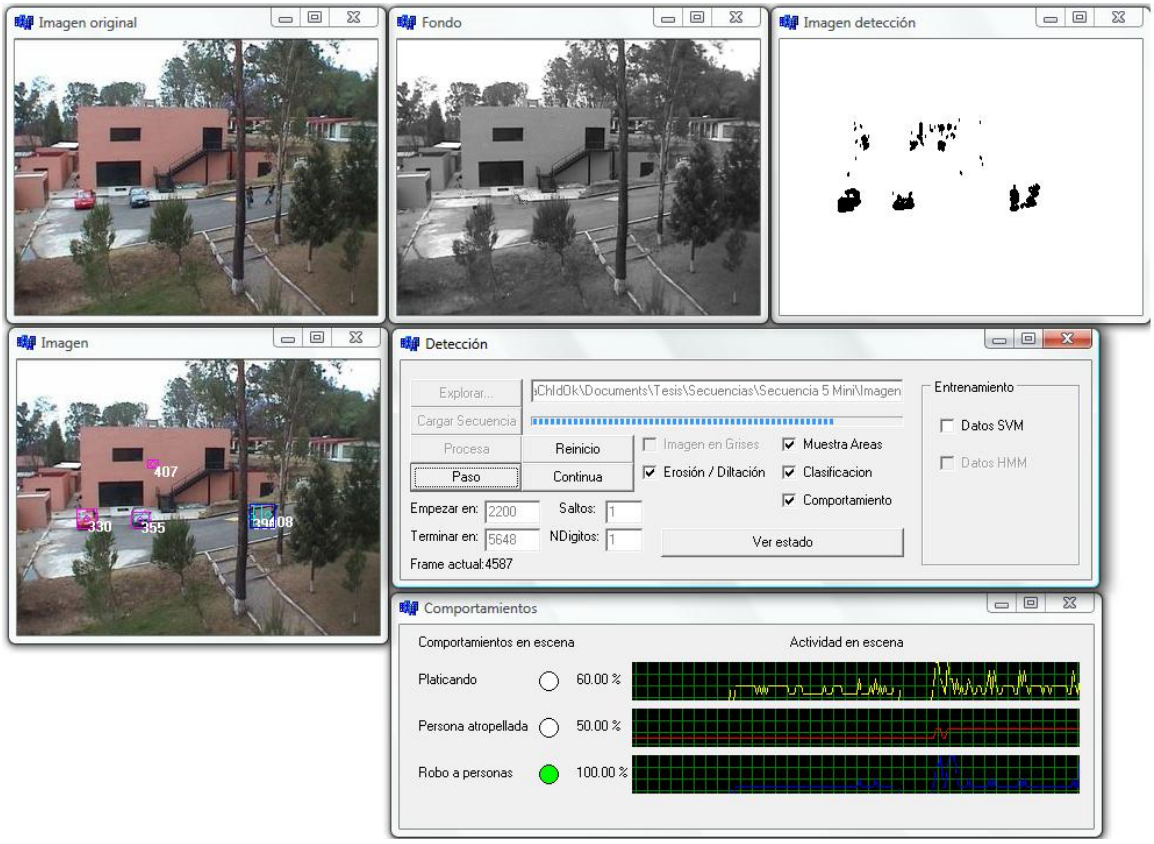

Figura 4.15: Reconocimiento del comportamiento *robo a persona* 2.

Conclusiones

En este capítulo se exponen las conclusiones principales de este trabajo junto con el trabajo propuesto para obtener mejores resultados.

En este trabajo se presentó un método o algoritmo capaz de identificar o reconocer comportamientos utilizando información de los objetos que se presentan en escena, para lograr este objetivo fue necesario desarrollar los módulos que realizan la segmentación y extracción de información de las imágenes.

Para la parte de detección de movimiento se combinaron dos métodos básicos: *diferencia de imágenes* y *fondo adaptable* obteniendo resultados satisfactorios, para mejorar la calidad de estos se utilizaron operaciones morfológicas, en este caso apertura, logrando una mejora importante en la detección, hasta este punto se puede observar que con operaciones básica es posible construir un buen detector de movimiento para cámara fija, una vez obtenido el detector se agregó una característica especial, *detección de objetos que se detienen en escena*, esta característica requiere un análisis en el tiempo de los píxeles por lo que se necesita guardar información adicional para cada uno de ellos.

En el seguimiento de objetos se presentó un algoritmo que detecta y sigue automáticamente a los objetos de interés, es decir, al detectar a los objetos los comienza a seguir sin necesidad de indicar manualmente, por medio de alguna selección, el objeto a seguir. Para detectar a los objetos utiliza información del módulo de detección de movimiento y a partir de las regiones obtenidas inicia su procesamiento.

Este algoritmo funciona para objetos con características similares a los mencionados en el capítulo de solución, esto es, que la trayectoria del objeto no presente variaciones *rápidas* sino que sean *lentas*, un ejemplo de trayectoria con variación *rápida* es la de una mosca o algunas aves, el algoritmo es capaz de mantener en seguimiento a objetos que no presenten *oclusión* total, aunque de ser así al reaparecer lo detectará y volverá a seguirlo, aunque el sistema no será capaz de identificar que se trata del objeto perdido.

Para la clasificación de los objetos se trabajo con *SVM*, gracias a la herramienta utilizada fue posible el uso de esta técnica. Los objetos que pueden identificarse son: autos y personas, las características utilizadas para distinguir entre clases son: ancho, alto, área, la razón ancho respecto al alto, elongación. Es posible agregar nuevas características para lograr una mejor clasificación, para ello hay que observar e identificar las características que resaltan la diferencia entre los objetos, para agregar nuevas características se deben agregar los valores de las características ya utilizadas del o los nuevos elementos junto con las nuevas y se vuelve a entrena, lo único que hay que modificar en el sistema desarrollado es el color con el que se identificará la nueva clase.

El utilizar medidas basadas sólo en el aspecto de los objetos muestra que con cálculos básicos se logra una buena clasificación, claro que sólo se tiene dos clases y son claramente distintas a la percepción humana.

Al escalar los datos utilizados para entrenamiento y pruebas se obtuvieron mejores resultados.

La combinación de métodos para clasificación presenta mejores resultados que un sólo método, en este caso el segundo método es el algoritmo de votación, con esto se logra *suavidad* en los resultados ya que no se presentan cambios repentinos en la clasificación de objetos.

Para el reconocimiento de comportamientos se utilizaron Modelos Ocultos de Markov, los comportamientos identificados son: personas platicando, persona atropellada y robo a personas. Cada comportamiento es modelado como una secuencia de observaciones, estas son obtenidas mediante el análisis de los objetos de la escena.

El utilizar Modelos Ocultos de Markov facilita el manejo de la incertidumbre en el procesamiento de video ya que no sólo se logra reconocer los comportamientos sino que se obtiene una probabilidad de certeza en su *clasificación*, mientras más *observaciones* se presenten mayor será la confianza del resultado. Si se compara con una máquina de estados se puede observar que el Modelo de Oculto de Markov es capaz de identificar el comportamiento aún con datos faltantes mientras que la máquina de estados no ya que se rompería el *camino* hacia el estado final el cuál funcionaría como estado de reconocimiento o identificación del comportamiento.

Los Modelos Ocultos de Markov son una buena herramienta para trabajos donde se presente incertidumbre en los datos. Los Modelos Ocultos de Markov son sensibles a los datos de entrada ya que el cambio o alteración de las observaciones produciría variaciones en la salida, es decir, en las probabilidades de reconocimiento de cada comportamiento.

Trabajo Futuro

Se pueden utilizar varias cámaras para el análisis de la escena desde varios ángulos y obtener nuevas medidas como *profundidad* de los objetos para evitar oclusiones entre ellos y perderlos, la información obtenida tendría que ser enviada a una unidad central de procesamiento donde se tomarían decisiones utilizando la información de todas las cámaras, con esto se evitarían puntos muertos.

Una alternativa para el tipo de cámaras son las cámaras en movimiento, al utilizarlas será necesario cambiar sólo el módulo para detección de movimiento ya que las técnicas utilizadas en este trabajo no son eficientes para esta variante.

En el seguimiento de objetos se pueden utilizar otras medidas de similitud como texturas o formas de los objetos en seguimiento para mayor credibilidad, para los objetos que son perdidos por obstrucción de otros objetos, que pasen detrás de otros objetos, se puede implementar algún tipo de predictor pero el reto en este caso sería minimizar el tiempo de respuesta del sistema ya que no sólo se calcularía para un objeto, sino para todos los que se encuentran en escena.

En la parte de clasificación se pueden probar nuevas características además de agregar nuevos objetos que se consideren de interés para la etapa final, el reconocimiento de comportamientos. Los nuevos objetos deberán ser considerados dentro de los nuevos comportamientos a reconocer por lo que se deben proponer nuevas observaciones.

Referencias

[1] Alan J. Lipton Robert T. Collins and Takeo Kanade. A system for video surveillance and monitoring. 2000.

[2] Juan Angel Resendiz Trejo. Las maquinas de vectores de soporte para identificación en línea. 2006.

[3] Christopher J.C. Burges. A tutorial on support vector machines for pattern recognition. 1998.

[4] Marcos Zúñiga François Brémond, Monique Thonnat. Video understanding framework for automatic behavior recognition. 2004.

[5] Svetha Venkatesh Hung Bui Nam T. Nguyen, Dinh Q. Phung. Learning and detecting activities from movement trajectories using the hierarchical hidden markov model.

[6] Geoff West Hung H. Bui Nam T. Nguyen, Svetha Venkatesh. Hierarchical monitoring of people's behaviors in complex environments using multiple cameras. 2002.

[7] Ian Reid Neil Robertson. A general method for human activity recognition in video. 2006.

[8] Tucker Balch Adam Feldman. Representing honey bee behavior for recognition using human trainable models. 2004.

[9] Sam Roweis. Hidden markov models.

[10] Lawrence R. Rabiner. A tutorial a hidden markov models and selected applications in speech recognitions. 1989.

[11] Takeo Kanade Hironobu Fujiyoshi David Duggins Yanghai Tsin David Tolliver Nobuyoshi Enomoto Osamu Hasegawa Peter Burt1 Robert T. Collins, Alan J. Lipton and Lambert Wixson1. *A System for Video Surveillance and Monitoring.* Carnegie Mellon University, Pittsburgh PA and The Sarnoff Corporation, Princeton, NJ, 2000.

[12] Raju S. Patil Alan J. Lipton, Hironobu Fujiyoshi. Moving target classification and tracking from real-time video. 2005.

[13] Jeff Johns and Sridhar Mahadevan. A variational learning algorithm for the abstract hidden markov model. 2005.

[14] Ian Reid Neil Robertson and Michael Brady. Behaviour recognition and explanation for video surveillance.

[15] Nuria Oliver Alex P. Pentland. Graphical models for driver behavior recognition in a smartcar. 2005.

[16] Ajoy K. Ray Tinku Acharya. *Image Processing Principles and Applications.* A Wiley-Interscience Publication, 2005.

[17] Troy. L. McDaniel. *Java HMM Pak User Manual*, 2006. Center for Cognitive Ubiquitous Computing (CUbiC), Arizona State University.

[18] Chih-Chung Chang and Chih-Jen Lin. *LIBSVM: a library for support vector machines*, 2001. Software available at `http://www.csie.ntu.edu.tw/~cjlin/libsvm`.

[19] Chih-Chung Chang Chih-Wei Hsu and Chih-Jen Lin. A practical guide to support vector classification. 1998.