UNIVERSIDAD AUTÓNOMA METROPOLITANA

DIVISIÓN DE CIENCIAS BÁSICAS E INGENIERÍA

CONTEO DE OBJETOS EN FLUJOS DE VIDEO H.264

Tesis que presenta
Gustavo Flores Chapa
Para obtener el grado de
Maestro en Ciencias y Tecnologías de la Información

Asesores:

Dr. John Goddard Close
Dr. Luis Martín Rojas Cárdenas

Jurado Calificador:

Presidente:	DRA. MARIKO NAKANO MIYATAKE	IPN-ESIME
Secretario:	DR. MARIO GERARDO MEDINA VALDEZ	UAM-I
Vocal:	DR. LUIS MARTÍN ROJAS CÁRDENAS	UAM-I

17 de noviembre de 2011

Casa abierta al tiempo

UNIVERSIDAD AUTÓNOMA METROPOLITANA

Fecha : 23/11/2011
Página : 1/1

CONSTANCIA DE PRESENTACION DE EXAMEN DE GRADO

La Universidad Autónoma Metropolitana extiende la presente CONSTANCIA DE PRESENTACION DE EXAMEN DE GRADO de MAESTRO EN CIENCIAS (CIENCIAS Y TECNOLOGIAS DE LA INFORMACION) del alumno GUSTAVO FLORES CHAPA, matrícula 209382346, quien cumplió con los 132 créditos correspondientes a las unidades de enseñanza aprendizaje del plan de estudio. Con fecha veinticinco de noviembre del 2011 presentó la DEFENSA de su EXAMEN DE GRADO cuya denominación es:

CONTEO DE OBJETOS EN FLUJOS DE VIDEO H.264

Cabe mencionar que la aprobación tiene un valor de 60 créditos y el programa consta de 192 créditos.

El jurado del examen ha tenido a bien otorgarle la calificación de:

Aprobar

JURADO

Presidenta

DRA. MARIKO NAKANO

Secretario

DR. MARIO GERARDO MEDINA VALDEZ

Vocal

DR. LUIS MARTIN ROJAS CARDENAS

UNIDAD IZTAPALAPA
Coordinación de Sistemas Escolares
Av. San Rafael Atlixco 186, Col. Vicentina, México, DF, CP 09340 Apdo. Postal 555-320-9000, Tels. 5804-4880 y 5804-4883 Fax: 5804-4876

Resumen

El conteo de objetos sobre flujos de video representa una funcionalidad altamente viable y novedosa para los sistemas de telemonitoreo actuales y ofrece una importante cantidad de aplicaciones en diversos ámbitos. Esta funcionalidad se ve apoyada con las características de los estándares actuales de codificación que permiten el procesamiento de secuencias de video de alta resolución con bajos costos de procesamiento y almacenamiento. En este trabajo presentamos un estudio de técnicas para el conteo de objetos móviles en video. Nuestra propuesta emplea flujos de video codificado, los cuales contienen información que describe como se mueven estos objetos en la escena. El uso de la información contenida en los flujos de video codificado reduce el costo de procesamiento, por lo que es posible generar sistemas que operen en tiempo real.

Por otra parte, el conteo de objetos sobre flujos de video requiere la identificación, además de un seguimiento, desde su aparición hasta el momento en que estos salen de la escena de video, lo cual evita que los objetos sean contabilizados en más de una ocasión. Cuando un objeto es identificado, es necesario reconocer su tipo, por lo que se requieren mecanismos que permitan una clasificación de los objetos en cuestión.

Específicamente, el presente trabajo propone el conteo de objetos empleando exclusivamente vectores de movimiento, información que se encuentra contenida en los flujos de video. La solución propuesta consiste en la identificación de los objetos para cada cuadro que contenga vectores de movimiento dentro de la secuencia de video. Para esto, manejamos la escena de como un espacio de dos dimensiones donde se ubican tales vectores, convirtiendo la tarea de identificación en un problema clásico de agrupamiento. Para esto, analizamos las técnicas de agrupamiento k-means, fuzzy c-means y mean shift, además de que proponemos un algoritmo

para la identificación de los objetos en movimiento basado en el agrupamiento de los bloques que corresponden a los vectores de movimiento. Adicionalmente mostramos los resultados obtenidos en la identificación de objetos en movimiento con estas técnicas.

Para reducir la incertidumbre generada por diversos fenómenos que afectan los resultados en las técnicas mencionadas, empleamos criterios para el filtrado de vectores de movimiento. Este tipo de filtrado se basa en las características observadas en los vectores que corresponden a los objetos en movimiento, tales como la magnitud y la dirección. Para el seguimiento de los objetos encontrados proponemos un método de predicción que contempla la información relacionada de los objetos reconocidos, tales como la ubicación, la magnitud de sus vectores y su orientación, mediante lo cual es posible estimar la posición que los objetos tendrán en cuadros siguientes. Esto permite aumentar la fiabilidad del conteo al considerar a los objetos reconocidos en una sola ocasión, además de tratar muchos de los fenómenos y errores presentes en el proceso de codificación.

De lo anterior surge una propuesta para el conteo de personas sobre flujos de video codificado. Es importante mencionar que aunque no se alcanza un sistema capaz de clasificar objetos de acuerdo a su naturaleza, mostramos que bajo condiciones limitadas es posible tener un conteo de personas sin llegar a una clasificación. Esta propuesta resulta una opción innovadora de bajo costo computacional, la cual cobra relevancia al emplear las características actuales de las principales normas de codificación. Esto significa que es posible emplearla en los sistemas de telemonitoreo actuales sin representar un cambio en la infraestructura existente.

Otro de los aspectos relevantes presentados en este trabajo es la presentación de una base experimental para el conteo de objetos sobre flujos de video H.264. Debido a la falta de una base de videos que cumpliera con las características necesarias para la experimentación de nuestra propuesta, fue necesario recopilar un conjunto de videos para el estudio de las técnicas para la tarea de identificación y seguimiento. Esto requirió, además de la generación de secuencias de video bajo la norma H.264, un proceso de etiquetado manual para la identificación de los objetos en cada uno de los cuadros de las secuencias, con lo cual se alcanza una experimentación automatizada y objetiva. Al finalizar mostramos resultados comparativos

entre las propuestas empleadas en la identificación, incluyendo métricas cuantitativas de su eficacia, lo cual resulta escaso en la literatura dado que la mayoría de las propuestas emplean solo una verificación visual sobre secuencias de video cuyas características no corresponden a la alta definición.

Agradecimientos

Agradezco al Consejo Nacional de Ciencia y Tecnología, CONACYT, por el apoyo que me fue otorgado durante mis estudios de posgrado. A la Universidad Autónoma Metropolitana-Unidad Iztapalapa y a todos los Académicos que la integran ya que son una parte muy importante de mi formación académica.

Al departamento de Ingeniería Eléctrica por todas las facilidades que me proporcionaron en el transcurso de mis estudios y los recursos que ofrecieron para el desarrollo de mi trabajo de tesis.

De manera muy especial agradezco a mis asesores, el Dr. Luis Martín Rojas Cárdenas y el Dr. John Goddard Close por todos los conocimientos compartidos conmigo, por su paciencia, dedicación y el gran apoyo brindado para la realización de este trabajo. Por los consejos e interés mostrados no solo en el aspecto académico. Porque la realización de este trabajo no hubiera sido posible sin el gran aporte de cada uno.

A mis hermanos José Luis, Rocio y Arely Elizabeth por ser la razón de todo en mi vida, por su apoyo, tolerancia y por el hecho de crecer a mi lado. A mis padres, por todo lo aprendido de ellos, a mi madre por ser la inspiración de todo en mi vida.

Agradezco a la Dra. Mariko Nakano y al Dr. Mario Medina por hacerme el honor de formar parte del jurado revisor. De manera especial, agradezco al Dr. Alfonso Prieto por la ayuda y comprensión otorgada durante mi formación académica.

Contenido

Lista de Figuras

Lista de Tablas

Capítulo 1
Introducción

En la última década, el desarrollo de sistemas para la automatización de tareas representa una de las áreas de mayor investigación dentro de las tecnologías de la información. En este documento presentamos el trabajo de investigación para el conteo de objetos en tiempo real sobre flujos de video codificado, el cual, se extiende para el conteo de personas limitando el tipo de objetos. Definimos el conteo de personas como el número total de individuos presentes en una secuencia de video. Esta tarea es posible llevarla acabo de diversas maneras, entre las que se encuentran:

- Dispositivos mecánicos: El caso más representativo son los torniquetes de acceso que se emplean para el conteo de personas. Estos mecanismos presentan un algo grado de fiabilidad, un alto costo y un bajo grado de automatización para el tratamiento de la información. En cuestiones de infraestructura, estas alternativas reducen considerablemente los espacios dedicados para la entrada/salida de las personas.

- Dispositivos químicos: Este tipo de alternativas emplean principalmente sensores luminosos para el conteo de objetos. Estos dispositivos tienen un alto grado de fiabilidad siempre que el número de objetos sea reducido, además de que tiene un alto costo.

- Flujos de video: El conteo de personas sobres flujos de video define técnicas para la extracción y procesamiento de la información contenida en las secuencias de video. Estas propuestas emplean la infraestructura presente en los sistemas de telemonitoreo actual y permiten un alto grado de automatización para el procesamiento de la información.

De las propuestas mencionadas, las propuestas que operan con flujos de video, toman como base las técnicas desarrolladas en la visión por computadora y representan una opción viable e innovadora, sobre todo considerando las aplicaciones que tiene el conteo de personas. En estas ultimas podemos mencionar:

- En los sistemas de transporte colectivo es posible estimar la horas de mayor afluencia de usuarios para establecer la cantidad de recursos dedicados a fin de conseguir un mayor grado de optimización costo-beneficio. Es posible además, estimar el número de usuarios en las trayectorias que conforman el sistema de transporte.

- El conteo de personas puede emplearse en actividades de carácter económico para establecimientos dedicados al comercio, en donde es posible conocer los lugares de mayor tránsito lo cual brinda información estadística para la toma de decisiones. Adicionalmente, es posible automatizar el control de los accesos para los usuarios, conociendo las necesidades de cada establecimiento.

- Aplicaciones industriales emplean el sistema visual humano como un método de control de calidad en los sistemas de producción. En este caso, aunque no se trata de un problema propio de conteo de personas, es posible adaptar los sistemas desarrollados para esta tarea. Esto permitiría tener líneas de producción donde se realizaría un conteo de los objetos producidos aumentado el grado de automatización.

- Es posible aplicar el conteo de personas para la administración de recursos en los denominados edificios inteligentes, en donde es posible conocer la cantidad de recursos, tales como la iluminación y calefacción, específicos para cada espacio del edificio, de acuerdo al número de personas presentes.

De lo anterior, resulta claro que los sistemas para el conteo de objetos tienen un amplio número de aplicaciones. Este tipo de sistemas se basan en dos áreas de investigación principales: codificación de video y visión por computadora.
La codificación de video define una serie de técnicas para la compresión de video con una pérdida de calidad reducida tomando como base las capacidades del sistema visual humano

[1]. En la actualidad, las técnicas empleadas para el proceso de codificación de video son robustas (tolerantes a fallas) y eficientes (alcanzan grandes niveles de compresión), sin embargo, presentan una alta complejidad.

Por otro lado, la visión por computadora, también conocida como visión artificial, es una disciplina que persigue la deducción automática de la estructura y propiedades de un escenario cambiante a partir de información capturada desde un dispositivo externo [2]. Esta disciplina busca emular las principales propiedades del sistema visual humano, el cual es capaz de procesar en decimas de segundo un gran cantidad de información distinguiendo formas, colores, texturas, condiciones de iluminación y ángulos de visión.

Aunque la situación actual de la visión por computadora dista mucho de las capacidades del complejo sistema visual humano, en las últimas décadas se ha visto un crecimiento considerable en el estudio de este campo [5]. Esto es impulsado principalmente por los avances tecnológicos en el procesamiento de medios multimedia, el cual permite, en la actualidad el manejo de grandes cantidades de información empleando los nuevos estándares de codificación.

El conteo de objetos sobre flujos de video emplea la información proporcionada por un medio externo (cámara de video) sobre la cual se busca extraer la información necesaria para el conteo correcto de los objetos. El conteo de objetos aunque, conceptualmente simple, requiere considerar una gran cantidad de aspectos tales como:

- Angulo de visión: Este representa el punto de visualización para el conteo de las personas. La decisión de este parámetro permita manejar fenómenos como las oclusiones (cubrimiento total o parcial de los objetos).
- Iluminación del medio: Fenómenos como la aparición de sombras que corresponden a las persona en movimiento se deben a la iluminación del medio.
- Reconocimiento del número correcto de personas: Es necesario tener un conocimiento del número de individuos presentes en la escena de video para un conteo veraz.
- Trayectoria de las personas: Se debe identificar la ruta que describen las personas desde

el momento en que estos aparecen hasta que salen de la escena de video, considerando para esto la irregularidad que pueden presentarse en cada trayectoria, tales como el cambio de sentido, el cambio de un estado móvil a un estado en reposo además de los fenómenos presentes al incrementar el número de objetos presentes.

Aunque el sistema visual humano es capaz de procesar mucha de esta información de manera correcta, las condiciones para que esto suceda deben ser restringidas y vigiladas, pues incluso para el ser humano, resulta una tarea compleja el conteo de personas cuando el flujo de estas es abundante. Debido a esto, existen aspectos que deben ser considerados para un sistema dedicado al conteo de personas sobre flujos de video. En este trabajo se plantea el desarrollo de un sistema para el conteo de objetos sobre flujos de video codificado empleando las características de los sistemas de codificación.

1.1. Metodología de investigación

El desarrollo de este trabajo se divide en:

1. Investigación acerca de las principales propuestas y enfoques empleados en el conteo de objetos para determinar el estado del arte en esta área.

2. Una análisis teórico y práctico de las bases para la codificación de video y las normas actuales de codificación. Un análisis teórico y práctico de las técnicas para la identificación de los objetos que permitan sustentar nuestra propuesta.

3. Extracción de los vectores de movimiento durante el proceso de decodificación para la simulación mediante el análisis de un decodificador. Implementación de las técnicas para la identificación y seguimiento de los objetos.

4. Análisis estadístico de las rutas formadas por los vectores de movimiento. Este análisis busca encontrar una correlación entre el tipo de objetos que aparecen en la escena y las trayectorias formadas por los vectores de movimiento que representan estos objetos.

5. Elaboración de un modelo que permita el seguimiento de los objetos en movimiento en los cuadros que forman las secuencias de video. Así mismo, comparar los resultados para el seguimiento de los objetos mediante nuestro sistema de predicción para lograr un conteo limitando las condiciones en cuanto al flujo de objetos en las secuencias de video.

6. Proponer la arquitectura de un sistema para el conteo de objetos en movimiento.

1.2. Contribución

Desde la aparición de los sistemas de video digital (principios de los 80s), el número de propuestas para la extracción de la información contenida en las secuencias de video avanzó de manera muy lenta, esto debido tal vez a la inexistente o apenas creciente visión por computadora y a la alta complejidad existente en las técnicas de codificación. En la actualidad, esta situación ha cambiado considerablemente. Existen esfuerzos encaminados a la extracción y procesamiento de la información contenida en las secuencias de video. Este tipo de propuestas pueden operar a nivel pixel, en cuyo caso se requiere todo un conjunto de métodos para el manejo y procesamiento de esta información.

Contrario a lo anterior, el presente trabajo emplea la información contenida en las secuencias de video codificada, en particular, la redundancia espacial y temporal que constituye la base para las actuales normas de codificación. Específicamente, este trabajo resulta novedoso por el empleo de la norma de codificación H.264, la cual es capaz de manejar una alta resolución con un bajo costo de almacenamiento. Nuestra meta es emplear distintas técnicas que operen sobre secuencias de alta resolución (1920x1080 pixeles) para el conteo de objetos. Igualmente, se presenta una propuesta para la identificación de los objetos en movimiento basada en bloques que tiene como característica principal una baja complejidad, lo que permite competir con las principales propuestas, debido a la reducida cantidad de recursos necesarios para su funcionamiento.

Explotando las técnicas para la identificación de objetos, presentamos un método para el seguimiento de objetos basado en predicciones de movimiento que emplea parámetros es-

tadísticos extraídos de la experimentación realizada.
Resulta importante mencionar que presentamos una base de datos de video preparada para la experimentación. Contrario a la mayoría de las propuestas, nuestra base de datos presenta secuencias de video con una alta resolución, sobre las cuales se realizo un procedimiento de etiquetado para la comparación objetiva de las trayectorias de los objetos en movimiento.

1.3. Estado del conocimiento

En la actualidad, los avances alcanzados en el terreno de la multimedia y la infraestructura de telemonitoreo presente permite obtener flujos codificados de video en tiempo real. El seguimiento de objetos sobre flujos de video ha sido una tarea abordada desde diversas perspectivas. Las principales propuestas están ubicadas en dos ámbitos particulares:

- Flujos de video no codificado: Empleando la información a nivel pixel.
- Flujos de video codificado: En esta, se emplea la información contenida y que es resultado del proceso de codificación.

Los flujos de video representan grupos de imágenes (GOP Group Of Pictures), las cuales contienen información como vectores de movimiento (VM) y coeficientes frecuenciales de luminancia y crominancia. Esta información puede ser empleada en la visión por computadora para el conteo de objetos.
Los vectores de movimiento describen como se desplazan regiones de imágenes en cuadros consecutivos, esta información se encuentra contenida solo en los cuadros del tipo P y B, excluyendo los cuadros I que se decodifican de diferente manera. Estos vectores de movimiento son empleados para la identificación de los objetos [4][7][9][10][12][14][15][16], representado esto como un problema de agrupamiento.
Un número considerable de propuestas emplean k-means para el agrupamiento de vectores [8], esta técnica representa una alternativa viable para la identificación de objetos debido a la simplicidad del método. Ademas, k-means presenta una gran flexibilidad para el agrupamiento de acuerdo a la representación de la información, por lo que es posible usar esta técnica

empleando la información del color (RGB), extraída en cada cuadro de la secuencia de video, para la identificación de las regiones homogéneas que corresponden a objetos en movimiento [9]. Esta técnica requiere conocer apriori, el número de objetos en movimiento presentes en cada cuadro donde se contengan vectores de movimiento o información de colores, lo cual representa el principal inconveniente, pues esta información no es posible conocerla al inicio del proceso de decodificación. Muchas de estas propuestas presentan técnicas para la elección del número de agrupaciones, sin embargo, esto representa un incremento en el costo computacional, además de que requieren un número adicional de información para representar el contorno de los objetos cuando se emplea la información del color.

Dada la necesidad de conocer el número de agrupaciones, propuestas como [15], [21] emplean fuzzy c-means. Esta técnica permite la identificación automática del número de agrupaciones empleando un grado de incertidumbre que indica la probabilidad de que un vector pertenezca a cada una de las agrupaciones analizadas. El proceso para encontrar el número de agrupaciones que optimicen (minimizar) el grado de incertidumbre significa un aumento en la cantidad de cálculos en comparación a k-means, lo cual cobra importancia, al considerar que este proceso debe realizarse sobre cada cuadro de la secuencia que contenga vectores de movimiento.

Bajo la misma línea de trabajo, encontramos a mean shift [22], técnica no paramétrica dado que no requiere conocer el número de agrupaciones, empleando el radio de las agrupaciones definido como ancho de banda. Propuestas como [23][24] se basan en esta técnica de agrupamiento para la segmentación de las imágenes en combinación con estimaciones para la realización del rastreo. En [24] se emplea mean shift como una técnica para el agrupamiento sobre el dominio YCrCb, además de mostrar cómo puede emplearse mean shift para una representación multi-nivel mediante la variación del ancho de banda. Sin embargo, mean shift presenta un alto costo computacional además de requerir el ancho de banda idóneo para la identificación de múltiples objetos. Adicionalmente, propuestas como [23][24] trabajan sobre un dominio que requiere una cantidad considerable de información para una segmentación basada en el color o combinan técnicas adicionales como k-means.

Otro enfoque empleado para el rastreo de objetos se basa en el uso adaptativo de bloques [9], estos bloques corresponden a las regiones de los objetos en movimiento. Este tipo de propuestas emplean la alta correlación espacial entre cuadros consecutivos para el rastreo de objetos usando la dirección que llevan los vectores de movimiento, lo que permite tratar fenómenos como las oclusiones y disoluciones. Inconvenientemente, estas propuestas requieren de un proceso de inicialización en el que interviene el usuario o algún método de segmentación como el presentado en [27], lo que impide tener un conteo de objetos autónomo viable para los sistemas actuales de telemonitoreo.

En una gran cantidad de trabajos se emplea, además, información que permite obtener un mayor grado fiabilidad en el reconocimiento y rastreo de objetos. En [9] se propone el uso de coeficientes DCT para aumentar el grado de fiabilidad en el rastreo de objetos, sin embargo, este tipo de enfoques exigen un mayor número de recursos computacionales. Otro aspecto a considerar que afecta altamente el desempeño y necesidad de recursos es la información necesaria para la detección y rastreo de los objetos, en [15] se aborda la tarea de agrupamiento de vectores mediante k-means, de manera similar, en [25] se emplean redes neuronales para el rastreo fiable de los objetos, mediante el uso de la información obtenida en los cuadros I. En [7] se propone el uso de la información contenida en los cuadros I para la proyección de objetos en movimiento.

La posición de las cámaras, es otro aspecto que juega un papel primordial para la eficiencia de los sistemas de reconocimiento y rastreo de objetos. De acuerdo al entorno es posible decidir el número y la posición adecuada de las cámaras, en muchas ocasiones el rango de visión de una sola cámara se ve limitado por la posición de esta e incluso afecta la fiabilidad del sistema de rastreo. [6] proponen el uso de múltiples cámaras para el rastreo de objetos, lo que aumenta la fiabilidad sus propuestas, especialmente en presencia de oclusiones. Este tipo de propuestas se basan en el análisis individual de la información procedente de cada fuente de video para su posterior fusión y análisis. Sin embargo, debido a la necesidad de una infraestructura específica y al costo computacional, estas propuestas se ven limitadas. Bartolini, Capellini y Giani en [11] muestran que la posición de la cámara, permite obtener

mejores resultados, cuando los ejes de la cámara son paralelos al flujo del video.
Existen pocas referencias que presenten una verificación formal para sus propuestas en el conteo de personas; en general se suele emplear una verificación visual para los resultados obtenidos en el conteo de personas sobre flujos de video; tanto en la parte de la identificación de los objetos para cada cuadro como la trayectoria que estos describen lo que impide poder realizar una comparación entre los distintos métodos encontrados. De igual forma, muchos de los trabajos de investigación emplean secuencias de video que no corresponden a las capacidades de procesamiento y resolución de las técnicas actuales de codificación.
Esto se debe en gran medida a la falta de una base experimental, sobre todo porque las secuencias de video deben guardar características específicas para el conteo de los objetos.

1.4. Organización del documento

Este trabajo está organizado de la siguiente manera:

- En el capítulo 2 se presentan las técnicas de base comúnmente empleadas en la codificación de video. Así mismo se destacan aquellos elementos empleados para el desarrollo de nuestro trabajo.

- En el capítulo 3 se presenta la propuesta y el análisis teórico que la sustenta. Detallamos la forma en que se interpreta el problema de conteo, además de las técnicas empleadas.

- En el capítulo 4 justificamos la creación de una base experimental para el desarrollo de nuestras pruebas. De igual forma se destacan los elementos empleados para su creación.

- En el capítulo 5 se presentan los resultados experimentales de las pruebas realizadas. Por último, en el capítulo 6 mostramos las conclusiones y el trabajo a futuro con base en las observaciones generadas.

1.5. Estado del conocimiento

En la actualidad, los avances alcanzados en el terreno de la multimedia y la infraestructura de telemonitoreo presente permite obtener flujos codificados de video en tiempo real. El seguimiento de objetos sobre flujos de video ha sido una tarea abordada desde diversas perspectivas. Las principales propuestas están ubicadas en dos ámbitos particulares:

- Flujos de video no codificado: Empleando la información a nivel pixel.
- Flujos de video codificado: En esta, se emplea la información contenida y que es resultado del proceso de codificación.

Los flujos de video representan grupos de imágenes (GOP Group Of Pictures), las cuales contienen información como vectores de movimiento (VM) y coeficientes frecuenciales de luminancia y crominancia. Esta información puede ser empleada en la visión por computadora para el conteo de objetos.

Los vectores de movimiento describen como se desplazan regiones de imágenes en cuadros consecutivos, esta información se encuentra contenida solo en los cuadros del tipo P y B, excluyendo los cuadros I que se decodifican de diferente manera. Estos vectores de movimiento son empleados para la identificación de los objetos [4][7][9][10][12][14][15][16], representado esto como un problema de agrupamiento.

Un número considerable de propuestas emplean k-means para el agrupamiento de vectores [8], esta técnica representa una alternativa viable para la identificación de objetos debido a la simplicidad del método. Ademas, k-means presenta una gran flexibilidad para el agrupamiento de acuerdo a la representación de la información, por lo que es posible usar esta técnica empleando la información del color (RGB), extraída en cada cuadro de la secuencia de video, para la identificación de las regiones homogéneas que corresponden a objetos en movimiento [9]. Esta técnica requiere conocer apriori, el número de objetos en movimiento presentes en cada cuadro donde se contengan vectores de movimiento o información de colores, lo cual representa el principal inconveniente, pues esta información no es posible conocerla al inicio del proceso de decodificación. Muchas de estas propuestas presentan técnicas para la elección

del número de agrupaciones, sin embargo, esto representa un incremento en el costo computacional, además de que requieren un número adicional de información para representar el contorno de los objetos cuando se emplea la información del color.

Dada la necesidad de conocer el número de agrupaciones, propuestas como [15], [21] emplean fuzzy c-means. Esta técnica permite la identificación automática del número de agrupaciones empleando un grado de incertidumbre que indica la probabilidad de que un vector pertenezca a cada una de las agrupaciones analizadas. El proceso para encontrar el número de agrupaciones que optimicen (minimizar) el grado de incertidumbre significa un aumento en la cantidad de cálculos en comparación a k-means, lo cual cobra importancia, al considerar que este proceso debe realizarse sobre cada cuadro de la secuencia que contenga vectores de movimiento.

Bajo la misma línea de trabajo, encontramos a mean shift [22], técnica no paramétrica dado que no requiere conocer el número de agrupaciones, empleando el radio de las agrupaciones definido como ancho de banda. Propuestas como [23][24] se basan en esta técnica de agrupamiento para la segmentación de las imágenes en combinación con estimaciones para la realización del rastreo. En [24] se emplea mean shift como una técnica para el agrupamiento sobre el dominio YCrCb, además de mostrar cómo puede emplearse mean shift para una representación multi-nivel mediante la variación del ancho de banda. Sin embargo, mean shift presenta un alto costo computacional además de requerir el ancho de banda idóneo para la identificación de múltiples objetos. Adicionalmente, propuestas como [23][24] trabajan sobre un dominio que requiere una cantidad considerable de información para una segmentación basada en el color o combinan técnicas adicionales como k-means.

Otro enfoque empleado para el rastreo de objetos se basa en el uso adaptativo de bloques [9], estos bloques corresponden a las regiones de los objetos en movimiento. Este tipo de propuestas emplean la alta correlación espacial entre cuadros consecutivos para el rastreo de objetos usando la dirección que llevan los vectores de movimiento, lo que permite tratar fenómenos como las oclusiones y disoluciones. Inconvenientemente, estas propuestas requieren de un proceso de inicialización en el que interviene el usuario o algún método de segmentación

como el presentado en [27], lo que impide tener un conteo de objetos autónomo viable para los sistemas actuales de telemonitoreo.

En una gran cantidad de trabajos se emplea, además, información que permite obtener un mayor grado fiabilidad en el reconocimiento y rastreo de objetos. En [9] se propone el uso de coeficientes DCT para aumentar el grado de fiabilidad en el rastreo de objetos, sin embargo, este tipo de enfoques exigen un mayor número de recursos computacionales. Otro aspecto a considerar que afecta altamente el desempeño y necesidad de recursos es la información necesaria para la detección y rastreo de los objetos, en [15] se aborda la tarea de agrupamiento de vectores mediante k-means, de manera similar, en [25] se emplean redes neuronales para el rastreo fiable de los objetos, mediante el uso de la información obtenida en los cuadros I. En [7] se propone el uso de la información contenida en los cuadros I para la proyección de objetos en movimiento.

La posición de las cámaras, es otro aspecto que juega un papel primordial para la eficiencia de los sistemas de reconocimiento y rastreo de objetos. De acuerdo al entorno es posible decidir el número y la posición adecuada de las cámaras, en muchas ocasiones el rango de visión de una sola cámara se ve limitado por la posición de esta e incluso afecta la fiabilidad del sistema de rastreo. [6] proponen el uso de múltiples cámaras para el rastreo de objetos, lo que aumenta la fiabilidad sus propuestas, especialmente en presencia de oclusiones. Este tipo de propuestas se basan en el análisis individual de la información procedente de cada fuente de video para su posterior fusión y análisis. Sin embargo, debido a la necesidad de una infraestructura específica y al costo computacional, estas propuestas se ven limitadas. Bartolini, Capellini y Giani en [11] muestran que la posición de la cámara, permite obtener mejores resultados, cuando los ejes de la cámara son paralelos al flujo del video.

Existen pocas referencias que presenten una verificación formal para sus propuestas en el conteo de personas; en general se suele emplear una verificación visual para los resultados obtenidos en el conteo de personas sobre flujos de video; tanto en la parte de la identificación de los objetos para cada cuadro como la trayectoria que estos describen lo que impide poder realizar una comparación entre los distintos métodos encontrados. De igual forma, muchos

de los trabajos de investigación emplean secuencias de video que no corresponden a las capacidades de procesamiento y resolución de las técnicas actuales de codificación.
Esto se debe en gran medida a la falta de una base experimental, sobre todo porque las secuencias de video deben guardar características específicas para el conteo de los objetos.

Capítulo 2
Codificación de video

Los sistemas computacionales permiten la representación de información del mundo externo, esta información debe ser almacenada de acuerdo a sus características y es caso de estudio de la teoría computacional. Pensemos por ejemplo en los sistemas manejadores de bases de datos; estos permiten el manejo de información que modela un conjunto de datos, permitiendo introducir procesar y modificar información. La cantidad de información que estos sistemas deben almacenar puede crecer de manera considerable originando la necesidad de políticas óptimas para su almacenamiento. De manera similar, los sistemas de video requieren características especiales para el almacenamiento y procesamiento de la información.

Los sistemas de video permiten la representación de un espacio tridimensional proyectándolo como un espacio bidimensional mostrando la escena de video [3]. Esta información se captura como una secuencia de imágenes. Tomemos como ejemplo una secuencia de video con una dimensión de 352 por 288 pixeles [1]. Asumiendo que los componentes primarios del espacio de colores RGB (rojo, verde y azul) son representados, comúnmente, como un pixel de 8 bits y que el movimiento de las imágenes se alcanza mostrando 30 imágenes por segundo requerimos entonces 352x288x8x3x30 = 72990720 bits por segundo para una secuencia de video con características típicas de la televisión analógica (CIF). El tiempo es un parámetro que interviene en el proceso cíclico de la presentación mediante el cual se obtiene un coeficiente conocido como bit rate[1] el cual representa la cantidad de información necesaria por segundo para la presentación del video. La tabla 2.1 muestra el bit rate necesario para los principales formatos de video bajo una representación de un espacio de colores YUV que será tratado

[1]En la codificación de imágenes se define un pixel como la unidad mínima de color homogéneo

Formato	Luminancia	Crominancia	Cuadros por segundo	Bitrate
ITU-R601	858x525	249x255	30	216 Mbps
CIF	352x288	176x144	30	36.5 Mbps
QCIF	176	88x72	15	4.6 Mbps

Tabla 2.1: Formatos de video y bit rate asociado

en la siguiente sección.

Es clara la necesidad de técnicas para el procesamiento de video que permitan reducir la cantidad información para su almacenamiento, procesamiento y transmisión; estas deben mantener un nivel bajo de pérdida de información, sobre todo cuando en la actualidad se ha incrementando la complejidad de los servicios de video tales como la alta definición HD, los servicios de transmisión de video por Internet, entre otros.

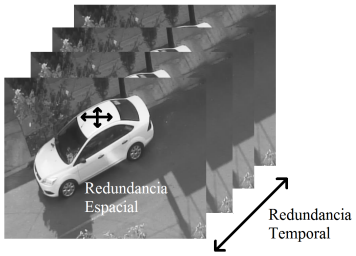


Figura 2.1: Redundancia espacial y temporal

Los codificadores deben considerar aspectos como la cantidad de recursos y el tiempo de procesamiento para la conversión de información, además de que el proceso inverso de transformación debe llevar a la información original. Sin embargo, resulta prácticamente imposible desarrollar un sistema para la compresión de la información que cumpla con estas características, sobre todo porque en la actualidad todos los procesos de transformación varían en uno o más aspectos pero siempre presentan pérdida de información cuando el nivel de compresión es alto. En las secciones siguientes se muestra un panorama general de la codificación/decodificación de video enfatizando aquellos elementos que se emplean en nuestro trabajo.

2.1. Codificador/decodificador de video

El proceso de compresión consiste en compactar la información en un número menor de bits, de acuerdo al tamaño original. La codificación de video digital es el proceso de compactar o condensar una secuencia de video dentro de un número menor de bits [1]. El video no codificado o *raw video* generalmente requiere una alta tasa de transferencia que vuelve impráctico el almacenamiento o transmisión del video digital.

Los sistemas de compresión de video implementan dos componentes básicos para comprimir (encoder o codificador) y descomprimir (decoder o decodificador) el video digital. El encoder provee al sistema los mecanismos necesarios para convertir la información original a un medio óptimo, reduciendo el número de bits empleados, para su almacenamiento y transmisión, mientras que, el decoder tiene la tarea de transformar la información comprimida a una representación del video original, tal representación no constituye el video original, pues durante el proceso de compresión se presenta perdida de información. El sistema conjunto enCOder/DECoder es conocido como CODEC.

Generalmente, la mayoría de los métodos de codificación de video explotan la redundancia temporal y espacial durante el proceso de compresión (ver Figura A.1). En el dominio espacial usualmente se emplea la alta correlación presente entre cada uno de los pixeles de la imagen, los cuales describen el brillo (luminancia) y color de la muestra, esto es, los valores de los vecinos de un pixel muestreado mantienen una alta similitud.

Un codificador de video (video encoder) consta de tres módulos principales: un módulo temporal, un módulo espacial y un codificador de entropía. La entrada del módulo temporal es una secuencia de video sin comprimir. El módulo temporal intenta reducir la redundancia temporal explotando las similitudes entre los cuadros vecinos del video, usualmente, mediante la construcción de una predicción del cuadro de video actual. La salida del módulo temporal es un cuadro residual (creado por la sustracción entre la predicción construida y el cuadro actual), obteniendo un conjunto de vectores de movimiento que describen como se compensa el movimiento entre los cuadros.

El cuadro residual constituye la entrada para el módulo espacial, el cual emplea las similitu-

des entre los elementos de la imagen (pixeles) muestreados pertenecientes al cuadro residual con el fin de reducir la redundancia espacial, generalmente, esta reducción se logra mediante la aplicación de una transformada a las muestras residuales para cuantificar los resultados. Los coeficientes cuantificados se emplean para remover los valores insignificantes, obteniendo un número reducido de coeficientes que proveen una representación más compacta del cuadro residual. La salida del módulo espacial es un conjunto de coeficientes cuantificados. La información resultante del módulo temporal (generalmente vectores de movimiento) y el módulo espacial (coeficientes) son compactados en el módulo de codificación de entropía. El módulo de codificación de entropía elimina la redundancia estadística en la información y produce un flujo de bits compacto o un archivo que puede ser empleado para transmitir o almacenar. El proceso de decodificación tiene una menor complejidad en comparación al proceso de codificación. Este reconstruye los cuadros del video contenidos en el flujo comprimido. Los coeficientes y vectores de movimiento son obtenidos por un decodificador de entropía, los cuales posteriormente se decodifican para construir una versión del cuadro residual. El decodificador emplea los vectores de movimiento y uno o más cuadros, decodificados previamente, para crear una predicción del cuadro actual y este es reconstruido agregando el cuadro residual para su predicción.

En la actualidad existen un número extenso de métodos desarrollados para el proceso de compresión de video, sin embargo en este trabajo solo estudiamos aquellos que representan la base del proceso de codificación de video. En lo que resta de la sección se presentan los detalles de los módulos que intervienen en el proceso de codificación y decodificación de video, además de una descripción del manejo de la escena, el proceso de muestreo y el espacio de colores.

2.2. Espacio de colores

Uno de los aspectos para la compresión es la representación óptima de la información de colores, esto permite reducir la cantidad de información necesaria para codificación de video manteniendo la resolución de las imágenes. Los colores perceptibles por el ojo humano pueden

ser obtenidos mediante la combinación de tres componentes primarios: rojo, verde y azul. En 1931, la CIE[2] presenta un estudio denominado **CIE 1931 color space XYZ**, en el que se muestra un nuevo modelo para un espacio de colores donde se consideran las cualidades del sistema visual humano.

Aunque la representación del espacio de colores RGB es frecuentemente empleado y aceptado para tareas de procesamiento de imágenes, esta no permite emular las capacidades del sistema visual humano debido a que cada uno de los tres componentes tienen la misma relevancia omitiendo las características del sistema visual humano; este es más sensible a cambios en la luminancia que los cambios que se producen en el tono de los colores, denominados crominancia. Este es el punto central de espacios del espacio de colores $YCbCr$, el cual guarda una mantiene una relación con el espacio de colores RGB como se muestra a continuación:

$$Y = 0{,}299R + 0{,}587G + 0{,}114B$$

$$Cb = -0,147R + 0{,}289G + 0{,}436B$$

$$Cr = 0{,}615R - 0{,}515G - 0{,}100B$$

Este sistema de ecuaciones representa un modelo de colores generalmente usado en las técnicas de codificación debido a que modela los cambios de brillo de manera más significativa que los que se presentan en los colores. Esta representación de colores permiten diferentes procesos de muestreo que están relacionados estrechamente con la calidad de las imágenes, en contraste al modelo RGB que requiere siempre el mismo tipo de muestreo $4:4:4$.

- Muestreo 4:4:4 Este tipo de muestreo representa la forma original de representación del espacio de colores YCrCb compuesta por un cuatro muestras de cada componente Y, Cr y Cb. Este formato no presenta perdida de información.

- Muestreo 4:2:2[3] Este tipo de muestreo consiste de cuatro componentes Y y una componente Cr y Cb. Esto representa la mitad de la información necesaria respecto al muestreo 4:4:4.

[2]International Commission on Illumination

[3]Descrito algunas veces como "12 bits por pixel"

- Muestreo 4:2:0 Este tipo de muestreo es empleado en secuencias de video entrelazado.

2.3. Escena de video

Una escena de video representa una proyección de un escenario del mundo real compuesto de múltiples objetos con características particulares, tales como el color, textura, granularidad, forma, contorno e iluminación en la cual los elementos pueden mantener un movimiento que debe ser capturado dentro del video. Estos elementos están sujetos al ángulo de proyección de la cámara, lo que influye en características tales como el brillo. Durante el proceso de captura de la escena se generan imágenes que son denominadas cuadros.

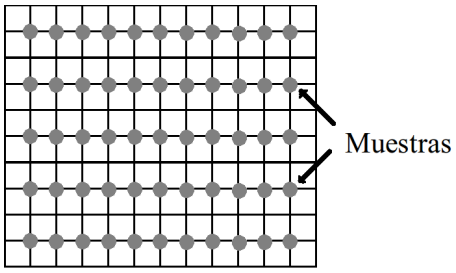


Figura 2.2: Muestreo temporal de video

Para la captura de video se requiere un muestreo espacial y temporal. El proceso de muestreo espacial define una maya sobre la que se divide la escena formando segmentos, denominados bloques, que se ubican en un espacio bidimensional (ver figura 2.2), donde las características de cada segmento de la imagen definen la resolución total del video. Por otro lado, el proceso de muestreo temporal busca lograr una apariencia de movimiento con la presentación contínua de imágenes secuenciales que corresponden a imágenes capturadas donde se observan los objetos en movimiento.

El proceso de muestreo puede derivar en la presentación de imágenes completas, este proceso se denomina (muestreo progresivo) o la presentación de segmentos de la imagen denominados campos (muestreo entrelazado). En el muestreo entrelazado cada imagen se divide en dos campos, un campo alto y un campo bajo separados, generalmente, por la mitad del periodo de presentación del cuadro. Los campos altos corresponden a las líneas impares de la imagen

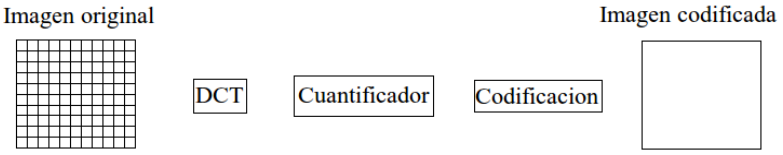


Figura 2.3: Compresión de imágenes

mientras que los campos bajos se extraen de las líneas pares. El proceso de presentación intercala los campos dentro de un muestreo espacial. Mientras que en el muestreo progresivo encontramos el concepto de campos y cuadros, solo este último está presente en el muestreo progresivo.

2.4. Redundancia espacial

Una de las características principales que se explotan en la codificación de video es la alta relación que presentan los elementos (pixeles) dentro de una imagen. Esta característica permite reducir la cantidad de información de las imágenes de referencia o residuales, estudiadas en secciones posteriores, que forman parte del proceso de compresión.

Para el proceso de compresión de imágenes [28], esta es divida en bloques de tamaño nxn pixeles; la elección del tamaño de los bloques influye en el factor de codificación aunque también se incrementa la complejidad computacional; generalmente se emplea un tamaño de $8X8$ o $16x16$ pixeles. La figura 2.3 muestra un panorama general para el proceso de compresión de imágenes que será descrito brevemente en lo que resta de la sección.

2.4.1. Transformación DCT

Una de las técnicas ampliamente usadas[4] para la compresión de imágenes es la Transformada de Coseno Discreta (DCT) [29]; esta permite representar una sucesión finita de elementos como la suma de funciones sinusoidales. Debido a la alta relación que existe entre los elementos de una imagen, es posible emplear la DCT siendo indistinguible los cambios, además de que

[4]Debido a los resultados próximos, términos de la compresión de energía, a la transformada Karhunen-Loeve la cual alcanza un alto nivel de compresión y costo computacional

es posible variar los coeficientes resultantes mediante la elección de la matriz de cuantización, lo que permite controlar la calidad visual resultante en el proceso de compresión. La DCT opera sobre un bloque o matríz de coeficientes, lo cuales son extraídos de las imágenes y se define de la siguiente manera:

$$CR = AXA^T \tag{2.1}$$

$$X = A^T CRA \tag{2.2}$$

Donde CR es la matriz de coeficientes resultantes, X es una matriz cuadra de muestras y A es la matriz de transformación. La ecuación 2.2 representa la inversa de la DTC (IDCT). Los coeficientes en la matriz A corresponden a los términos de funciones cosinusoidales, expresadas como:

$$C_i = \begin{cases} \frac{1}{n} & Si \quad i = 0 \\ \sqrt{\frac{2}{n}} & Si \quad i > 0 \end{cases} \tag{2.3}$$

En la esquina superior izquierda de la matriz resultante se ubica un coeficiente denominado DC[5], mientras que los coeficientes restantes son denominados AC; estos últimos codificados de distinta manera respecto al coeficiente DC. De esta manera la DCT logra concentrar la mayor parte de la información (en la esquina superior izquierda) en un número reducido de coeficientes siendo posible representar el bloque total con una menor cantidad de información.

2.4.2. Cuantificación

Una vez que se conoce la forma en que se concentra la información, es necesario determinar cuáles coeficientes de la matriz resultante deben codificarse sin que exista una pérdida significativa en la calidad visual. Para determinar el nivel real de los coeficientes transformados se emplean matrices de cuantificación. Estas matrices están determinadas por el tipo de componente de color y la calidad buscada.

[5]El término DC (Corriente Directa) empleado en la literatura tienen su origen en la electrónica, donde se refiere a un valor de carga eléctrica continuo. De manera análoga se emplea el termino AC (Corriente alterna)

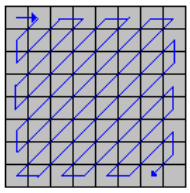


Figura 2.4: Secuencia de ordenación

Al finalizar la cuantización, la mayoría de los coeficientes resultantes toman el valor cero, esto permite una codificación entre códigos de longitud variable (como por ejemplo Huffman) y códigos de longitud de series(RLE). Al finalizar este proceso se tiene el orden de codificación como el mostrado en la figura 2.4.

2.5. Modelo de redundancia temporal

Aunque la redundancia espacial permite tener mecanismos de compresión, estos resultan ser insuficientes considerando que un video es la presentación de un conjunto de imágenes. Imaginemos una secuencia de video capturada en la que la cámara no presenta movimiento; esta secuencia muestra un automóvil desplazando por la escena, resulta claro que el codificar solo los cambios presentes entre los cuadros del video reduce la cantidad de información. Las técnicas de codificación suelen emplear la alta relación que se presenta entre cuadros consecutivos; denominada redundancia temporal.

Los cambios entre cuadro consecutivos de las secuencias de video pueden ser producidos por objetos rígidos y deformables en movimiento, por el movimiento de la cámara, por el fenómeno de las oclusiones/disoclusiones[6] o cambios de iluminación; estos movimientos representan un cambio en la ubicación de los elementos de los cuadros.

Como se menciono anteriormente, un flujo de video está formado por un conjunto de cuadros denominado GOP[7]. El modelo temporal busca eliminar la redundancia entre los cuadros presentes en la secuencia de imágenes. Para esto, algunas propuestas emplean imágenes de

[6]Las oclusiones son fenómenos en el que los objetos son eclipsados total o parcialmente por algún otro objeto. De manera opuesta, las disoclusiones constituyen un fenómeno en el que los objetos aparecen en la escena después de haber sido ocultados por otro objeto.

[7]Group of Pictures, Grupo de imágenes

referencia (ver Figura 2.5); estas sirven para calcular la diferencia con cuadros siguientes, la diferencia entre los cuadros se calcula como la resta de cada elemento del cuadro actual con los cuadros siguientes. Los cuadros de referencia son codificadas completamente mientras que para los cuadros siguientes solo decodifican las variaciones con respecto a los cuadros de referencia.

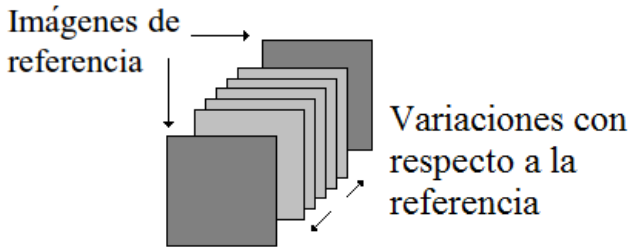


Figura 2.5: Estructura de un GOP con cuadros de referencia

Este tipo de técnica presenta resultados altos en la compresión cuando el video no se presenta un movimiento significativo de los objetos (como aplicaciones de telemonitoreo o videoconferencia), sin embargo, en otros casos suele ser necesario otro tipo de codificación.

2.5.1. Estimación de movimiento basado en bloques

El generar una predicción de movimiento para las imágenes tomando como base el cambio entre pixeles presenta una carga computacional considerable que depende de la resolución de las imágenes. Es por esto que, una de las prácticas empleadas en las técnicas de codificación consiste en la segmentación de las imágenes en bloques de tamaño mxn pixeles. Cada uno de estos bloques representa una región de la imagen sobre la cual se realizará un proceso de predicción, tomando una imagen de referencia, como se describe:

1. Se define una región de búsqueda en el cuadro sobre la imagen de referencia. Se realiza un proceso de búsqueda en la que se compara la región de la imagen actual con una o todas las regiones de tamaño mxn en el área de búsqueda. Uno de los criterios que se suelen emplear consiste en la menor diferencia entre ambas regiones en términos de la

energía residual [8].

2. La región seleccionada (un bloque de tamaño mxn) del área de búsqueda es sustraída de la imagen para ser codificada de manera conjunta con el cambio de posición de esta región.

Este proceso genera vectores de movimiento los cuales describen el cambio de posición de las regiones entre cuadros consecutivos. Un número importante de estándares[9] emplean el denominado *macrobloque* en el modelo de predicción y compensación de movimiento. Para un flujo de video 4:2:0, un macrobloque está estructurado como se muestra en la figura 2.6.

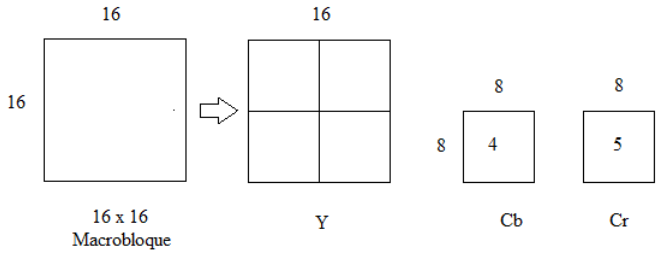


Figura 2.6: Estructura de un macrobloque con un muestreo 4:2:2

2.6. Norma de codificación H.264

En la actualidad existen una amplia gama de normas para la codificación de video, estas varían en sus características de acuerdo al tipo de aplicación (almacenamiento, videoconferencias, indexado, entre otras). Desde la década de los 90, existen diversos grupos enfocados al desarrollo de técnicas para la codificación de video. El grupo de expertos en imágenes en movimiento (MPEG) de ISO/EC son los desarrolladores de estándares para la codificación de video para diversas aplicaciones de televisión y almacenamiento; en tanto que el Grupo de Expertos en Codificación de Video (VCEG) de la ITUT-T se enfocaron al desarrollo de

[8]Relación señal a ruido de pico PSNR

[9]Estándares como MPEG-1, MPEG-2, MPEG-4, H.261, H.263 y H.264 emplean bloques de tamaño de 16x16 pixeles

aplicaciones para videoconferencia. En esta sección se presentan las principales características de la norma de codificación de alta compresión H.264 [8], las cuales son empleadas en nuestro trabajo.

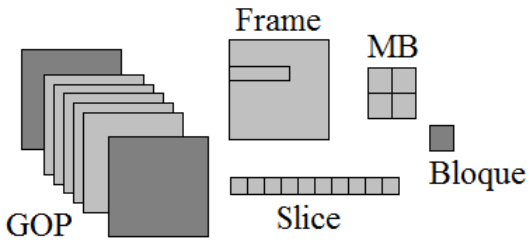


Figura 2.7: Estructura del video en las normas de codificación MPEG-2

La norma H.264 o MPEG-4 parte 10 para codificación de video surge como un esfuerzo conjunto de estos dos grupos de trabajo. Esta norma presenta características como:

- Alcanza un alto nivel en calidad de video y compresión, además de ser robusto a errores. Esto último lo logra, incorporando un ordenamiento tolerante de macrobloques FMO(Flexible Macroblock Ordering) y la transmisión redundante de rebanadas para evitar la propagación de errores.

- Representa una extensión de los formatos predecesores (MPEG-1, MPEG-2, H.261, H.263) al mantener la base para la codificación. La norma H.264 emplea un módulo de transformación para el manejo de la información residual y correlación espacial de los cuadros, incorpora además, un módulo para la predicción de movimiento para el tratamiento de la redundancia temporal. Mantiene la estructura del video presente en normas anteriores (ver Figura 2.7) e incorpora un tamaño de macrobloques de $4x8$, $8x4$ y $4x4$.

- Extiende el tipo rebanadas I (codificadas utilizando predicción Intra, en la cual se emplean solo muestras de la rebanada a codificar, sin emplear referencias a otras), P(codificadas empleando técnicas de predicción INTER en una sola dirección con respecto a cuadros previamente codificados) y B (se codifican empleando técnicas de pre-

dicción en dos direcciones tomando tramas previas y posteriores) incorporando: rebanadas SP, las cuales se codifican de manera similar a las rebanadas P y permiten manipular la resolución entre tramas de manera óptima; rebanadas SI las cuales se codifican de manera similar a las rebanadas SP permitiendo el cambio entre diferentes flujos.

- Presenta diferentes perfiles para el flujo de datos (bitstream). Estos especifican un conjunto de características y alcances del codificador tales como elementos de la sintaxis del estándar; esto determina la carga que opera sobre el codificador. En la norma H.264 existen tres perfiles bases: *baseline* el cual se emplea para aplicaciones de teleconferencia; *main* empleado para aplicaciones de video digital de alta calidad; *extended* el cual es emplea en aplicaciones multimedia para Internet.

Aunque la codificación especificada en la norma H.264 es tema de un profundo estudio, pocos son los elementos que se emplean en el presente trabajo. Siendo los principales: el orden de decodificación de los cuadros, el cual permite conocer aquellos en los que no existen vectores de movimiento de acuerdo a la norma(tales como los cuadros I); el orden de decodificación de los macrobloques; la información resultante del proceso de codificación de redundancia temporal (vectores de movimiento).

Capítulo 3
Conteo de personas sobre flujos de video codificado

Los sistemas que procesan y transmiten flujos de información de tipo media, tales como el audio, el video y las imágenes, han tenido una fuerte penetración en la última década debido, entre otros, a la creación y normalización de técnicas para la compresión, la baja en los costos de fabricación y el despliegue global de redes de datos tales como Internet. Un caso representativo son los sistemas de telemonitoreo, los cuales se encuentran instalados por miles en casi todas partes donde haya actividad humana. Hoy en día, estos sistemas han pasado, de ser simples sistemas para la adquisición de imágenes, a ser sistemas dotados de cierta inteligencia que les permite ser capaces de emular tareas normalmente realizadas por el hombre, tal como la detección de incendios, la detección de conductas criminales, el conteo de personas, etc. Entre estas tareas, el conteo de personas constituye la motivación principal de este trabajo de tesis; este problema es abordado limitando el tipo de objetos. Aunque para el ser humano resulta conceptualmente simple el conteo de objetos -siempre y cuando el flujo sea poco denso-, esto no resulta equivalente para los sistemas computacionales, los cuales deben cubrir una cantidad importante de aspectos. Debido a eso, en esta sección mostramos los aspectos cubiertos para la realización de un conteo de personas sobre flujos de video codificado.

Un primer paso para la realización del conteo de personas, consiste en verificar que los objetos en movimiento son realmente personas sobre la secuencia de video. Nuestra propuesta para el conteo de personas sobre flujos de video requiere identificar los objetos que existen dentro de la secuencia. Asumiendo que las personas que aparecen en la secuencia de video constituyen

un tipo particular de objetos, los cuales comparten un conjunto característico de propiedades, las cuales hacen a estos objetos diferentes de otros. En nuestra propuesta, las personas no pueden clasificarse como un tipo de objeto rígido (tales como automóviles) debido al hecho de que tienen partes móviles (tales como los brazos, las piernas, la cabeza, el torso) que pueden presentar un movimiento distinto entre sí (ver figura 3.1); además de influir (arrastrar o empujar) en el movimiento de otro tipo de objetos.

En principio, la tarea de conteo de objetos se puede aplicar para el conteo de personas siempre que se restrinja el tipo de objetos encontrados en la secuencia. Esto permite adaptar el sistema de conteo a otros contextos variando las características de los objetos a contabilizar, por ejemplo, pensemos en dos objetos de naturaleza distinta, tales como los automóviles y las personas, la proporción del área cubierta por estos en la escena de video y el comportamiento de sus vectores de movimiento difiere de manera significativa, por lo que es posible adaptar las características del sistema propuesto para el conteo de estos dos tipos de objetos, sin llegar a una clasificación.

Dado que las personas pertenecen al conjunto de objetos en movimiento, es necesaria la tarea de seguimiento para conocer el momento en que estos *aparecen/desaparecen* de la escena de video. Aunque es posible realizar un conteo de objetos para cada cuadro de la secuencia, es posible que el número de objetos encontrados difiera para cuadros consecutivos, incluso cuando la cantidad real de objetos no ha sufrido ningún cambio. Esto puede ser causado por errores en la codificación o por fenómenos como las *oclusiones/disoclusiones* en el que regiones de las imágenes son *cubiertas/descubiertas* y no contienen vectores de movimiento. Por esta razón, el conocer la información del desplazamiento de los objetos en la línea temporal permite reducir el número de errores, siendo esta idea la base para nuestro sistema de seguimiento de objetos.

En términos generales, las técnicas de conteo deben estar acotadas en tiempo de ejecución y recursos computacionales siempre que se pretenda tener un sistema que opere en tiempo real. Con este objetivo en mente y con el fin de reducir la complejidad de los algoritmos para el conteo se emplean exclusivamente los vectores de movimiento, sin embargo, también es

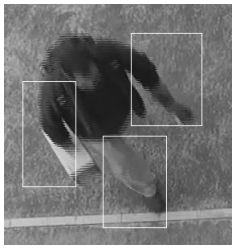


Figura 3.1: Partes moviles de las personas

necesario implementar criterios de filtrado que operen sobre los vectores de movimiento a fin de tener solo aquellos que corresponden a los objetos en movimiento. Aunque el uso de los vectores limita la identificación a exclusivamente objetos en movimiento, esto resulta idónea para el presente trabajo.

Lo que resta del capítulo, se presenta la representación de la escena y el tratamiento de los vectores de movimiento. Se prosigue mostrando el funcionamiento teórico y su adaptación de las técnicas empleadas para la identificación de objetos en movimiento. En la parte final del capítulo, mostramos el método de *predicción de posición* empleado para el seguimiento de los objetos, el cual incrementa la precisión del proceso de seguimiento de objetos.

3.1. Escena de video

En este trabajo se considera la escena de video como un espacio bidimensional sobre el que se ubican los objetos. Esto se logra tomando como unidad el pixel, considerando por ejemplo, un video cuya resolución es de $8x8$ pixeles tenemos que la escena alcanza una dimensión de $8x8$ unidades. De esta manera el tamaño total de la componente en x y y está determinado por la dimensión horizontal y vertical, respectivamente, del video como se muestra en la figura 3.2.

Bajo estas consideraciones, es posible representar la ubicación de un pixel como un par ordenado (x, y). Continuando con el ejemplo, la figura 3.2 muestra un pixel en las coordenadas *(4,5)*; la representación de la escena como un espacio euclidiano permite calcular el desplazamiento de los pixeles de acuerdo al cambio de su ubicación. La figura 3.2 muestra el

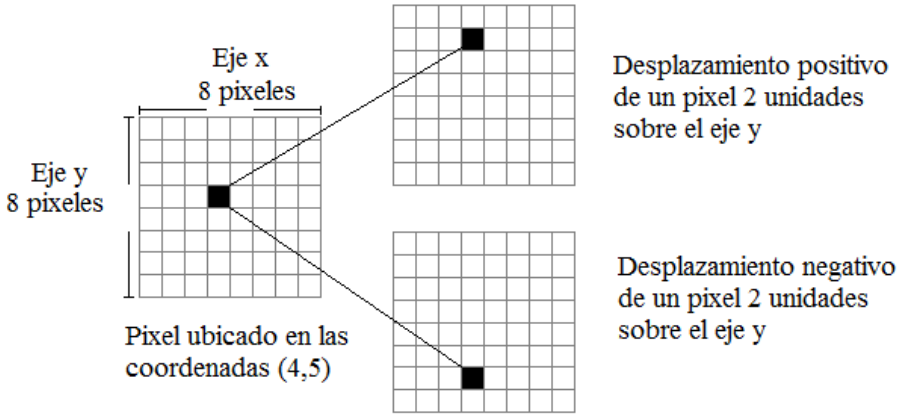


Figura 3.2: Ejemplos del posible desplazamiento de un pixel

desplazamiento del pixel cuando este varia de manera negativa y positiva en su componente y. Este tipo de representación se puede extender para agrupaciones de pixeles, conocidos como macrobloques, donde se toma la posición superior izquierda del macrobloque como la ubicación de la posición original a partir de la cual se calcula el desplazamiento que este presenta. La figura 3.3 muestra un macrobloque en el cual se aprecia la posición que es tomada para esta agrupación de pixeles.

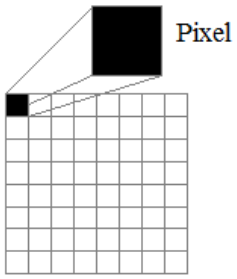


Figura 3.3: Estimación de movimiento basado en macrobloques

3.2. Vectores de movimiento

Los vectores de movimiento representan la información base para la identificación de los objetos en el presente trabajo [1]; estos vectores son generados durante el proceso de codificación

del flujo de video e indican la forma en que desplazan las regiones (bloques) de una imagen entre cuadros consecutivos (ver Figura 3.4).

Para comprender el proceso de decodificacion de vectores y su significado consideremos lo siguiente: El proceso se realiza sobre un GOP[1], el cual está formado por un número variable de cuadros. De acuerdo al tipo de cuadro del que se trate, este contendrá o no vectores de movimiento. Mientras que los cuadros I se decodifican de manera independiente, los cuadros P y B se decodifican tomando como referencia imágenes anteriores o posteriores de acuerdo al orden que guardan dentro del GOP. Por esta razón, los cuadros I no presentan vectores de movimiento siendo excluidos de nuestro trabajo, esto permite reducir la cantidad de información; sin embargo, también hace necesario un método de predicción que refleje la *continuidad de los objetos en movimiento* en los casos en los que se carezca de los vectores de movimiento.

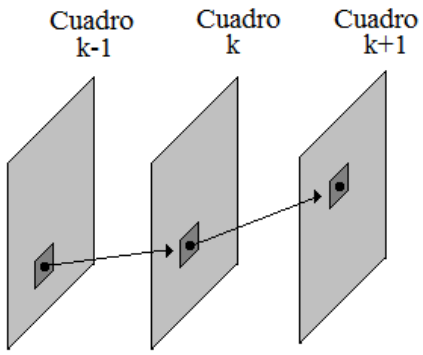


Figura 3.4: Vectores de movimiento generados sobre una secuencia continua de cuadros

Internamente, un cuadro contenido en un GOP, está formado por un número finito de pixeles, los cuales generalmente se agrupan en macrobloques. Dado que la codificación de video permite dividir una imagen en bloques, es posible calcular el número de estos bloques en base a la resolución del video y el tamaño de los bloques.

$$mb_{total} = (\frac{video_w}{mb_w}) \cdot (\frac{video_h}{mb_h}) \tag{3.1}$$

En donde mb_{total} es el número total de macrobloques, $video_w$ y $video_h$ es el tamaño horizontal

[1]Group of Pictures

y vertical respectivamente, de manera análoga para los macrobloques; tenemos que mb_w y mb_h es el tamaño horizontal y vertical respectivamente. Considerando por ejemplo, un video con una resolución de 1920x1088 pixeles y macrobloques con un tamaño de 16x16 pixeles tendremos un total de 8160 macrobloques.

En la norma H.264, existe una variedad considerable de macrobloques; sin embargo en este trabajo solo nos concentramos en aquellos que guardan información de movimiento. Cada uno de los cuales mantiene un número que indica su ubicación dentro de la imagen, aunque este índice no representa la ubicación de esta región dentro de la imagen es posible realizar un proceso de conversión para poder llevar esta información a la representación clásica de un vector de movimiento. La conversión de esta información se realiza de la siguiente manera:

$$\begin{aligned} mb_{x0} &= (i \cdot mb_w)\, MOD\, (video_w) \\ mb_{y0} &= (i \cdot mb_h)\, DIV\, (video_w) \end{aligned} \tag{3.2}$$

donde mb_x0, mb_y0 son las componentes x, y del macrobloque con índice i, mb_w y mb_h son las dimensiones del macrobloque, $video_w$ es la dimensión horizontal del video (ver ecuación 3.1), MOD es la operación módulo y DIV es la división entera. Esta transformación se realiza para cada macrobloque que tiene movimiento, los cuales guardan información acerca del desplazamiento que tienen en cuadros futuros; esta información se emplea para construir el vector de movimiento para el correspondiente macrobloque:

$$\begin{aligned} mb_{x1} &= mb_{x0} + \Delta x \\ mb_{y1} &= mb_{y0} + \Delta y \end{aligned} \tag{3.3}$$

donde (mb_x1,mb_y1) representan la ubicación destino del macrobloque y (Δx, Δy) son las componentes en (x, y) del desplazamiento.

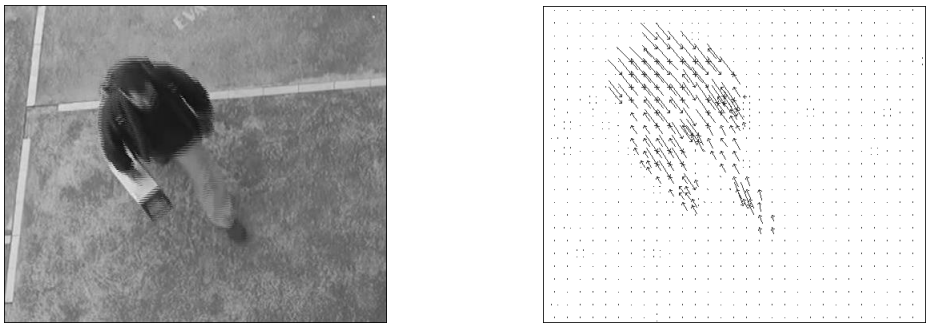
3.2.1. Extracción de vectores

A cada GOP, en la secuencia de video, se aplica un procedimiento inverso sobre el flujo de video a fin de reconocer la sintaxis de los elementos codificados; es en estos donde encontramos la información acerca del desplazamiento de los macrobloques.

Para extraer el índice y el desplazamiento de los macrobloques para cada cuadro dentro del GOP, se empleó el decodificador H.264 incluido en la librería *libavcodec*. Libavcodec es una librería para la codificación/decodificación de diversos flujos de video. Este decodificador fue estudiado y analizado; lo cual permitió modificarlo a fin de extraer los siguientes datos:

- El orden de presentación/decodificación de los cuadros.
- El tipo de cuadro, rebanadas y macrobloque.
- La información de desplazamiento de los macrobloques, la cual es empleada para la construcción de los vectores de movimiento.
- Información de colores (contenidas en los coeficientes DCT).

El proceso de extracción permite la construcción de los vectores de movimiento; además de conocer el orden de presentación de las imágenes. La figura 3.5 muestra el resultado de la extracción y construcción de los vectores de movimiento.



(a) (b)

Figura 3.5: Extracción de vectores

3.2.2. Técnicas de filtrado

Debido a la presencia de diversos fenómenos tales como el movimiento de la cámara y errores de codificación, es posible apreciar vectores que no corresponden a los objetos en movimiento que se desean tratar; la cantidad de estos vectores representa un incremento significativo en la carga computacional para las técnicas de agrupamiento, además de afectar los resultados

obtenidos por las técnicas para la identificación de los objetos. Técnicas como k-means (las cuales serán estudiadas en las siguientes secciones) operan calculando la distancia media entre cada uno de los elementos por lo que la presencia de vectores que no correspondan a los objetos en movimiento repercute en el cálculo del centro de las agrupaciones.

La figura 3.6a muestra un conjunto de vectores los comparten características como la dirección y magnitud; mientras que la figura 3.6b muestra el *centroide* de esta agrupación. La presencia de vectores con características diferentes influye en la ubicación del centro de la agrupación como se muestra en la figura 3.6d.

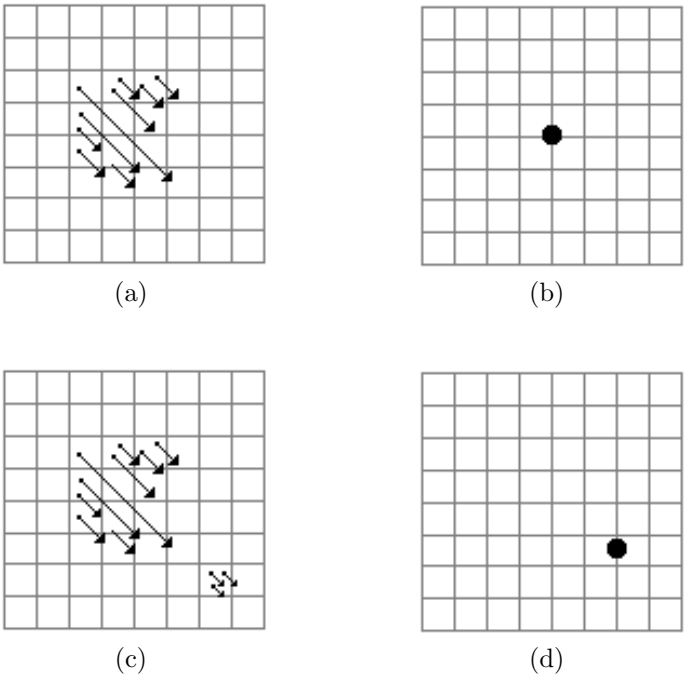


Figura 3.6: Efectos de los vectores de sobre la identificación de objetos

Esto muestra la necesidad de criterios que permitan reducir los vectores a fin de trabajar con aquellos que corresponden exclusivamente a los objetos en movimiento. En este trabajo empleamos un filtraje que opera sobre la magnitud de los vectores de movimiento de la siguiente manera: Calculamos la magnitud del vector como:

$$|v_i| \;=\; \sqrt{(mb_{x1} - mb_{x0})^2 + (mb_{y1} - mb_{y0})^2} \tag{3.4}$$

Donde $|v_i|$ es la magnitud del vector de movimiento asociado al macrobloque con índice i;

este índice representa la ubicación del macrobloque dentro de la imagen. Si la magnitud es menor al umbral definido, el vector se omite. En este trabajo definimos un umbral para el filtrado de los vectores de movimiento de 16 unidades (pixeles), el cual, según un análisis visual representa la frontera entre el ruido y la información útil, la figura 3.7 muestra los resultados del filtrado para el cuadro mostrado en la figura 3.5.

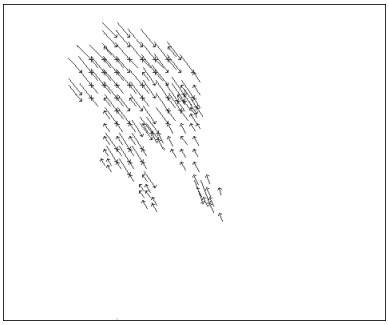


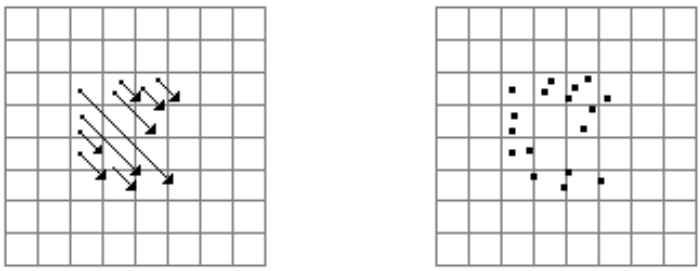
Figura 3.7: Filtrado de vectores de movimiento

3.3. Identificación de objetos en movimiento

El objetivo de esta sección es presentar las técnicas empleadas para la identificación de los objetos empleando exclusivamente los vectores de movimiento. Tomando como referencia la sección anterior, podemos asumir que los vectores de movimiento que corresponden a los objetos comparten características particulares, tales como la orientación y la magnitud.
La tarea de agrupamiento (clustering), también conocida como clasificación no supervisada es un método para generar grupos de objetos (clústeres), donde los elementos de un mismo clúster comparten un conjunto de características entre sí (o medidas de similitud). Este tipo de clasificación no supervisada suele ser empleada en áreas como la minería de datos, procesamiento de imágenes, compresión de datos y reconocimiento-clasificación de patrones. Dentro de esta área, podemos hablar de dos tipos de agrupamiento: agrupamiento duro (hard) y difuso (fuzzy). En el agrupamiento duro podemos mencionar las siguientes características:

$$\begin{aligned} X &= \{x_1, ..., x_n\} \\ x_i \in C_j,\ 1 \leq i \leq n,\ 1 \leq j \leq k \\ |C_j| &> 0 \end{aligned} \tag{3.5}$$

Donde X es el conjunto de datos a agrupar cuya cardinalidad es n. C_j representa el $j-esimo$ conjunto etiquetado dentro del cual se ubicarán los elementos (del conjunto de datos a clasificar X) que comparten características o medidas de similitud, en este sentido, el agrupamiento duro representa una técnica de clasificación excluyente, dado que un elemento no puede pertenecer a más de un clúster; k representa el número total de agrupaciones. Por otro lado, existen problemas en que los elementos no pueden ser claramente separables, es decir, existe un grado de ambigüedad en la asignación de un elemento hacia un determinado clúster. Esto es conocido como agrupamiento difuso (fuzzy) y es representado de manera análoga al agrupamiento duro, con la salvedad de que en este caso se cuenta con un coeficiente que permite conocer el grado de ambigüedad con que los elementos son agrupados.



(a) Vectores de movimiento

(b) Puntos agrupar

Figura 3.8: Problema de agrupamiento

El proceso de agrupamiento requiere un conjunto de tareas que permiten la representación de un modelo para el reconocimiento de patrones. Dentro de estas podemos mencionar las siguientes:

- Representación de la información. Esto se define en base a las características de la información que se maneja para el problema de agrupamiento. En este trabajo la in-

formación la constituyen los vectores de movimiento, particularmente se emplean los puntos correspondientes al destino y origen de los vectores (ver fig. 5.1).

- Definición de una medida de proximidad que opera sobre la información que se desea agrupar. Dentro del agrupamiento de datos, términos como coeficiente de similitud, medida de similitud, coeficiente de disimilitud o distintica son empleados para describir la similitud o disimilitud que mantienen los elementos entre sí. El criterio empleado en este trabajo es la distancia Euclidiana que existe entre los vectores de movimiento. Supongamos que se cuenta con dos objetos en la secuencia de video, etiquetados como A y B. Es posible asumir que la distancia de los vectores de movimiento del objeto A es menor entre sí comparado con la distancia a los vectores del objeto B, por lo que es posible emplear la distancia Euclidiana.

- Abstracción de la información. En algunas ocasiones se requiere una interpretación adicional de la información. En este trabajo sintetizamos la información de los objetos en movimiento empleando el centroide de los clústeres considerando la orientación y magnitud promedio para describir su desplazamiento.

Asumiendo que la información (vectores de movimiento) que se tiene corresponde exclusivamente a los vectores de movimiento, es posible tratar la tarea de identificación de objetos como un problema de agrupamiento, en el que cada vector de movimiento es asignado a un objeto en movimiento. Para la identificación de los objetos que tienen movimiento autónomo (o generado por la interacción de estos) se asumen dos propiedades relacionadas al ángulo de visión y posición de la cámara:

1. La cámara se encuentra ubicada desde una vista superior respecto al flujo de los objetos en movimiento, lo que permite reducir el impacto de los fenómenos de las oclusiones. Aunque no se logra evitar por completo este fenómeno, es posible tener, en el peor de los casos, un ocultamiento parcial de los objetos en movimiento.

2. Teóricamente, la cámara no presenta ningún movimiento, por lo que no se generan

vectores de movimiento por el desplazamiento de la escena de video. Sin embargo, en la práctica esto es algo pocas veces conseguido.

3. Siempre que el movimiento de la cámara genere vectores de movimiento, cuyas características sean diferenciables de las generas por los objetos de movimiento, es posible realizar un proceso de filtrado de tal manera que se conserve solo la información de los vectores de movimiento.

Como veremos a continuación, estas tareas se emplean para la identificación de los objetos en movimiento. Además de esto, muchas de las técnicas empleadas en este trabajo requieren la definición de parámetros que permiten adaptar el agrupamiento de vectores para una variedad de objetos.
En lo que resta de la sección se muestran los detalles de las técnicas k-means, fuzzy c-means y mean shift. En técnicas como k-means y fuzzy c-means se requiere conocer el número de los clusters por lo que es necesario un método para determinar este parámetro; presentamos entonces un método ligero para la elección de este parámetro.

3.3.1. K-means

K-means es una técnica paramétrica para el agrupamiento y clasificación de objetos u observaciones cuya complejidad es $O(nkl)$, donde n es el número de objetos u observaciones, k es el número de agrupaciones y l el número total de iteraciones; esto hace de *k-means* una de las técnicas más empleadas para el agrupamiento y clasificación de datos, sin embargo, cuando el número de observaciones crece resulta altamente costosa.
Esta técnica realiza la partición de un espacio de observaciones basado en el criterio del error cuadrático de las características de las observaciones [32]; este algoritmo funciona correctamente cuando las muestras permanecen concentradas y suficientemente diferenciables. *K-means* realiza la asignación de los elementos dentro de *k-particiones* basado en la medida de similitud con respecto al *centroide* de las particiones; estas iteraciones se realizan hasta que no haya un cambio significativo en la similitud de las observaciones con respecto a los centroides.

Es importante resaltar que *k-means* requiere conocer, a priori, el número de agrupaciones. Esto representa uno de los principales inconvenientes de esta técnica, en particular cuando esta información se desconoce. A continuación mostraremos el funcionamiento del algoritmo *k-means* para la partición de un espacio de muestras S.

Funcionamiento del algoritmo k-means

El algoritmo particiona un conjunto S y en k particiones. Devuelve un conjunto $C = \{C_1, ..., C_k\}$ de *centroides* asignados como sigue:

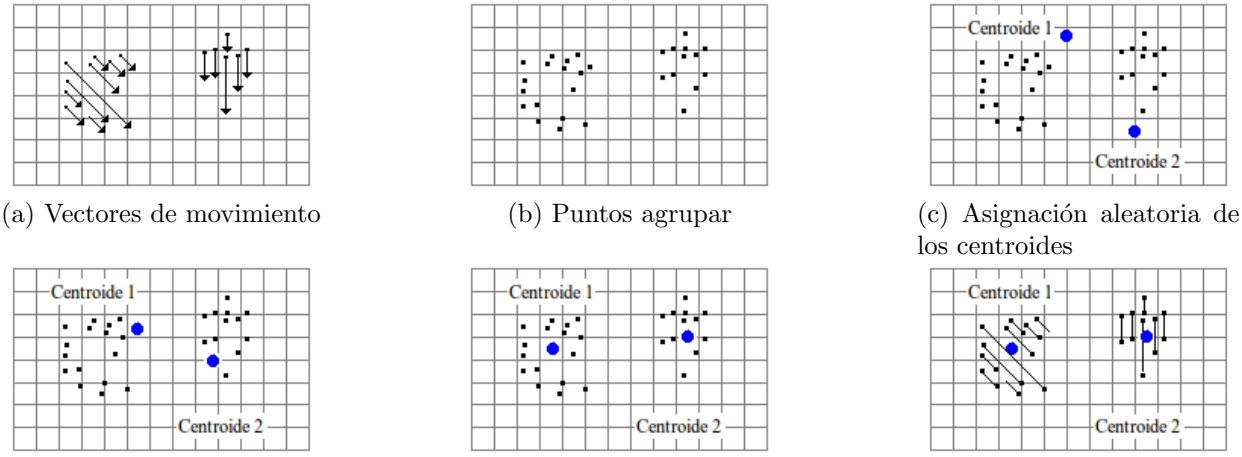
- **Paso 1**: Elige aleatoriamente k *centroides* $C_1, ..., C_k$.
- **Paso 2**: Asigna $x \in S$ al agrupamiento con *centroide* C_i tal que $1 \leq i \leq k$ cuya distancia sea menor.
- **Paso 3**: Ajusta el *centroide* para cada agrupamiento C_i tal que $1 \leq i \leq k$ de acuerdo a los elementos asignados.
- **Paso 4**: Ve a 2 hasta alcanzar la condición de convergencia.

Identificación de objetos en movimiento: K-means

En este trabajo empleamos *k-means* para la identificación de los objetos en movimiento por representar una referencia clásica para el problema de agrupamiento. La baja complejidad de esta técnica resulta eficiente cuando el flujo de objetos en movimiento es bajo, sobre todo porque el filtrado de vectores de movimiento reduce considerablemente el número de vectores a agrupar.

Empleando la representación de la información vista en la sección 5.2, tenemos que *k-means* funciona empleando la distancia Euclidiana como criterio para el cálculo de similitud entre los vectores, de esta manera los vectores más cercanos pertenecerán al mismo objeto en movimiento.

Supongamos que se tiene un cuadro con los vectores de movimiento como se muestran en la figura 3.12a. En esta imagen es posible notar dos agrupamientos de vectores (las cuales corresponden a objetos en movimiento) que comparten una misma dirección. Asumiendo que se conoce el parámetro k=*2*, el procedimiento para el agrupamiento se describe a continuación:



(a) Vectores de movimiento (b) Puntos agrupar (c) Asignación aleatoria de los centroides

(d) Cambio en la posición de los centroides (e) Centroides sin cambios significativos en su posición (f) Identificación de los objetos en movimiento

Figura 3.9: Agrupamiento con k-means

Selecciona k centroides aleatoriamente. Estos puntos representan el centroide de los objetos en movimiento (ver Figura 3.12c).

Continuando con el procedimiento *k-means* (paso 2), para cada uno de los puntos (origen/destino de los vectores de movimiento) se calcula la distancia (Euclidiana) a cada uno de los centroides, asignado a la agrupación cuya distancia al centroide sea menor. Al finalizar este paso, cada uno de los componentes de los vectores de movimiento pertenece a una agrupación (objetos en movimiento).

El procedimiento para reajustar los k centroides se realiza sobre tomando la media de las distancias de los componentes de los vectores con respecto al centroide (ver Figura 3.12f). Este procedimiento (paso 2 y 3) se realiza hasta alcanzar un criterio de convergencia; en este trabajo empleamos los siguientes criterios:

- Número máximo de iteraciones. Este criterio permite controlar el tiempo máximo que se dedica a la identificación de los objetos para cada cuadro del GOP. En este trabajo

se emplea un criterio de 250 iteraciones como máximo.

- Cambio despreciable en la posición. Considerando dos iteraciones consecutivas, para las cuales la diferencia en la posición del centroide es menor a 10 unidades, el algoritmo *k-means* finaliza para el cuadro actual.

Al finalizar este procedimiento *k-means* tenemos *k-agrupaciones* las cuales son candidatas para ser considerados objetos en movimiento en el cuadro actual (ver Figura 3.9f).

Ventajas y desventajas de k-means para la identificación de objetos

- *K-means* presenta una baja complejidad, calculada en función del número de vectores de movimiento y agrupaciones. Esto cobra relevancia cuando se considera que es posible emplear *k-means* cuando los flujos de objetos en movimiento son bajos y mantienen una distancia significativa entre ellos.

- La fiabilidad de los resultados que se obtienen con esta técnica dependen en gran medida de la distancia existente entre los objetos en movimiento. Cuando los objetos se encuentran lo suficientemente alejados, estos son identificados con una alta precisión (debido a que la distancia entre los vectores de cada objeto es mayor a la existente entre los vectores de cada objeto); caso contrario ocurre cuando los objetos tienen una alta proximidad.

- Uno de los criterios que repercute en el número de iteraciones es la asignación inicial de los centroides, ya que al ser asignados de manera próxima al agrupamiento de los vectores se reduce el número de iteraciones, caso contrario ocurre cuando esta asignación presenta una distancia lejana al agrupamiento de vectores.

- El principal inconveniente de esta técnica es el requerir el parámetro k, el cual representa en este trabajo, el número de objetos en movimiento. En principio, esta información no

es conocida, a priori, por lo que es necesario un método adicional para determinar este parámetro.

3.3.2. Fuzzy c-means

Como vimos en la sección anterior, k-means opera sobre un conjunto finito de datos. El proceso de agrupamiento se realiza asignando cada uno de estos datos a un agrupamiento de los k existentes. Este tipo de agrupamiento obtiene resultados adecuados cuando se asume que los conjuntos son disjuntos entre sí, sin embargo, en la práctica existen problemas en que los elementos a agrupar no pueden ser claramente asignados a un agrupamiento en particular. Este problema está presente en el agrupamiento realizado por el algoritmo k-means, el cual no emplea ningún criterio que refleje la incertidumbre con que los datos son agrupados.
El algoritmo fuzzy c-means (FCM) representa una técnica de agrupamiento difusa siendo ampliamente utilizado debido a que permite conocer la ambigüedad con que los datos son agrupados [33][34]. Esta ambigüedad (fuzziness) refleja el hecho de que un elemento del conjunto de datos no pueda ser claramente asignado a un agrupamiento y descartado del resto. En la práctica, el grado de fuzziness de un elemento del conjunto de datos esta dado por un coeficiente, por lo que, un elevado valor en este coeficiente significa que el elemento tiene una alta ambigüedad en su asignación.
El algoritmo iterativo FCM constituye una técnica paramétrica para la optimización de la función objetivo mostrado a continuación:

$$J(U, c_1, ..., c_c) \; = \; \sum_{i=1}^{c} \sum_{j=1}^{n} u_{i\,j}^{m} d_{i\,j}^{2} \tag{3.6}$$

Donde c es el número de clústeres esperados $(c_1, \ldots, c_c)$. En la práctica, se suele iniciar con un valor $c = 2$ para la partición del conjunto de datos. Dado que es posible conocer la calidad de los resultados para un valor de $c = 2$, es posible incrementar de manera progresiva el valor de c a fin de encontrar el optimo para la función 3.6.
U es la matriz de pertenencia de tamaño nxc cuyos coeficientes son asignados al inicio del algoritmo. En este sentido, el elemento u_{kl}, $1 \leq k \leq c$, $1 \leq l \leq n$, representa el grado de

fuzziness que tiene el $i-esimo$ dato respecto a la agrupación c_j. m es el cociente que controla el grado de pertenencia. Un valor mayor de m significa una mayor característica difusa en la asignación de los datos, caso contrario ocurre cuando el valor de m es reducido. En la literatura, típicamente se suelen emplear valores para $m = 1{,}0, 1{,}25, 2{,}50$. d_{ij}^2 es la medida de similitud definida, esta medida se basa en el criterio del error cuadrático.

Funcionamiento del algoritmo fuzzy c-means

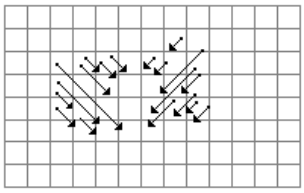
El algoritmo (FCM) inicia calculando los centros para cada uno de los c agrupamientos. Continua calculando para cada punto del conjunto de datos el grado de fuzziness que presenta para cada uno de los agrupamientos. A partir de estos datos, es posible identificar los datos de mayor relevancia para cada uno de los agrupamientos. Este es uno de los aspectos de importancia en la elección de las técnicas para la identificación de objetos ya que permite tener un marco para la asignación de los vectores de movimiento, sobre todo cuando la distancia entre los objetos en la escena de video es reducida, por lo que tenemos que un vector de movimiento puede pertenecer. A continuación se muestra el algoritmo fuzzy c-means:

- **Paso 1**: Se fija el número de clústeres, $2 \leq c \leq n$. Donde n es el número de datos a agrupar. Inicializa la matriz de pertenencia U. Inicializa el coeficiente de pertenencia m.
- **Paso 2**: Calcular los centroides de los $c-clusteres$ empleando la medida de similitud.
- **Paso 3**: Actualizar la matriz U con los valores de pertenencia calculados.
- **Paso 4**: Se realiza la comparación de los valores de la matriz calculada en la iteración actual y anterior. Si $\left\|U^{(i+1)} - U^{(i)}\right\| \leq \epsilon_F$ el algoritmo termina. En caso contrario ve al paso 2.

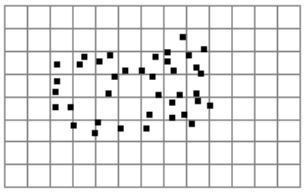
En el algoritmo, ϵ_F es un criterio de particionado para los centroides de los clústeres. Este valor típicamente se fija como 0,01.

Identificación de objetos en movimiento: fuzzy c-means

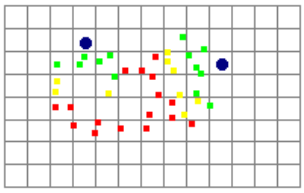
En este trabajo, se emplea fuzzy c-means para la identificación de objetos debido a que permite conocer el grado de ambigüedad con que se agrupan los vectores de movimiento. Este aspecto se vuelve relevante cuando la distancia entre los objetos es reducida. La figura 3.12a muestra dos agrupamiento de vectores que corresponden a dos objetos en movimiento con una dirección de desplazamiento distinta, esto es posible notarlo por la orientación que presentan los vectores de movimiento. Sin embargo, resulta claro que cuando la distancia de los objetos (ver figura 3.12b) es reducida, la tarea para el agrupamiento no resulta claro.



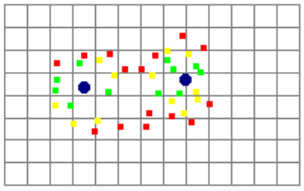
(a) Vectores de movimiento



(b) Puntos agrupar



(c) Asignación aleatoria de los centroides



(d) Cambio en la posición de los centroides

Figura 3.10: Agrupamiento con fuzzy c-means

El algoritmo fuzzy c-means inicia eligiendo la ubicación de los centroides de los clústeres (ver figura 3.12c). En este sentido, el algoritmo fuzzy c-means mantiene un alto parecido al funcionamiento descrito en k-means, sobre todo considerando que se emplea el mismo criterio de similitud. En este trabajo empleamos $m = 1{,}5$ y un valor $\epsilon_F = 0{,}001$.

Una vez que el algoritmo ha realizado la asignación de cada punto al centroide correspondiente, se calcula el grado fuzziness como se puede observar en la figura 3.12c. En esta figura, los elementos que corresponden altamente a los centroides de los clústeres están marcados de color verde, por otro lado, a medida que el grado fuzziness aumenta, los elementos toman un

color amarillo llegando hasta un tono rojo que representa que los elementos son asignados con un alto grado de ambigüedad.
Mientras no se alcance el criterio de convergencia, se reasigna el centroide de los clústeres hasta que se tiene un bajo grado de ambigüedad para la mayoría de los elementos, esto se muestra en la figura 3.12f.

Ventajas y desventajas de fuzzy c-means para la identificación de objetos

- De manera similar a k-means, fuzzy c-means requiere conocer el número de clústeres a identificar. Lo cual representa una limitante en esta técnica.
- Contrario a k-means, fuzzy c-means permite conocer la calidad con la que los vectores de movimiento son asignados. Esto permite eliminar aquellos vectores que no proporcionan información útil para la identificación de los objetos en movimiento.
- Esta técnica tiene una complejidad que la hacen una opción viable para la identificación de los objetos en movimiento. Sobre todo considerando que es posible implementarlo para tareas que operen en tiempo real.

3.3.3. Mean Shift

Hasta este momento, hemos tratado exclusivamente técnicas paramétricas que requieren conocer el número de agrupaciones a reconocer. Como se ha mencionado, esto representa una limitante para los sistemas automatizados en el conteo de objetos. Debido a eso, en esta sección presentamos una técnica que supera esa limitante.
Mean shift constituye una técnica no paramétrica presentada por primera vez en [22]. Esta técnica, en contraste a k-means y fuzzy c-means, no requiere conocer la información acerca del número de agrupaciones o la distribución de los datos. Esta es la principal característica y la razón de estudio de este método de agrupamiento para el conteo de objetos. Además de esto, mean shift recientemente ha sido estudiada en tareas de visión por computadora para el análisis del espacio de colores y seguimiento de caras [31].

La idea principal detrás de mean shift es tratar el problema de agrupamiento de un espacio d-dimensional como una función de densidad de probabilidad [30], donde las regiones corresponden a los máximos, estos puntos máximos son interpretados como el centroide de los objetos. Para cada punto de los datos por agrupar, se realiza un procedimiento para calcular el gradiente de la función de densidad hasta su convergencia. A continuación se presenta la idea general de fuzzy c-means, no se profundiza en los detalles pero se enfatizan las cualidades de esta técnica en la identificación de objetos.

Funcionamiento del algoritmo mean shift

Sea un conjunto $X = \{x_1, ..., x_n\}$ el conjunto de n muestras en un espacio Euclidiano *d-dimensional* R^d. Se define la norma de $x \in X$ es un número no negativo $\|x\|^2 = \sum_{i=1}^{n} |x_i|^2$. Una funcion $K : X \to R$ tal que $K(x) = k(\|x\|^2)$, se dicen entonces que K es un kernel si ademas:

1. k es no negativo.
2. $k(a) \geq k(b)$ si $a < b$.
3. k es contínuo por segmentos y $\int_0^\infty k(r)dr < \infty$.

A partir de esta definición es posible definir el algoritmo mean shift. Sea $S \subset X$ un conjunto finito, en este caso, S es el conjunto de muestras por agrupar. Sea K un kernel y la función $w : S \to 0, \infty$. La media de las muestras con kernel k para $x \in X$ esta definido como:

$$\hat{f}_K = \frac{1}{nh^d} \sum_{i=1}^{n} K\left(\frac{x - x_i}{h}\right) \tag{3.7}$$

Donde h define el radio del kernel k denominado parametro de *ancho de banda*. La simetria radial para la simetria del kernel esta definido como:

$$K(x) = c_k K(\|x\|^2) \tag{3.8}$$

Donde c_k representa una constante de normalizacion. Es posible calcular el gradiente de la funcion de densidad de la ecuacion 3.7 como:

$$\frac{2C_{k,d}}{nh^{d+2}} \underbrace{\left[\sum_{i=1}^{n} g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)\right]}_{termino\ a} \underbrace{\left[\frac{\sum_{i=1}^{n} x_i g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)}{\sum_{i=1}^{n} g\left(\left\|\frac{x-x_i}{h}\right\|^2\right)} - x\right]}_{termino\ b} \tag{3.9}$$

Donde $g(x) = -k^{'}(x)$ denota la derivada del kernel. El *termino a* es proporcional a la estimación de x calculado con el kernel k. El *termino b* es conocido como el vector m *mean shift*, el cual indica la direcion de mayor incremento en la densidad. El procedimiento *mean shift* para cada punto x_i esta dado como:

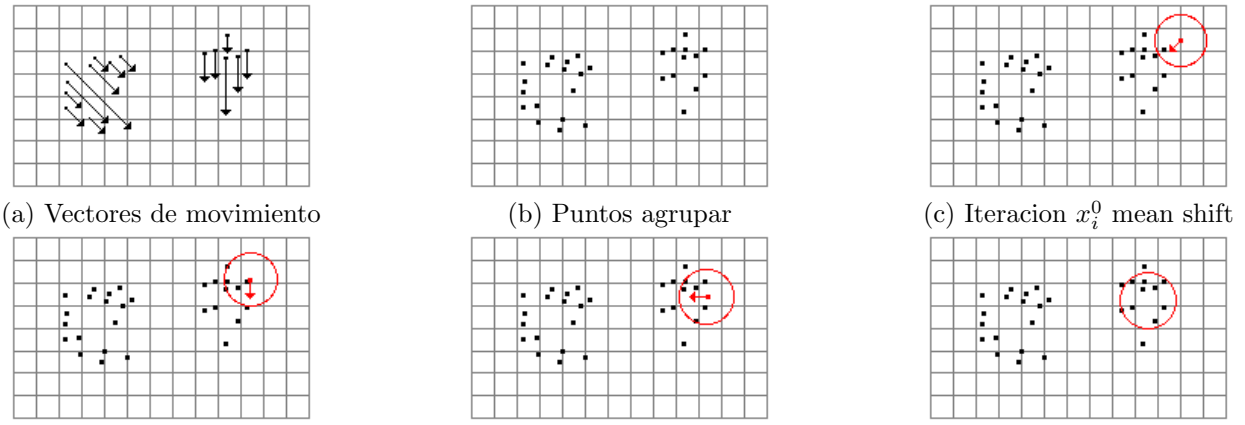
1. Computa el vector mean shift $m(x_i^t)$.
2. Translada el centroide de acuerdo a la estimacion de la densidad: $x_i^{t+1} = x_i^t + m(x_i^t)$
3. Se realizan los pasos 1 y 2 hasta la convergencia, es decir, $\nabla f(x_i) = 0$.

Figura 3.11: Ancho de banda para el algoritmo mean shift.

Identificación de objetos en movimiento: mean shift

Para la elección del *ancho de banda* se realizó un promedio acerca de la proporción de los objetos que se deseaba identificar. En este caso, se calculó el promedio del radio de kernel que mejor se adaptaba a las personas en movimiento; en la figura 3.11 se puede ver un ancho

de banda de 105 unidades(pixeles). Este parámetro limita el tipo de objetos que manejara esta técnica de manera adecuada. Respetando la interpretación de los vectores que se ha empleado en este trabajo, el procedimiento mean shift es muestra en la 3.12.



(a) Vectores de movimiento (b) Puntos agrupar (c) Iteracion x_i^0 mean shift

(d) Iteracion x_i^1 mean shift (e) Iteracion x_i^2 mean shift (f) Convergencia $\nabla f(x_i) = 0$ para x_i

Figura 3.12: Agrupamiento con mean shift

Ventajas y desventajas de mean shift para la identificación de objetos

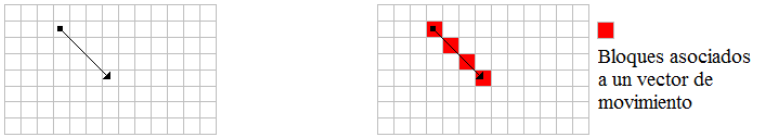
- La principal ventaja de mean shift es el identificar de manera automática el numero de agrupaciones existentes en base al parámetro del ancho de banda.
- El principal inconveniente de esta técnica es la complejidad de la técnica, lo que representa un inconveniente para un sistema que opere en tiempo real.
- El ancho de banda limita el tipo de objeto que es posible identificar, sin embargo, es posible realizar una adaptación de este parámetro, lo que otorga un alto grado de robustez para la identificación de objetos.

3.4. Agrupamiento por bloques

En esta parte de nuestro trabajo, presentamos una propuesta para la identificación de objetos en movimiento basado en el agrupamiento de bloques. Esta propuesta está inspirada en

las técnicas de codificación. Las técnicas de codificación actuales eliminan la redundancia temporal que se presenta entre cuadros consecutivos dentro de la secuencia de video, para esto emplean una ventana de rastreo, de tal manera que un elemento de la imagen k es buscado en la imagen $k+1$ a fin de codificar solo el cambio de posición de este elemento. Esta es la base para la técnica de agrupamiento basada en bloques.

Esta técnica recibe su nombre al realizar un agrupamiento tomando como base la forma en que se segmenta la imagen, en donde los pixeles son agrupados en unidades denominadas macrobloques. En la norma de codificación H.264, el tamaño de los macrobloques varia de manera significativa, comparado con normas de codificación anteriores, tales como MPEG-1, MPEG-2, H.261 y H.264, por lo que es posible encontrar macrobloques de 4x4, 8x4, 4x8, 8x8 y 16x16. Aunque en el agrupamiento propuesto se emplea un tamaño de bloques de $16x16$, es posible cambiar este parámetro sin representar un cambio significativo en la propuesta del algoritmo; la variación de este parámetro permite manipular la granularidad con que el contorno de los objetos es identificado. Sin embargo, se eligió un tamaño de bloque de $16x16$ por estar presente en otras normas de codificación, lo que permitiría emplear esta propuesta en estos estándares.



(a) Vector de movimiento

(b) Región cubierta por un vector de movimiento

Figura 3.13: Aglomeración de vectores de movimiento

Esta técnica no paramétrica, dado que no requiere conocer el número de agrupaciones a identificar, presenta ventajas considerables respecto a las técnicas mostradas para la identificación de objetos en movimiento, siendo la principal, la simplicidad que presenta. Sin embargo, para la identificación de los objetos es necesario conocer la proporción (área) que toman dentro de la escena de video. Esta información es empleada como un criterio de filtrado para los objetos a identificar. Resulta importante resaltar, que en esta técnica se alcanzan resultados óptimos cuando el proceso de filtrado de vectores de movimiento elimina gran parte de la

información que no corresponde a objetos en movimiento.

Funcionamiento del agrupamiento basado en bloques

La idea detrás de nuestra propuesta es la interpretación de un vector de movimiento como un conjunto de bloques, los cuales corresponden a la imagen segmentada, por lo que cada uno de estos tiene un índice asociado, a partir del cual es posible conocer su ubicación dentro de la imagen. La figura 3.13b muestra los bloques asociados a un vector. Este tipo de interpretación permite manejar la información de los vectores de movimiento como un conjunto de índices:

$$mvindex = \{b_1, \ldots, b_2\} \tag{3.10}$$

Donde *mvindex* es un conjunto de índices de bloques en el que cada b_i corresponde a un bloque ubicado dentro de la imagen (ver figura 3.14). La cardinalidad del conjunto *mvindex* está determinada por la longitud del vector de movimiento.

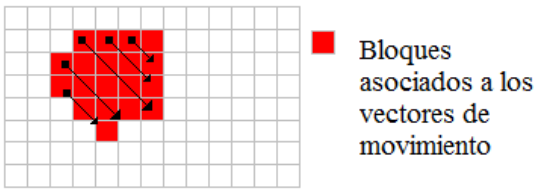


Figura 3.14: Agrupamiento de vectores de movimiento basado en bloques

Resultaría natural pensar que este tipo de interpretación provoca el agrupamiento de un gran número de vectores, sobre todo considerando el caso en el que se presenten vectores de movimiento que atraviesen toda la escena de video, sin embargo, esto último no ocurre, dado que las técnicas de codificación realizan la generación de los vectores de movimiento en base a un tamaño determinado de la ventana de rastreo, este tamaño, sin embargo no puede cubrir toda la escena de video.

Esta interpretación se realiza para todos los vectores de movimiento que cumplen el criterio de filtrado definido (en base a la magnitud). Es importante resaltar que el orden de generación

de estos bloques se basa en la forma en que se decodifican los vectores de movimiento. Por lo que cada uno de los vectores se transforma en un conjunto de bloques que definirán una agrupación nueva o se anexarán a una ya existente. Para esto es necesario definir:

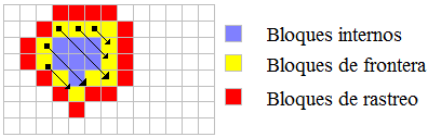
$$amv = \{agmv_1, \ldots, agmv_k\} \tag{3.11}$$

Donde amv define un conjunto de agrupaciones encontradas en el cuadro de video, donde cada uno de los elementos $amgv_i$ es una agrupación de $mvindex$. La tarea de agrupamiento opera sobre amv y los $mvindex$ identificados de la siguiente manera:
Esto se realiza de la siguiente manera:

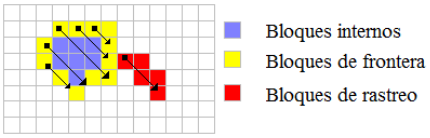
- Una vez decodificado el *i-ésimo* vector de movimiento, este es interpretado como un $mvindex$.
- La tarea de agrupamiento se alcanza comparando la posición de los bloques del $mvindex$ generado con las agrupaciones del conjunto amv. Esto se logra definiendo bloques de rastreo para cada una de las agrupaciones $agmv_j$, $1 \leq j \leq m$, donde m es la cardinalidad del conjunto amv. Estos bloques de rastreo se generan sobre los *bloques de frontera* de las agrupaciones. Los *bloques de frontera* corresponden a aquellos ubicados en los límites de las agrupaciones, como se muestra en la figura 3.15a. Es importante recordar que $agmv_j$ y $mvindex$ son un conjunto de índices, por lo que proceso de comparación de los bloques de rastreo con el $mvindex$ consiste simplemente en un cotejo de índices.
- Si la comparación de bloques resulta positiva, esto es, los bloques del $mvindex$ se ubican próximos a los bloques del $agmv_j$, los bloques del $mvindex$ se agregan al $agmv_j$ como se muestra en la figura 3.15c. En caso contrario, el $mvindex$ se considera un $amgv_i$ por lo que se agrega al conjunto amv.

Filtrado de agrupaciones

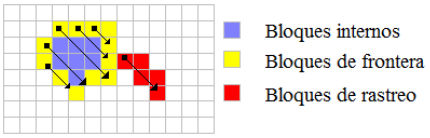
Resulta claro que el procedimiento anterior considera incluso agrupaciones que no corresponden a objetos en movimiento. Pensemos por un momento en los vectores de movimiento que



(a) Bloques de frontera de las agrupaciones



(b) Comparación de los bloques de un vector de movimiento con una agrupación



(c) Agrupamiento de bloques

Figura 3.15: Procedimiento del agrupamiento de bloques

son producidos, por ejemplo, por el movimiento de la cámara. Es posible que estos vectores sobrepasen el criterio de la magnitud definido para el filtrado de vectores, en cuyo caso formarán una agrupación *amgv* sin tratarse de un objeto en movimiento. Esta es la principal razón, por la que es necesario un proceso de filtrado para las agrupaciones identificadas.

El proceso de filtrado para las agrupaciones toma en consideración el área que cubren los objetos dentro de la escena de video. Por lo que es posible emplear esta información como un criterio para definir la densidad de las agrupaciones. En este momento es importante considerar que el área de las agrupaciones está estrechamente ligada al número de bloques para cada agrupación, por lo que una agrupación con una cantidad reducida de bloques tiene un área pequeña, caso contrario cuando las agrupaciones presentan una cantidad mayor de bloques.

El número de bloques que forman las agrupaciones sirve además, para identificar aquellas en las que se trata de más de un objeto en movimiento. En nuestra propuesta, analizamos imágenes en la que se presentan personas de diversas complexiones, de tal manera que es posible definir un área promedio de 95 bloques para las personas. Esto permite tratar el hecho de que dos o más personas presenten una distancia reducida, de tal manera que la agrupación de bloques sea un número próximo a un múltiplo de 95, en donde es posible identificar el número de personas. Además de que agrupaciones que tengan un número de

bloques considerablemente menor a 95 sean eliminados.

Ventajas y desventajas del agrupamiento

- La principal ventaja del método propuesto para el agrupamiento de bloques lo constituye la simplicidad para el procesamiento de los datos y el manejo de la información, dado que solo se representa una agrupación como un conjunto de índices.
- La técnica propuesta no requiere conocer el número de agrupaciones presentes en la escena de video, además de que el criterio de filtrado permite aumentar la fiabilidad con la que los objetos son identificados.
- El principal inconveniente de este técnica, para la identificación de objetos, es el hecho de necesitar el parámetro del área que es cubierta por lo objetos. Lo cual requiere de un análisis estadístico de las características de los objetos en movimiento. Sin embargo, esto también brinda un alto grado de adaptación para la identificación de objetos de diversa naturaleza.

3.4.1. Elección rápida del parámetro K y C

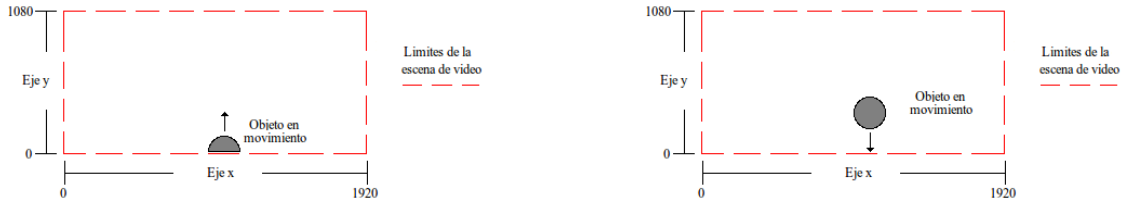
Como se mostró, las técnicas *k-means* y *fuzzy c-means* requieren conocer el número de clústeres a identificar (representados en los parámetros k y c). Este es el principal inconveniente para el funcionamiento automático de estas técnicas, dado que no sé conoce el número de objetos en movimiento presentes en la secuencia de video.
En este sentido, se suele asumir que las técnicas *k-means* y *fuzzy c-means* operan identificando más de un clúster, esto porque se trata de técnicas de agrupamiento y para el caso de un solo clúster resulta trivial, pues todos los elementos pertenecerán al único clúster existente. Aunque esto resulta cierto en una gran cantidad de casos, para la identificación de objetos sobre secuencias de video, no se cumple este criterio, pues aun cuando se hable de un solo objeto, es necesario calcular la ubicación de su centroide. Más aun, no es posible asegurar

que en la secuencia de video se presenta solo un objeto en movimiento, lo que representa un grave impedimento para el funcionamiento de las técnicas *k-means* y *fuzzy c-means*.
Además de esto, es necesario considerar los siguientes fenómenos que provocan variaciones en el número de objetos presentes:

- Entrada de objetos en la escena: La aparición de nuevos objetos dentro de la escena se considera cuando estos se encuentra ubicados en las fronteras espaciales de la escena de video, además de la orientación de su desplazamiento. Considerando por ejemplo un video con una resolución de $1920x1080$ como el mostrado en la figura 3.16a, consideramos a un objeto entrante sé se encuentra ubicado en los limites espaciales de la escena y la orientación de su desplazamiento lo conduce hacia el centro de la escena.

- Salida de objetos de la escena: De manera similar a la entrada de objetos, esta ocurre cuando un objeto se encuentra ubicado en las fronteras espaciales de la escena de video debido a la orientación de su desplazamiento (ver figura 3.16b).

Resulta claro que el número de objetos durante la secuencia de video no permanece constante. Entonces es necesario un método para determinar el número de objetos para cada cuadro que contenga vectores de movimiento. Aunque existen técnicas para determinar los parámetros de *k-means* y *fuzzy c-means* estas tiene un alto costo computacional lo cual afecta el rendimiento de los sistemas para el conteo de personas, los cuales deben operar en tiempo real.



(a) Entrada de un objeto (b) Salida de un objeto

Figura 3.16: Aparición de objetos en la escena de video

En esta parte de nuestro trabajo, presentamos un método ligero para la elección de los parámetros k y c. En nuestra propuesta empleamos el orden de decodificación de los bloques

para cada cuadro (ver figura 3.17), el cual se realiza iniciando en la parte superior izquierda del cuadro continuando hacia parte derecha-inferior.

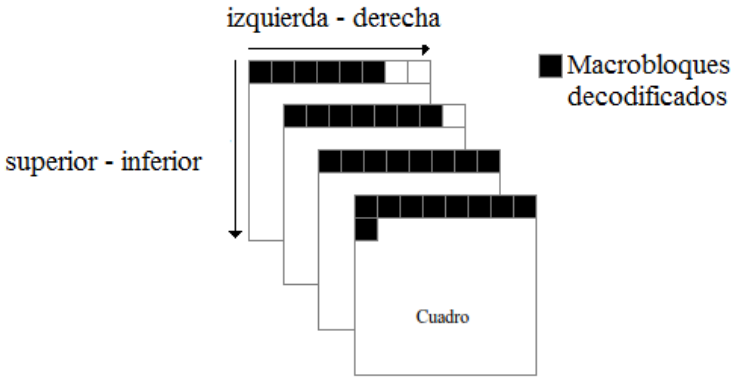


Figura 3.17: Orden de decodificación de los macrobloques

El método propuesto realiza un agrupamiento rápido de los vectores de movimiento, de tal manera que es posible conocer el número de clústeres presentes en cada cuadro. Para esto, una vez que se encuentra un vector de movimiento que cumple los criterios de filtrado se define una *ventana de rastreo*. En este sentido, la *ventana de rastreo* opera de manera similar a la definida en el método de identificación basado en bloques visto en la sección 3.4, sin embargo, en este caso, el contorno que se obtiene de los objetos no está definido con una exactitud alta, por lo que el método requiere una cantidad reducida de recursos para el manejo de la información.

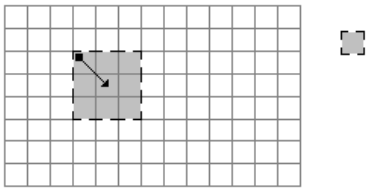


Figura 3.18: ventana de rastreo

En el proceso para determinar el número de objetos en movimiento se define $CMV = cmv_1, \ldots, cmv_m$ como el conjunto de agrupaciones de vectores de movimiento. En teoría, estos conjuntos tienen una alta probabilidad de corresponder a objetos en movimiento. El

siguiente procedimiento es realizado para todos los vectores de movimiento que cumplen el criterio de filtrado:

- **Creacion/Ajuste de agrupaciones de vectores**: El vector de movimiento decodificado es buscado dentro de cada una de las agrupaciones del conjunto CMV. Esto se realiza comparando la localización del punto de origen y destino del vector de movimiento con respecto a las coordenadas espaciales de la ventana de rastreo de las agrupaciones del conjunto CMV.

  - **Caso 1**: Si el origen o destino del vector de movimiento se ubica en el área de la ventana de rastreo de la agrupación cmv_i, se considera que el vector de movimiento pertenece a la agrupación cmv_i. En caso de ser necesario se ajusta la ventana de rastreo de la agrupación de vectores de tal manera que el área cubra todos los vectores de movimiento.
  - **Caso 2**: En caso contrario, es decir, el vector de movimiento no se localiza en la ventana de rastreo se define una ventana de rastreo de acuerdo y se agrega al conjunto CMV.

El procedimiento descrito permite conocer el número de agrupaciones de vectores de movimiento, sin embargo debido a que es posible encontrar agrupaciones de vectores que no correspondan a objetos de movimiento es necesario un proceso de filtrado que elimine tales agrupaciones. El proceso de filtrado toma como base el área total de las agrupaciones, esto permite adaptar este sistema para la identificación de objetos de diversa naturaleza.

Ventajas y desventajas del método para la elección rápida de los parámetros k y c

- Este procedimiento permite conocer el número de agrupaciones de vectores de movimiento con un bajo costo computacional debido a la simplicidad del método sin alcanzar el reconocimiento del centroide de los objetos en movimiento.

- Los resultados para la identificación de los objetos son adecuados cuando los flujos de objetos en movimiento son bajos.

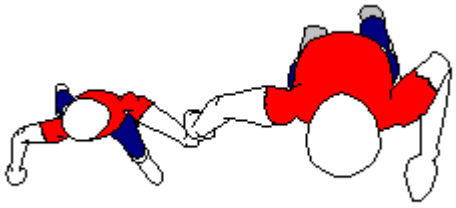
- Dado que el método realiza un filtraje de las agrupaciones de vectores de movimiento empleando el criterio del área que estos cubren, se reduce la cantidad de información (vectores de movimiento) para las técnicas de agrupamiento *k-means* y *fuzzy c-means*.

- Es posible obtener una aproximación de los centroides de los objetos para las técnicas *k-means* y *fuzzy c-means* considerando el área de las agrupaciones identificadas. Esto permite reducir el número de iteraciones empleados para las técnicas empleadas en la identificación de objetos. Esto representa la principal ventaja del método presentado. *E*l método arroja resultados incorrectos cuando el flujo de objetos en la secuencia de video es alta. Esta es la principal desventaja del método para la elección rápida del parámetro c y k.

3.5. Seguimiento de los objetos en movimiento

Aunque las técnicas de identificación de objetos en movimiento permiten conocer el número de objetos para cada cuadro que contenga vectores de movimiento, esto no permite realizar un conteo del número total de objetos que atraviesan el área definida por la escena de video. Sobre todo, considerando que el número de objetos cambia por la entrada y salida de estos. Además de esto último, existen casos en los que se afecta la continuidad de los objetos. Por ejemplo, supongamos que en la escena de video aparecen dos objetos como se muestra en la figura 3.19a, estos objetos mantiene una distancia constante durante su trayectoria. Si alguno de los objetos se queda estático por un momento, este no generará vectores de movimiento durante este lapso, por lo que no las técnicas de identificación de objetos no podrán identificarlo, aun cuando este no haya salido de la escena de video, en este caso es importante conocer la ultima ubicación de este objeto y el cuadro donde desapareció. Continuando con el ejemplo, supongamos que finalizado el período en el que el objeto se mantiene estático, este retoma su trayectoria por la escena de video, es necesario recuperar

la última posición conocida de este objeto, esto permite evitar contabilizar en más de una ocasión un objeto.



(a) Objetos en movimiento

(b) Vectores de movimiento generados

Figura 3.19: Seguimiento de objetos

Bajo estas mismas consideraciones, es posible encontrar cuadros, en donde el número de vectores de movimiento se reduzca afectando el cálculo del centroide de los objetos identificados. Esto se debe, principalmente a la naturaleza de las técnicas de codificación. Supongamos que en dos cuadros consecutivos la única diferencia entre estos, es el movimiento generado por una porción del objeto (como lo sería el movimiento de un brazo), por lo que los vectores de movimiento presentes corresponderán a la porción del objeto en movimiento afectando el cálculo del centroide del objeto identificado. En este caso, el centroide del objeto se ubicará en la porción que presenta movimiento.

Esto muestra la necesidad de un método para el seguimiento de los objetos. En esta sección se muestra el método desarrollado para el seguimiento de objetos, este método se basa en predicciones de la ubicación de los objetos en cuadros futuros, basado en características como la velocidad, posición y orientación de los objetos. En este trabajo introducimos un proceso de predicción para asegurar la continuidad de los vectores de movimiento. Este proceso considera

los siguientes aspectos:

- Dimensiones de los objetos: Para cualquier objeto en movimiento, el número de vectores de movimiento que estos producen está relacionado con sus proporciones dentro de la escena de video. Por lo que objetos pequeños (en comparación con las dimensiones del video) generan un número reducido de vectores de movimiento.

- Dirección promedio de los vectores de movimiento: Los objetos en movimiento guardan una orientación que puede ser tratada en un espacio Euclidiano. Esta dirección es mantenida por la mayoría de los vectores de movimiento.

- Magnitud promedio de los vectores de movimiento. La magnitud de los vectores de movimiento está relacionada con la velocidad de los objetos, mientras que los objetos que llegan a un estado de reposo mantienen una magnitud baja, los objetos con una alta velocidad presentan vectores de movimiento con una magnitud cercana a las dimensiones de los objetos.

El sistema de predicción permite mantener una continuidad en la trayectoria de los objetos, aun cuando estos no aparezcan de manera consecutiva en los cuadros. Este proceso debe tomar en cuenta la ubicación espacial de los objetos dentro de la escena de video, adicionalmente se almacena la información de todos los objetos encontrados durante la reproducción de la secuencia.

3.5.1. Parámetros para la predicción de movimiento

Existen parámetros que permiten conocer la forma en que se desplazan los objetos dentro de la escena de video. Este tipo de información representa un modelo temporal para el seguimiento de los objetos e incrementa la precisión del conteo. Podemos describir estos parámetros como:

- Latencia del objeto: Introducimos un coeficiente que refleja la continuidad de los objetos en cuadros consecutivos. El coeficiente de este parámetro esta normalizado, de tal manera que un valor cercano a 1 indica que los objetos han aparecido de manera consecutiva en los cuadros presentados hasta el momento.

- Coeficiente de avance normalizado: Este parámetro indica la razón de cambio esperada entre cuadros consecutivos de acuerdo a la velocidad que los objetos guardan, es decir, la magnitud entre la distancia de los objetos en cuadros consecutivos.

- Orientacion del objeto: El conocer cuando los objetos se encuentran cercanos a las fronteras de la escena de video permite conocer la salida o entrada de los objetos, lo que permite tener un conteo preciso. Esto se logra principalmente, conociendo la ubicación y orientación de los objetos. Es posible conocer la orientación que tienen los objetos considerando la orientación de la mayoría de los vectores de movimiento. Este parámetro es empleado para el método de predicciones

3.5.2. Identificación y extracción de parámetros de los objetos en movimiento

Una vez que un objeto ha sido identificado, con alguna de las técnicas estudiadas hasta el momento, es posible conocer la información de estos objetos. Esta información constituye los parámetros para el método de predicción y podemos definirlos como:

$$\mu mo_i = \frac{\sum_{j=1}^{n} |mv_j|}{n-1} \tag{3.12}$$

$$\mu do_i = \frac{\sum_{j=1}^{n} \theta mv_j}{n-1} \tag{3.13}$$

Donde la ecuación 3.12 permite conocer la magnitud promedio de los vectores de movimiento. n es el número total de vectores de movimiento y mv_j es el *j-ésimo* vector de movimiento que pertenecen al *i-ésimo* objeto denotado como μmo_i. Esta ecuación realiza la suma de las magnitudes de todos los vectores de movimiento que corresponden a un objeto. Por otro parte, la ecuación 3.13 permite conocer la orientación promedio de los vectores de movimiento; donde $thetamv_j$ es la orientacion que presenta el *j-ésimo* vector. La orientación de un vector de movimiento se calcula como el ángulo que forma, tomando el origen del vector como el punto de intersección de las dos semirectas (ver figura 3.20).

Además de esta información que es calculada para cada cuadro donde se presentes objetos en

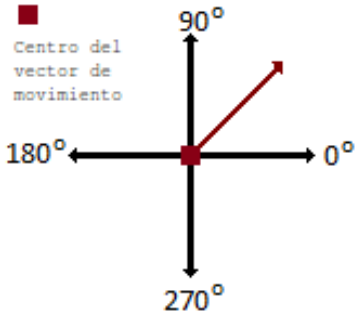


Figura 3.20: Extracción del ángulo para un vector de movimiento

movimiento, las técnicas para la identificación de los objetos devuelve el centroide de estos. Este tipo de información permite conocer la ubicación de los objetos dentro de la escena y estimar la orientación que sigue, sin embargo, esto no incorpora el seguimiento de los objetos dado que no se conoce la información de la trayectoria descrita.

3.5.3. Latencia de los objetos

El conocer la información de los objetos para cada cuadro permite realizar un seguimiento de estos interpolando la información de todos los cuadro de la secuencia. Para esto definimos un conjunto de la siguiente manera:

$$CO = \{O_1, \ldots, O_k\}$$

Donde CO es un conjunto donde se almacenan los objetos identificados y O_i es el *i-ésimo* objeto encontrado durante el proceso de decodificación de la secuencia de video. Es importante notar que los objetos en este conjunto no son encontrados en el mismo cuadro.
Cada uno de los objetos encontrados tiene los siguientes parámetros que operan para el seguimiento de los objetos:

$$\Gamma O_i = \frac{1}{(f_l - f_a)} \tag{3.14}$$

$$\Delta O_i = \frac{\sum_{i=1}^{n-1}(CentroideO_{i+1} - CentroideO_i)}{k} \tag{3.15}$$

Donde la ecuación 3.14 permite calcular el coeficiente de latencia que indica la continuidad entre los cuadros, f_l es el índice del último cuadro donde aparece el objeto y f_a es el cuadro actual en el proceso de decodificación. Este coeficiente permite conocer la continuidad con que los objetos aparecen a fin de definir un tiempo de vida para los objetos. Es posible notar que cuando la diferencia entre el último cuadro f_l y el cuadro actual decodificado f_a el coeficiente ΓO_i disminuye, lo que representa que la latencia del objeto ha disminuido por falta de continuidad en los cuadros.

Por otra parte, ΔO_i es la distancia promedio de entre los centroides de los objetos y k es el número de cuadros en la secuencia de video. Esto permite conocer la distancia promedio de desplazamiento entre cuadros consecutivos. Mediante esta información es posible generar una predicción de la ubicación del centro del objeto para los cuadros posteriores como se muestra en la figura 3.21.

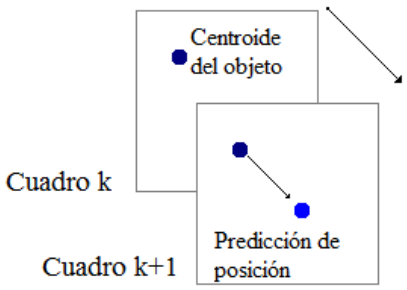


Figura 3.21: Seguimiento de los objetos en base a estimaciones de movimiento

3.5.4. Venta de rastreo

Aunque es posible generar una predicción de la posición de los objetos en cuadros posteriores, la probabilidad de que esta predicción corresponda al centroide encontrado por las técnicas de identificación es baja. Debido a esto, nosotros introducimos el manejo de *ventana de rastreo para la predicción de los objetos*. La ventana de rastreo es un área dentro de la escena de video, en la que se espera encontrar el centroide de un objeto después de presentarse un desplazamiento en cuadros posteriores. Debido a que la continuidad de los objetos no está asegurada entre cuadros consecutivos, el área debe tomar en cuenta los últimos datos

acerca de la orientación, desplazamiento y latencia promedio de los objetos en movimiento.

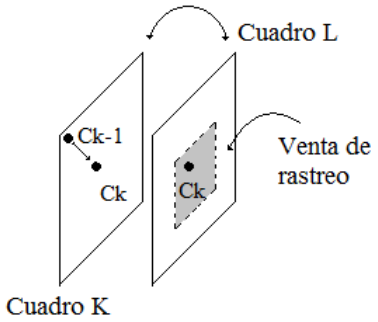


Figura 3.22: Ventana de rastreo generada para el seguimiento de un objeto.

Las dimensiones de la ventana de rastreo se calculan en base a la siguiente ecuación:

$$d_x = M_x \, \Delta O_i \, (l_f - a_c) \tag{3.16}$$

$$d_y = M_y \, \Delta O_i \, (l_f - a_c) \tag{3.17}$$

$$M_x = \mu do_i \, Mod \, 180^o \tag{3.18}$$

$$M_y = \mu do_i \, Mod \, 90^o \tag{3.19}$$

Donde d_x es la dimensión de la ventana de rastreo, ΔO_i es el desplazamiento promedio del objeto (ecuación 3.15), f_l es el último cuadro que se tiene registrado donde aparece el objeto y f_a es el cuadro actual, μdo_i es la dirección promedio del objeto (ecuación 3.13). M_x y M_y son los coeficientes que indican el sentido x y y de la ventana de rastreo de mayor crecimiento, lo que está determinado por la orientación del objeto.

3.5.5. Seguimiento de los objetos

Los elementos descritos anteriormente son empleados para la generación de un sistema de seguimiento para los objetos en movimiento. A continuación se muestra el proceso realizado para esta tarea, este procedimiento se realiza para cada cuadro donde existan objetos en movimiento.

Si las técnicas de identificación encuentran un objeto con centroide $CentroideO_i$ en el cuadro k, este será buscado en cada uno de los objetos del conjunto CO, definido en la sección 3.5.3.

El proceso de búsqueda verifica si la posición del $CentroideO_i$ del objeto identificado se ubica dentro de la ventana de rastreo de algún objeto O_i del conjunto CO, cuya cardinalidad es m. Cada elemento del conjunto CO corresponde a objetos identificados en cuadros anteriores. En este paso es posible identificar dos casos:

- **Caso 1**: El con centroide $CentroideO_i$ del objeto se ubica en la ventana de rastreo del objeto O_j con $1 \leq j \leq m$. En este caso, O_j es actualizado con la información descrita en la sección 3.5.3.

- **Caso 2**: El objeto no corresponde a ningún elemento del conjunto CO. En este caso, se verifica que las coordenadas de $CentroideO_i$ se encuentren próximas a las fronteras de la escena de video, en cuyo caso el objeto se agrega al conjunto CO.

Capítulo 4
Base experimental

En este capítulo presentamos el desarrollo de una base experimental para el conteo de objetos sobre flujos de video codificado. Esto es motivado por la falta de un marco experimental, que cubra las características de nuestro trabajo. Aunque en la literatura se suele trabajar con secuencias de video ampliamente conocidas, estas están enfocadas principalmente a la verificación de las técnicas desarrolladas para la codificación de video, empleando secuencias CIF, QCIF, SQCIF y 4CIF, cuyas resoluciones no corresponden a la alta definición presente en los sistemas de telemonitores actuales.

La base experimental está integrada por un conjunto de secuencias de video codificadas bajo la norma H.264. Esta secuencias de video muestra una variedad de objetos, tantos personas (adultos e infantes) como objetos rígidos (carreolas, plataformas, mochilas entre otros). Para cada una de estas secuencias, se realizó un proceso de etiquetado manual a fin de identificar los centroides de los objetos para todos los cuadros de cada una de las secuencias de video. Esto permite comparar el proceso realizado por un persona contra un sistema automatizado para la identificación de objetos. Por otra parte, este proceso de etiquetado permite conocer la trayectoria descrita por los objetos.

Uno de los aspectos principales que motivaron la generación de esta base experimental es la falta de resultados cuantitavos para el conteo de objetos. El número de propuestas que emplean una verificación formal de sus métodos es reducido, limitándose a una validación visual de sus resultados para la segmentación de las imágenes. Esta base experimental que se presenta tiene por los siguientes objetivos:

- Extracción de la información para las técnicas de codificación. La propuesta presentada

en este trabajo se basa en el uso de flujos de video codificado, para lo cual, es necesario contar con un número considerable de secuencias de video para la extracción de los vectores de movimiento.

- La verificación de los métodos de filtrado de vectores de movimiento. Dado que los valores definidos para el criterio de filtrado son resultado de un análisis estadístico de los vectores resultantes. Esto se realizo empleando las secuencias de video aplicando los métodos de filtrado.

- La validación y comparación de los métodos para la identificación de objetos en movimiento, proceso que se realiza para cada cuadro que conforma la secuencia de video. Esto permite cuantificar las ventajas de tales métodos a fin de ser capaces de seleccionar la mejor opción para un sistema de conteo de personas. Esta cuantificación de los resultados se realiza mediante la comparación de los centroides encontrados por las técnicas contra los datos etiquetados en la base de datos.

- Busca validar el método de seguimiento de los objetos presentes en la secuencias. La base experimental constituyo la base para la comprobación del método

- Identificar los fenómenos presentes en las secuencias de video que afectan el rendimiento de las técnicas presentadas. El contar con una base experimental que integre una diversidad en cuanto al tipo de objetos, además de fenómenos como la iluminación permite identificar aquellos errores presentes en las técnicas.

De lo anterior, es clara la amplia utilidad de una base de datos. En lo que resta del capítulo, se describe el proceso realizado para la recopilación de las secuencias, se muestra los criterios empleados para el proceso de anotaciones sobre las imágenes a partir de los cual se extraen las rutas de cada objeto y el tipo de objetos que se cubren.

4.1. Características de las secuencias de video

Las secuencias de video fueron creadas empleando una cámara que se coloco arriba y apuntando hacia abajo con respecto al flujos de objetos. Como se ha mencionado, esto permite reducir el efecto de fenómenos como las oclusiones. Estas secuencias fueron grabadas en un ámbito exterior con fuente de iluminación natural, debido a lo cual, los objetos proyectan una sombra visible en la escena de video empleando una video cámara que arroja flujos de video codificado. El conjunto de secuencias comparten características particulares, dentro de las cuales podemos destacar:

- Resolución: 1920x1080 FullHD
- Posición de la cámara: Top View
- Norma de codificación: H.264 AVC
- Perfil de codificación: Baseline
- Modo: Progresivo
- Cuadros por segundo: 25

4.2. Creación de la base de videos

Las secuencias de video muestran diferentes tipos de objetos por lo que a continuación se muestra una breve descripción del tipo de objetos en las secuencias de video y la trayectoria que estos describen, además de que se enfatiza las peculiaridades existentes. En estas secuencias se muestran un objetos clasificando estos como rígidos (objetos que no tienen partes móviles en su estructura) y personas. La tabla 4.1 sintetiza el tipo de objetos y el número total de cuadros para cada secuencia de video.

- **Secuencia 1**: La secuencia muestra una persona que aparece desde la parte superior de la escena de video hacia la parte inferior, esta persona no arrastra ningún objeto.

- **Secuencia 2**: La secuencia muestra una persona que aparece desde la parte superior de la escena de video hacia la parte inferior, esta persona no arrastra ningún objeto.
- **Secuencia 3**: La secuencia muestra una persona que aparece desde la parte superior de la escena de video hacia la parte inferior, esta persona no arrastra ningún objeto.
- **Secuencia 4**: La secuencia muestra una persona que aparece desde la parte superior de la escena de video hacia la parte inferior, esta persona no arrastra ningún objeto.
- **Secuencia 5**: La secuencia muestra una persona que aparece desde la parte superior de la escena de video hacia la parte inferior, esta persona no arrastra ningún objeto.
- **Secuencia 6**: La secuencia muestra una persona que aparece desde la parte superior de la escena de video hacia la parte inferior, esta persona no arrastra ningún objeto.
- **Secuencia 7**: La secuencia muestra dos personas que aparecen en la parte superior de la escena de video. Una persona tras otra, desaparecen de la escena de video.
- **Secuencia 8**: La secuencia muestra una persona que aparece en la parte superior de la escena de video. La persona no arrastra ningún objeto.
- **Secuencia 9**: La secuencia muestra una carreola empujada por una persona desde la parte superior hasta la parte inferior de la escena de video.
- **Secuencia 10**: La secuencia de video muestra una carreola siendo empujada por una persona desde la parte lateral izquierda hacia la parte lateral derecha de la escena de video.
- **Secuencia 11**: La secuencia de video muestra una carreola siendo empujada por una persona desde la parte inferior hacia la parte superior de la escena de video. La persona no describe una trayectoria rectilínea.
- **Secuencia 12**: La secuencia de video muestra una carreola siendo empujada por una persona desde la parte inferior derecha describiendo una trayectoria en zigzag. Ob-

servamos un tercer objeto que corresponde a una persona en sentido contrario al que describe la carreola, empujada por la primer persona.

- **Secuencia 13**: La secuencia de video muestra una carreola siendo empujada por una persona desde la parte superior de la escena de video hasta la parte inferior, la persona describe una trayectoria en zigzag.

- **Secuencia 14**: La secuencia de video muestra un total de tres objetos. Una de las personas presentes en la secuencia de video corresponde a una persona empujando una carreola. Un tercer objeto se mueve en sentido contrario a la trayectoria de la carreola.

- **Secuencia 15**: La secuencia de video muestra tres personas en movimiento que aparecen en la parte superior de la escena con dirección hacia parte inferior. Se identifica una persona adulta y dos infantes. Hasta la parte media de la escena de video los dos infantes mantienen una distancia reducida hasta el grado de tocarse hombro a hombro. La persona adulta acarrea en la mano un objeto pequeño que en cuadros de la secuencia se oculta parcial o totalmente. Uno de los infantes juega con lo que parece ser una vara de madera. La trayectoria de los infantes es distinta al de la persona adulta, en la cual se puede observar una trayectoria rectilínea.

- **Secuencia 16**: La secuencia de video muestra tres personas en movimiento que aparecen en la parte inferior de la escena con dirección hacia la parte superior. Identificando una persona adulta y dos infantes. La persona adulta camina al ritmo de uno de los infantes y carga un objeto en la mano derecha, mientras que el segundo infante aparece hasta la desaparición de las otras personas. El segundo infante aparece corriendo en la secuencia de video. La trayectoria del segundo infante es rectilínea mientras corre.

- **Secuencia 17**: La secuencia de video muestra tres personas en movimiento que aparecen en la parte central derecha de la escena con dirección a la parte central izquierda. Identificando una persona adulta y dos infantes. La persona adulta carga un objeto en la mano derecha mientras que no camina al ritmo de los infantes. Los infantes

describen trayectorias distintas al adulto que describe una trayectoria rectilínea. Las personas detienen su movimiento por aproximadamente un minuto para proseguir con su trayectoria.

- **Secuencia 18**: La secuencia de video muestra a seis personas en movimiento que aparecen en la parte central superior de la escena con dirección a la parte central inferior de la escena. Es posible reconocer a cuatro personas adultas y dos infantes. Dos de las personas adultas llevan sujetadas por la mano a un infante. Las personas caminan por parejas. Debido a la posición de la cámara y a las dimensiones de los infantes, a estos sufren en algunos cuadros un ocultamiento parcial por el cuerpo de los adultos. Dos de las personas adultas llevan cargando mochilas lo que deforma su silueta. La trayectoria de los infantes es rectilínea debido a que se ve forzada por los adultos. Los infantes marcan pasos más pronunciados debido a la diferencia de velocidad con respecto a los adultos.

- **Secuencia 19**: La secuencia de video muestra un objeto rígido (auto a control remoto) con una trayectoria rectilínea desde la parte superior de la escena hacia la parte inferior. El objeto no es empujado por algún objeto externo. Describe movimientos de avance y retroceso sobre su trayectoria.

- **Secuencia 20**: La secuencia de video muestra un objeto rígido (auto a control remoto) con una trayectoria diagonal desde la parte superior de la escena de video. El objeto no es empujado por algún objeto externo. Describe movimientos de avance y retroceso sobre su trayectoria.

Secuencia	# Cuadros	# Objetos	Personas	Otros
1	146	1	1	0
2	149	1	1	0
3	130	1	1	0
4	104	1	1	0
5	125	1	1	0
6	110	2	2	0
7	153	2	2	0
8	153	1	1	0
9	300	2	1	1
10	294	2	1	1
11	198	2	1	1
12	211	3	2	1
13	157	2	1	1
14	167	2	1	1
15	420	3	3	0
16	469	3	3	0
17	1090	3	3	0
18	780	4	4	0
19	984	1	0	1
20	504	1	0	1

Tabla 4.1: Características de las secuencias de video

4.2.1. Anotaciones sobre imágenes

El proceso de anotaciones sobre las imágenes, además de identificar el tipo y número de objetos, constituye la base experimental para las técnicas de identificación y seguimiento de objetos. El proceso de anotación consiste en identificar el centroide de los objetos presentes en cada cuadro de la secuencia de video. Este proceso es realizado por una persona, la cual ubica, para cada cuadro de las secuencias, el centroide de los objetos. La figura 4.1a muestra el cuadro 56 de la secuencia de video en el que se aprecia una persona en movimiento, por otra parte, la figura 4.1b muestra el centroide anotado para este cuadro.

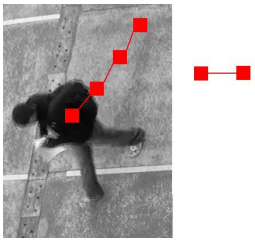
Resulta importante mencionar que en el caso de los objetos con partes móviles (brazos y piernas) es necesario contemplar las deformaciones que sufren durante su trayectoria, dado que en muchos casos, estas deformaciones repercuten en los vectores de movimiento. Sin embargo, para los objetos que no sufren deformaciones en su forma, este tipo de consideraciones se omiten.

Dado que se cuenta con la información de los centroides de los objetos para cada cuadro de

la secuencia de video, es posible generar la trayectoria que describe el movimiento de cada uno de los objetos. Para esto basta considerar la secuencia ordenada de centroides registrados para cada cuadro como una sucesión de puntos, de esta manera, la figura 4.1c

(a) Cuadro 56 de la secuencia de video 3

(b) Centroide reconocido para el objeto del cuadro 56



(c) Ruta descrita por los objetos

Figura 4.1: Anotaciones sobre imágenes

Capítulo 5
Experimentación

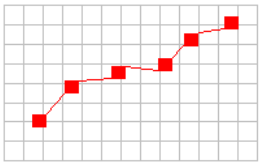
Las pruebas que presentaremos fueron realizadas para cada una de las secuencias de video contenidas en la base de videos. La experimentación presentada a continuación, se enfoca en dos tareas principales:

- Identificación: Se muestran los resultados obtenidos para la identificación de objetos empleando k-means, fuzzy c-means, mean shift y agrupamiento por bloques.
- Seguimiento: Verificamos el funcionamiento del método propuesto para el seguimiento de objetos.

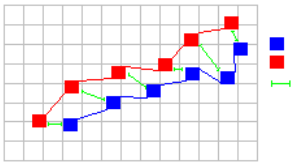
El proceso para evaluar los resultados, requiere definir métricas que permitan cuantificar estos de manera objetiva. Mientras que en la literatura se suelen emplear simplemente resultados visuales, en nuestro trabajo, tomamos como base la propuesta presentada en [20], donde los autores definen métricas para los métodos de seguimiento de objetos.

5.1. Análisis para la experimentación

La idea básica de la experimentación consiste en comparar las trayectorias que se obtienen mediante las técnicas presentadas en este trabajo y las recopiladas en la base de videos. Aunque bien, no es posible asegurar que las trayectorias contenidas en la base de videos tiene una precisión absoluta, si es posible tomar estas trayectorias como un marco experimental. Sobre todo, considerando que las variaciones de las trayectorias identificadas por más de una persona no tienen una diferencia considerable.



(a) Ejemplo de una trayectoria



(b) Comparación de trayectorias

Figura 5.1: Métricas para el seguimiento de objetos

Para esto, podemos definir una trayectoria como una secuencia de posiciones que varían sobre la línea del tiempo. De acuerdo a esto, se generaliza la definición de una trayectoria T como una secuencia de posiciones (x_i, y_i) sobre la línea del tiempo como:

$$T = \{(x_1, y_1, t_1), (x_2, y_2, t_2), ..., (x_n, y_n, t_n)\} \tag{5.1}$$

Este tipo de definición varia cuando se habla de video, donde suelen manipular cuadros (imágenes ordenadas), por lo que es posible omitir t_i, dado que esta se puede obtener conociendo el número de imagen dentro de la trayectoria. Debido a esto, podemos reescribir la ecuación 5.1 como:

$$T = \{(x_1, y_1), (x_2, y_2), ..., (x_n, y_n)\} \tag{5.2}$$

La figura 5.1a muestra un trayectoria descrita por un móvil. Una de las características importantes de la definición anterior es que permite la comparación de dos trayectorias. Para esto podemos considerar segunda trayectoria T_B, por lo que es posible calcular la diferencia (distancia) d_i con respecto a la trayectoria T_A en el *i-esimo* punto como:

$$d_i = |d_i| = \sqrt{(p_i - x_i)^2 + (q_i - y_i)^2} \tag{5.3}$$

Donde $(p_i, q_i) \in T_B$ y $(x_i, y_i) \in T_A$. De lo anterior, d_i es la distancia Euclidiana que existe entre los dos puntos de las trayectorias T_A y T_B. La figura 5.1b muestra la comparación posición a posición de las trayectorias T_A y T_B.

5.2. Métricas para la comparación de trayectorias

De la sección anterior, podemos mencionar métricas que se suelen emplear para la comparación de trayectorias en la visión por computadora. Para esto, se define $D(T_A, T_B)$ como el conjunto de n distancias d_i encontradas para cada punto de las trayectorias T_A y T_B. Entre las principales métricas podemos mencionar:

$$\textbf{Media} \quad \mu(\, D(T_A, T_B)\,) = \frac{1}{n}\sum_{i=1}^{n} d_i \tag{5.4}$$

$$\textbf{Mediana} \quad \text{median}(\, D(T_A, T_B)\,) = \begin{cases} d_{\frac{n+1}{2}} & \text{si n es impar} \\ \frac{1}{2}(d_{\frac{n}{2}} + d_{\frac{n}{2}+1}) & \text{en otro caso} \end{cases} \tag{5.5}$$

$$\textbf{Desviación estándar} \quad \sigma(\, D(T_A, T_B)\,) = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(d_i - \mu(d_i))^2} \tag{5.6}$$

$$\textbf{Mínimo} \quad min(\, D(T_A, T_B)\,) = \text{el menor } d_i \tag{5.7}$$

$$\textbf{Máximo} \quad max(\, D(T_A, T_B)\,) = \text{el mayor } d_i \tag{5.8}$$

Estas métricas proveen información estadística acerca del error de los métodos de seguimiento. Por una parte, la media provee información acerca de la distancia promedio entre las trayectorias T_A y T_B, mientras que la desviación estándar indica la variación de la distancia esperada respecto a la media. En estas métricas, se espera que el valor de la mediana sea menor al de la media, ya que esta última considera los valores máximos y mínimos. Por último, el valor mínimo es el d_i menor de todo el conjunto $D(T_A, T_B)$, caso contrario, ocurre con el valor máximo.

5.3. Resultados experimentales

El objetivo es comparar las trayectorias resultantes de las técnicas y las extraídas en la base de datos, sin embargo, hay que considerar el hecho de que la base de datos contiene información de todos los cuadros que conforman una secuencia de video mientras que las técnicas funcionan exclusivamente sobre aquellos cuadros que contienen vectores de movimiento. Debido a

esto, la cantidad de posiciones en las trayectorias detectadas por las técnicas es estrictamente menor al número que conforman las trayectorias en la base de videos. Además, para cada secuencia de video se realiza la extracción de vectores de movimiento para cada cuadro, esto permite controlar la experimentación, por lo que es posible aislar los casos en los que las técnicas suelen presentar problemas.

El primer resultado que presentamos es la comparación de trayectorias generadas por tres personas, cada una de estas personas realizó la anotación del centroide de los objetos en movimiento para cada cuadro en movimiento sobre la secuencia de video 2 (tabla 5.1) y 19 (tabla 5.2) de la base de videos. Por una parte, en la secuencia 2 podemos encontrar una persona en movimiento, mientras que en la secuencia 19 se encuentra un objeto rígido.

Es posible notar que la diferencia en el trazado de las trayectorias para objetos con partes móviles (en este caso, personas) presenta una mayor diferencia que los resultados obtenidos para objetos rígidos. Esto se debe principalmente al hecho de que es posible identificar con mayor fiabilidad el centro de un objeto que no presenta deformaciones.

	$\mu(D(T_A,T_B)$	$median(D(T_A,T_B)$	$\sigma(D(T_A,T_B)$	$min(D(T_A,T_B)$	$min(D(T_A,T_B)$
Persona 1 - Persona 2	15.95	7.65	18.41	7.58	24.9
Persona 1- Persona 3	14.34	12.70	17.54	3.89	20.45

Tabla 5.1: Trayectorias descritas por objetos con partes moviles

	$\mu(D(T_A,T_B)$	$median(D(T_A,T_B)$	$\sigma(D(T_A,T_B)$	$min(D(T_A,T_B)$	$min(D(T_A,T_B)$
Persona 1 - Persona 2	9.65	9.56	12.84	1.29	14.78
Persona 1 - Persona 3	7.60	8.04	10.90	0.73	13.85

Tabla 5.2: Trayectorias descritas por objetos rigidos

Tomando esto como referencia, a continuación se muestran los resultados obtenidos por las técnicas *k-means*, *fuzzy c-means*, *mean shift* y *agrupamiento por bloques*. Las secuencias de video empleadas presentan un solo objeto en la escena de video. Como se ha mencionado anteriormente, aunque pareciera trivial pensar en la identificación de un solo objeto en movimiento, esto cobra importancia al considerar que, en general, ningún tipo de información de los objetos se conoce.

Los resultados anteriores muestran que en general, mean shift obtiene mejores resultados

	$\mu(D(T_A,T_B)$	$median(D(T_A,T_B)$	$\sigma(D(T_A,T_B)$	$min(D(T_A,T_B)$	$max(D(T_A,T_B)$
K-means	35.49	31.89	39.29	8.00	96.74
Fuzzy c-means	37.56	35.01	38.85	9.32	76.09
Mean shift	33.58	40.31	36.32	8.06	58.52
Bloques	40.76	35.63	40.29	12.45	89.94

Tabla 5.3: Secuencia de video 1

	$\mu(D(T_A,T_B)$	$median(D(T_A,T_B)$	$\sigma(D(T_A,T_B)$	$min(D(T_A,T_B)$	$max(D(T_A,T_B)$
K-means	36.49	40.56	50.86	14.09	95.83
Fuzzy c-means	38.09	55.30	41.92	2.23	98.01
Mean shift	35.31	45.96	39.30	4.00	70.51
Bloques	30.29	37.46	37.35	5.34	71.34

Tabla 5.4: Secuencia de video 2

cuando se trata de un solo objeto en movimiento. En estos casos, podemos notar que la técnica para la *identificación rápida del parámetro k y c* tiene un alto grado de efectividad. Además, podemos ver que las técnicas *k-means*, *fuzzy c-means*, y *agrupamiento por bloques* presenta resultados cercanos a *mean shift*.

Los resultados mostrados para secuencias de video donde se observa un solo objeto (una persona en este caso) suelen estar próximos a los registrados por la trayectoria identificada por una persona, sobre todo considerando las resoluciones de los videos. Las técnicas empleadas tanto para la generación del número de objetos, la identificación del centroide de estos y el seguimiento presentan resultados cercanos entre sí.

Uno de los aspectos obligatorios a considerar, es el funcionamiento de las técnicas en presencia de un número mayor de objetos. A continuación se presentan los resultados para secuencias de video con más de un objeto.

	$\mu(D(T_A,T_B)$	$median(D(T_A,T_B)$	$\sigma(D(T_A,T_B)$	$min(D(T_A,T_B)$	$max(D(T_A,T_B)$
K-means	30.63	36.65	34.65	98.02	1.00
Fuzzy c-means	31.87	35.87	33.97	99.03	3.24
Mean shift	28.81	35.00	34.43	96.67	2.23
Bloques	31.38	34.29	39.81	98.67	9.56

Tabla 5.9: Secuencia de video 6

	$\mu(D(T_A,T_B)$	$median(D(T_A,T_B)$	$\sigma(D(T_A,T_B)$	$min(D(T_A,T_B)$	$max(D(T_A,T_B)$
K-means	42.95	70.06	45.47	6.00	98.72
Fuzzy c-means	45.67	70.34	48.34	11.32	91.40
Mean shift	44.49	61.08	49.01	7.28	98.79
Bloques	46.87	65.89	50.03	4.06	78.33

Tabla 5.5: Secuencia de video 3

	$\mu(D(T_A,T_B)$	$median(D(T_A,T_B)$	$\sigma(D(T_A,T_B)$	$min(D(T_A,T_B)$	$max(D(T_A,T_B)$
K-means	33.37	20.50	36.89	3.00	89.00
Fuzzy c-means	30.85	27.94	35.03	4.56	90.81
Mean shift	33.67	24.62	38.01	3.16	94.04
Bloques	30.91	32.04	35.46	2.49	86.33

Tabla 5.6: Secuencia de video 4

	$\mu(D(T_A,T_B)$	$median(D(T_A,T_B)$	$\sigma(D(T_A,T_B)$	$min(D(T_A,T_B)$	$max(D(T_A,T_B)$
K-means	33.69	30.56	37.60	21.84	87.82
Fuzzy c-means	29.43	27.19	32.94	16.27	63.78
Mean shift	28.19	25.14	28.63	20.09	49.09
Bloques	31.67	29.71	29.67	12.84	56.00

Tabla 5.10: Secuencia de video 7

	$\mu(D(T_A,T_B)$	$median(D(T_A,T_B)$	$\sigma(D(T_A,T_B)$	$min(D(T_A,T_B)$	$max(D(T_A,T_B)$
K-means	61.48	56.09	68.09	10.56	73.05
Fuzzy c-means	56.86	50.46	67.57	13.48	78.89
Mean shift	37.09	34.57	45.56	11.47	58.09
Bloques	42.93	36.45	50.63	14.67	59.45

Tabla 5.11: Secuencia de video 9

	$\mu(D(T_A,T_B)$	$median(D(T_A,T_B)$	$\sigma(D(T_A,T_B)$	$min(D(T_A,T_B)$	$max(D(T_A,T_B)$
K-means	62.34	58.72	78.34	11.34	99.99
Fuzzy c-means	63.34	45.03	64.86	14.09	89.34
Mean shift	48.23	39.51	54.78	7.23	70.43
Bloques	44.58	40.37	50.48	6.49	68.59

Tabla 5.12: Secuencia de video 10

	$\mu(D(T_A,T_B)$	$median(D(T_A,T_B)$	$\sigma(D(T_A,T_B)$	$min(D(T_A,T_B)$	$max(D(T_A,T_B)$
K-means	73.87	69.96	85.93	19.34	91.74
Fuzzy c-means	69.34	54.06	78.96	21.05	88.00
Mean shift	40.23	36.00	51.04	12.34	59.34
Bloques	50.45	49.34	63.49	17.34	77.02

Tabla 5.13: Secuencia de video 11

	$\mu(D(T_A,T_B)$	$median(D(T_A,T_B)$	$\sigma(D(T_A,T_B)$	$min(D(T_A,T_B)$	$max(D(T_A,T_B)$
K-means	36.64	50.39	41.38	8.91	99.10
Fuzzy c-means	33.69	47.12	38.47	3.00	98.47
Mean shift	32.60	49.64	35.48	5.00	78.72
Bloques	35.38	46.73	38.64	4.59	87.09

Tabla 5.7: Secuencia de video 5

	$\mu(D(T_A,T_B)$	$median(D(T_A,T_B)$	$\sigma(D(T_A,T_B)$	$min(D(T_A,T_B)$	$max(D(T_A,T_B)$
K-means	37.04	39.09	38.56	7.56	39.01
Fuzzy c-means	36.76	37.03	37.00	7.00	38.45
Mean shift	27.56	25.65	33.93	1.34	34.67
Bloques	21.05	27.78	25.74	3.05	27.09

Tabla 5.8: Secuencia de video 8

	$\mu(D(T_A,T_B)$	$median(D(T_A,T_B)$	$\sigma(D(T_A,T_B)$	$min(D(T_A,T_B)$	$max(D(T_A,T_B)$
K-means	41.05	56.59	53.34	14.94	86.34
Fuzzy c-means	39.04	45.29	59.37	11.09	80.31
Mean shift	36.05	29.05	45.76	7.23	60.34
Bloques	33.49	36.86	44.23	8.93	58.39

Tabla 5.14: Secuencia de video 12

	$\mu(D(T_A,T_B)$	$median(D(T_A,T_B)$	$\sigma(D(T_A,T_B)$	$min(D(T_A,T_B)$	$max(D(T_A,T_B)$
K-means	40.34	45.34	67.34	17.29	89.34
Fuzzy c-means	41.03	29.45	56.75	12.05	70.97
Mean shift	39.59	30.34	47.30	4.56	67.78
Bloques	38.00	41.34	57.06	11.93	71.60

Tabla 5.15: Secuencia de video 13

	$\mu(D(T_A,T_B)$	$median(D(T_A,T_B)$	$\sigma(D(T_A,T_B)$	$min(D(T_A,T_B)$	$max(D(T_A,T_B)$
K-means	91.73	92.34	94.34	16.78	99.99
Fuzzy c-means	87.83	90.07	95.67	14.09	99.99
Mean shift	59.87	60.04	67.87	10.99	70.03
Bloques	70.71	78.02	80.63	14.65	79.46

Tabla 5.16: Secuencia de video 14

	$\mu(D(T_A,T_B)$	$median(D(T_A,T_B)$	$\sigma(D(T_A,T_B)$	$min(D(T_A,T_B)$	$max(D(T_A,T_B)$
K-means	74.60	75.51	82.32	16.74	82.92
Fuzzy c-means	73.34	70.05	77.58	15.83	81.48
Mean shift	60.03	67.45	66.34	13.00	76.89
Bloques	56.78	56.39	59.39	12.30	69.39

Tabla 5.17: Secuencia de video 15

	$\mu(D(T_A, T_B)$	$median(D(T_A, T_B)$	$\sigma(D(T_A, T_B)$	$min(D(T_A, T_B)$	$max(D(T_A, T_B)$
K-means	91.57	95.23	99.30	16.47	99.99
Fuzzy c-means	80.46	76.36	87.40	12.48	99.09
Mean shift	56.48	51.20	67.39	9.27	71.29
Bloques	64.39	67.23	78.26	9.65	81.49

Tabla 5.18: Secuencia de video 16

	$\mu(D(T_A, T_B)$	$median(D(T_A, T_B)$	$\sigma(D(T_A, T_B)$	$min(D(T_A, T_B)$	$max(D(T_A, T_B)$
K-means	77.46	67.49	80.34	17.30	83.98
Fuzzy c-means	78.57	70.35	81.49	17.38	87.00
Mean shift	70.37	68.49	79.46	15.49	83.28
Bloques	75.34	74.40	82.59	14.57	85.09

Tabla 5.19: Secuencia de video 17

	$\mu(D(T_A,T_B)$	$median(D(T_A,T_B)$	$\sigma(D(T_A,T_B)$	$min(D(T_A,T_B)$	$max(D(T_A,T_B)$
K-means	94.78	87.59	99.99	34.56	99.99
Fuzzy c-means	91.34	78.45	97.34	20.34	99.99
Mean shift	78.94	80.34	85.49	19.23	89.75
Bloques	89.57	78.63	91.34	19.59	91.50

Tabla 5.20: Secuencia de video 18

Los resultados mostrados en las tablas muestran que las técnicas reducen su desempeño cuando el número de objetos aumenta, esto se debe principalmente a las siguientes razones:

- La fiabilidad del método para la *identificación rápida del parámetro k y c* se reduce cuando los objetos mantienen una alta proximidad.

- El número de vectores se reduce en cuadros de las secuencias de video. Además, de que los objetos presentan una alta cercanía, por lo que los vectores de movimiento no se pueden diferenciar entre los objetos.

De los resultados anteriores, podemos observar que *mean shift* arroja los mejores resultados. El *agrupamiento por bloques* presenta un mejor rendimiento que las técnicas *k-means* y *fuzzy c-means*.

Uno de los hechos en los que estábamos interesados era el funcionamiento de estas técnicas en presencia de otro tipo de objetos. Esto permite emplear este sistema para una amplia variedad de objetos. Por esto, a continuación presentamos los resultados obtenidos para la identificación de objetos para secuencias de video donde aparecen objetos con movimiento autónomo. Estas escenas de video muestran dos carros manejados a control remoto. Las tablas 5.21,5.22.

Video 19	$\mu(D(T_A,T_B)$	$median(D(T_A,T_B)$	$\sigma(D(T_A,T_B)$	$min(D(T_A,T_B)$	$max(D(T_A,T_B)$
K-means	67.04	63.05	74.40	14.59	79.40
Fuzzy c-means	68.39	69.03	70.49	16.34	85.40
Mean shift	56.37	50.62	63.62	10.14	77.34
Bloques	59.68	53.56	70.09	10.00	78.23

Tabla 5.21: Secuencia de video 19

Video 20	$\mu(D(T_A, T_B)$	$median(D(T_A, T_B)$	$\sigma(D(T_A, T_B)$	$min(D(T_A, T_B)$	$max(D(T_A, T_B)$
K-means	63.56	59.67	66.70	15.04	93.12
Fuzzy c-means	52.20	52.31	54.58	16.83	84.30
Mean shift	51.85	62.03	53.64	7.61	79.34
Bloques	71.23	63.67	81.56	17.47	91.34

Tabla 5.22: Secuencia de video 20

Los resultados mostrados en las tablas 5.21 y 5.22 permiten verificar el hecho de que las técnicas mostradas en este trabajo se adaptan para objetos de diversas naturaleza. Esto se logra variando los parámetros para la identificación de los objetos.

Capítulo 6
Conclusiones y trabajo a futuro

En este trabajo presentamos un estudio sobre las técnicas aplicadas para el conteo de objetos sobre flujos de video codificado. Se denominó a la propuesta conteo de objetos sobre flujos de video codificado. Esto es así, debido al hecho de que, aunque no se alcanzó una clasificación de los objetos contenidos en las secuencias de video, sí es posible hablar de un conteo de objetos de diversa naturaleza, siendo posible emplear las técnicas para el conteo de personas, cuando se hable específicamente de este tipo de objetos. Esto se debe a que, parámetros como la proporción de los objetos en la escena de video reflejan la naturaleza de los objetos.

En nuestro trabajo, es posible alcanzar un sistema para el conteo de objetos controlando parámetros como la posición de las cámaras, lo que permite tratar muchos de los fenómenos existentes. Este sistema se basa en dos tareas principales: identificación y seguimiento de objetos.

Para la primera, presentamos métodos para la identificación de objetos empleando exclusivamente vectores de movimiento. Esto lo logramos tratando el problema de identificación como una tarea de agrupamiento y empleando las técnicas k-means, fuzzy c-means, mean shift y agrupamiento por bloques. Esta última es una propuesta novedosa al operar a un bajo costo computacional, este método emplea la base teórica de las principales normas de codificación de video. Adicionalmente, presentamos un método para la elección rápida del parámetro en las técnicas k-means y fuzzy c-means. Esta toma retoma la idea de nuestra propuesta de agrupamiento por bloques y constituye una opción viable para el funcionamiento en conjunto con k-means o fuzzy c-means en tiempo real, ya que presenta una baja complejidad y reduce el número de iteraciones en k-means y fuzzy c-means.

Para la segunda tarea que forma parte del sistema de conteo, se desarrollo un método para el seguimiento de objetos. Este método se basa en predicciones acerca de la posición de los objetos considerando la posición y desplazamiento promedio que mantienen durante su recorrido. Este método propuesto permite incrementar la continuidad de los objetos en cuadros en los que no se presentan vectores de movimiento además de permitir manejar fenómenos como las oclusiones, este método presenta una baja complejidad, por lo que es posible emplearlo para un sistema real que opere en tiempo real.

Uno de los principales aportes de nuestro trabajo es el desarrollo de una base experimental que consiste en un número de secuencias de video codificadas bajo la norma H.264, la cual permite simular las características de los principales sistemas de telemonitoreo. Esto se debió al hecho de no poder encontrar una base de videos que se adaptara a las secuencias de video actuales ni a la posición de la cámara. Todas las secuencias de video fueron etiquetadas para conocer el número de objetos y la posición del centroide de cada una de estas, lo que permite realizar una comparación cuantitativa de los métodos empleados para la identificación de los objetos. La existencia de la base de videos permite mostrar resultados cuantitativos en nuestra propuesta. Mean shift fue la técnica que obtuvo mejores resultados para nuestra experimentación, sin embargo, debido a la alta complejidad computacional de la técnica, no es posible asegurar que esta sea la mejor propuesta para un sistema que opere en tiempo real. Es importante recordar que nuestra propuesta para la identificación de objetos presenta en muchos casos mejores resultados que k-means y fuzzy c-means además de estar muy próximo a los resultados obtenidos con mean shift. Si bien, nuestra propuesta obtiene buenos resultados cuando se habla de objetos que guardan una distancia que permite diferenciar claramente los vectores de cada una de las agrupaciones, esta también reduce su eficacia cuando aumenta el número de objetos.

6.1. Desarrollo

Para la elaboración de este trabajo fue necesario un profundo estudio de las técnicas de codificación de video, específicamente se centro en la norma H.264 la cual presenta las mejores características en cuanto a compresión y manejo de errores. Fue necesario el estudio de un decodificador de video H.264, el cual se modificó para las necesidades de nuestro trabajo, esto es, para la extracción de vectores de movimiento y coeficientes DCT.

Las técnicas k-mean, mean shift y fuzzy c-means fueron desarrolladas en un principio en Matlab 7.7 (2008b) por las ventajas que presenta en cuanto a la agilidad de desarrollo de los algoritmos. Esto permitió realizar una experimentación controlada sobre las secuencias de video. Sin embargo, dado que el costo computacional se ve incrementado por el uso de Matlab, nos dimos a la tarea de migrar estas técnicas al lenguaje C/C++ una vez verificada su eficacia para la identificación de objetos. Caso similar ocurrió con la técnica de agrupamiento por bloques, la cual fue implementada en un principio en lenguaje Python para su posterior migración al lenguaje C/C++; caso similar ocurre con el método de seguimiento de objetos.

Para la generación de la base de datos se desarrollo una aplicación en lenguaje Java que permitiera generar, a partir de la secuencia de video, un conjunto de imágenes sobre las cuales se realizo la anotación de las trayectorias. Esto fue motivado por la gran cantidad de horas que requería esta actividad, además de que reduce el tiempo empleado para la extensión de la base de videos.

6.2. Trabajo a futuro

Durante la realización de este trabajo de investigación surgieron una cantidad de propuestas sobre las que es posible trabajar a futuro. Estas tareas pueden aumentar la fiabilidad del sistema para el conteo de objetos y son resultados de las observaciones durante la experimentación:

1. Filtrado adaptativo de vectores de movimiento. En este trabajo se presento un filtrado de vectores de movimiento que toma como parámetro, la magnitud de estos. Sin em-

bargo, este tipo de filtrado elimina incluso vectores de movimiento de los objetos, lo que reduce la cantidad de información necesaria para el proceso de agrupamiento. Bajo esta línea, es posible investigar acerca de un proceso de filtrado adaptativo que permita eliminar los vectores de movimiento excepto aquellos de los objetos en movimiento lo cual permitiría tener agrupaciones densas de vectores.

2. Uso de color para el seguimiento de los objetos. El uso de la información de los colores permitiría aumentar la fiabilidad del seguimiento de objetos ya que estos datos permanecen constantes durante su recorrido en la escena de video. Esto se puede representar con el uso de los coeficientes DCT sin incrementar considerablemente la cantidad de información.

3. Clasificación de los objetos. Uno de los principales alcances en la visión por computadora es la clasificación de objetos sobre secuencias de video. En este trabajo, aunque no se alcanzo una clasificación del tipo de objetos presentes, surgieron propuestas que es posible estudiar para su aplicación en la clasificación de objetos. Entre estas, podemos mencionar la idea de emplear la diferencia entre las trayectorias generadas por los objetos rígidos y con partes móviles, por una parte, los objetos rígidos presentan trayectorias uniformes, mientras que los objetos con partes móviles deforman su contorno lo que genera trayectoria con variaciones marcadas.

4. Adaptación automática para diversos objetos. Las técnicas empleadas en este trabajo pueden adaptarse a las características de varios objetos, lo que hace al sistema robusto. Sin embargo, es necesario trabajar sobre un método de aprendizaje automático. Esta es una línea sobre la que se debe investigar.

Además de esto, una de las líneas de investigación que se pretende estudiar, es la generación de un sistema embebido en equipo de telemonitoreo. Para esto, es necesario considerar el tiempo de ejecución de cada una de estas propuestas a fin de encontrar la mejor opción para un conteo de objetos en tiempo real.

Apéndice A
Sistema para el conteo de objetos sobre secuencias de video

Dado que en los capítulos anteriores se enfatiza dos tareas principales: la identificación y el seguimiento de los objetos en movimiento, a continuación de muestra el sistema desarrollo para el conteo de objetos, el cual integra los elementos descritos anteriormente. Como se ha mencionado, aunque no se alcanza un sistema capaz de clasificar los objetos entre personas u otros, si es posible realizar un conteo de estos limitando las condiciones, es decir, el tipo de objetos en movimiento presentes en la secuencias de video. Nuestra propuesta consta de ocho módulos sin incluir el módulo que opera sobre el decodificador de video, la combinación de estos permite realizar el conteo de objetos en movimiento. El diagrama a bloques del sistema propuesto se presenta en la figura y en lo que resta se describe el funcionamiento de cada uno de sus componentes.

Decodificación-Filtrado VM

El módulo de *Decodificación VM* se desarrollo tomando como base las especificaciones de la norma H.264, este opera sobre el proceso de decodificación del flujo de video, específicamente, en la rutina VLC (definir que es VLC) inverso el cual permite reconocer la sintaxis de los diferentes elementos (GOP, Cuadro, Rebanadas, MB, etc.) en el flujo codificado entre los cuales se encuentran los vectores de movimiento (VM), además del número de cuadro decodificado. Los VM constituyen la información de entrada para el módulo *Filtro VM*, el cual realiza un proceso de filtrado basado en la magnitud de los VM (ver capítulo). Se espe-

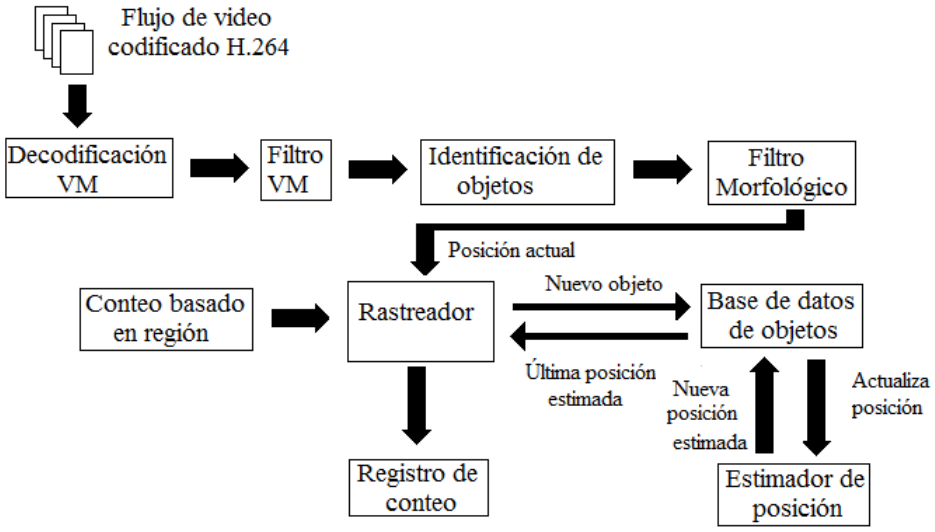


Figura A.1: Arquitectura del sistema propuesto para el conteo de objetos

ra que la salida de este módulo sea un conjunto de VM que correspondan a los objetos en movimiento eliminado VM generados por los fenómenos existentes.

Identificación de objetos

A partir del conjunto de VM filtrados, es posible aplicar cualquiera de las técnicas para la identificación de los objetos en movimiento. Dado que se ha enfatizado en esto a lo largo del presente trabajo, solo se mencionara que la salida de este módulo representan candidatos que posteriormente pueden ser considerados objetos:

- Centroide del objeto
- Total de VM asociados
- Número de cuadro

Filtro morfológico

Este módulo recibe el nombre debido al hecho de realizar un proceso de filtrado sobre el conjunto de objetos identificados en el cuadro actual decodificado. Esto se debe a dos factores principales: 1) el proceso de filtrado de VM considera solo aquellos cuya magnitud sea mayor a un umbral, sin embargo, es posible encontrar VM que cumplan este criterio sin corresponder a objetos en movimiento; 2) las técnicas para la identificación de objetos pueden tomar estos vectores para el agrupamiento, por lo que el módulo de *Identificación de objetos* producirá errores. El módulo de *Filtro morfológico* realiza un proceso de eliminación tomando como criterio el número de VM (definido como *densidad de las agrupaciones*) para los objetos identificados. Esto módulo permite incrementar la fiabilidad en la identificación de los objetos.

Seguimiento

Después de la identificación de los objetos en el cuadro actual decodificado es necesaria una comparación con aquellos encontrados en cuadros anteriores a fin de reconocer la trayectoria que estos han descrito. Esta tarea es realizada en el módulo *Rastreador*, el cual opera de manera conjunta con el modulo *Base de datos de objetos*. Este último almacena la información relacionado a los objetos encontrados en cuadros anteriores, por lo que se realiza una comparación entre los elementos de los módulos *Rastreador-Base de datos de objetos* a fin de poder aplicar las técnicas de seguimiento descritas en este trabajo.

Análisis de complejidad del agrupamiento basado en bloques

En esta sección mostramos un análisis de la complejidad del método propuesto para la identificación de objetos, para esto mostramos el proceso de agrupamiento para un conjunto de vectores que corresponden al mismo objeto, sin perdida de generalidad se puede extender la idea a más de un objeto, se asume que un proceso de filtrado ha sido realizado anticipadamente. El método propuesto emplea una representación de índices que corresponden a bloques de video como se describe a continuación.

Sea vm_0 el primer vector de movimiento decodificado para el cuadro k de la secuencia de video. Es posible definir una matriz de indices de la siguiente manera:

$$vm_0 = \begin{bmatrix} vmr_{0ij} & vml_{0ij} \\ vmr_{0(i+1)(j+1)} & vmr_{0(i+1)(j+1)} \\ vmr_{0(i+2)(j+2)} & vmr_{0(i+2)(j+2)} \\ \cdots & \cdots \\ vmr_{0(i+n)(j+n)} & vmr_{0(i+n)(j+n)} \end{bmatrix} \tag{A.1}$$

Donde vml_0ij es el índice del macrobloque de frontera en el extremo izquierdo en el $i-esimo$ renglón del cuadro de video, de manera análoga, vmr_0ij es el índice del macrobloque de frontera en el extremo derecho en el $i-esimo$ renglón. Gracias a esta definición, $vmr_{0(i+n)(j+n)}$ y $vmr_{0(i+n)(j+n)}$ corresponden a los indices de los macrobloques de frontera $n-renglones$ hacia abajo.

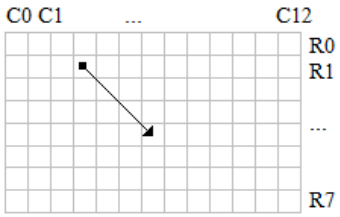


Figura A.2: Construcción de vm_0

Considerando el cuadro de la figura A.2, donde $R0$ representa el primer renglón que forma el cuadro de video, al igual que $C0$ la columna de este. Tenemos que vm_0 se define como:

$$vm_0 = \begin{bmatrix} 16, 16 \\ 28, 28 \\ 42, 42 \\ 55, 55 \end{bmatrix} \tag{A.2}$$

Es importante remarcar que los indices de esta matriz corresponde a los macrobloques dentro del cuadro, por lo tanto, el elemento $vm_0[0, 0] = 16$ corresponde al macrobloque 16 de un total 104 bloques(13x8). Computacionalmente, es posible generar esta matriz conociendo la pendiente m que forman los puntos (origen-destino) del vector de movimiento, este procedimiento se muestra a continuación:

Algoritmo 1 Generacion del vector mv_0

Entrada: mb_{origen} $mb_{destino}$ $video_w$
Salida: vm_0
1: $x_1 \leftarrow (mb_{origen}$ MOD $video_w)$
2: $x_2 \leftarrow (mb_{destino}$ MOD $video_w)$
3: $y_1 \leftarrow (mb_{origen}$ DIV $video_w)$
4: $y_2 \leftarrow (mb_{destino}$ DIV $video_w)$
5: $\frac{m \leftarrow (y_2 - y_1)}{(x_2 - x_1)}$
6: $c \leftarrow (y_2 - (mx_2))$
7: Crea Matriz $vm_0 = [y_2 - y_1][2]$
8: $j \leftarrow 0$
9: **para** $i = x_1$ hasta x_2 **hacer**
10: mv[0][j] $\leftarrow im + c$
11: mv[1][j] $\leftarrow$ mv[0][j]
12: **fin para**

Podemos observar que el proceso para la generación de la matriz vm_0 depende de la magnitud que tenga el vector de movimiento, sin embargo, resulta casi prácticamente imposible contar con vectores de movimiento cuya magnitud sea próxima al tamaño del video.

Bibliografía

[1] L. Richardson. H.264 and MPEG-4 Video Compression Video Coding for Next-generation Multimedia. Great Britain, John Wiley & Sons Ltd, 2003. 306p.

[2] J. Bernd, H. Horst. Computer vision and applications: a guide for students and practitioners. USA, Academic Press, 2000. 679p.

[3] P. Symes. Digital Video Compresison. USA, McGraw-Hill, 2004. 394p.

[4] W. You, M.S. Hoauri, M. Kim. Moving Object Tracking in H.264/AVC Bitstream. Proceedings of the 2007 International Conference on Multimedia Content Analysis and Mining, pages 483-492 , 2007.

[5] J. S. Somolinos. Avances de robotica y visión por computador. Madrid, Cuenca, 2002. 285p.

[6] M. Bhuyan, B. Lovell, A. Bigdeli. Tracking with Multiple Cameras for Video Surveillance. In Digital Image Computing Techniques and Applications. 9th Biennial Conference of the Australian Pattern Recognition Society, Glenelg, Australia, pages 592-599, December 2007.

[7] W. Zeng, J. Du, W. Gao, Q. Huang. Robust Moving Object Segmentation on H.264/AVC Compressed Video Using the Block-Based MRF model. Journal of Visual and Image Representation. Vol. 20. 428-437. 2005.

[8] Wiegand T, Sullivan GJ, Bjntegaard G, Luthra A. Overview of the H.264/AVC Video Coding Standard. IEEE Transaction on Circuits Systems for Video Technology. 2003.

[9] K. Hariharakrishnan, D. Schonfeld. Fast Object Tracking Using Adaptive Block Matching. IEEE Transactions on Multimedia. 853-859. 2005.

[10] R. Achanta, M. Kankanhalli, and P. Mulhem. Compressed Domain Object Tracking for Automatic Indexing of Objects in MPEG Home Video. In IEEE International Conference on Multimedia and Expo, volume 2, pages 61-64, 2002.

[11] F. Bartolini, V. Capellini, C. Giani. Motion Estimation and Tracking for Urban Traffic Monitoring. Proc. Of ICIP, Vol. 3, pp. 787-790, 1996

[12] T. Yokoyama, T. Iwasaki, T. Watanabe. Motion Vector Based Moving Object Detection and Tracking in the MPEG Compressed Domain. Proc. Seventh International Workshop on Content-Based Multimedia Indexing. pages 201-206. 2009.

[13] A. M. Tekalp, P. Van Beek, C. Toklu, B. Gunsel. Two-Dimensional Mesh-Based Visual Object Representation for Interactive Synthetic. Natural Digital Video. Proc. IEEE, Vol. 86, no. 5, pp. 1029-1051, Jun. 1998.

[14] C. Pope, S. Bruyne, T. Paridaens, R. Walle. Moving Object Detection in the H.264/AVC Compressed Domain for Video Surveillance Applications. Journal of Visual Communication and Image Representation.pp. 428-437, August 2009.

[15] H. Eng, K. Ma. Motion Trajectory Extraction Based on Macroblock Motion Vectors for Video Indexing. In International Conference on Image Processing, volume 3, pages 284-288, 1999.

[16] M. Ritch, N. Canagarajah. Motion-Based Video Object Tracking in the Compressed Domain.Image Processing, 2007. ICIP 2007. IEEE International Conference on IEEE International Conference. Oct 2007

[17] V. Bhaskaran, k. Konstantinides. Image and Video Compression Standards: Algorithms and Architectures. Kluwer Academic Publisher Ed. Six Edition. 2003.

[18] Z. Qiya, L. Zhicheng. Moving Object Detection Algorithm for H.264/AVC Compressed Video Stream. Proc. International Colloquium on Computing, Communication, Control, and Management. August, 2009.

[19] A. Bobick, J. Davis. The recognition of Human Movement Using Temporal Templates. IEEE Transactions on Pattern Analysis and Machine Intelligence. Vol 23, No 3, March 2001.

[20] C. Needham, R. Boyle. Performance Evaluation Metrics and Statistics for Positional Tracker Evaluation. 3rd International conference on computer vision systems. pp. 278-289, 2003.

[21] Eng H., Ma KK. Spatiotemporal segmentation of moving video objects over MEPG compressed domain. Proceedings of the IEEE International Conference on Multimedia and Expo 2000.

[22] K. Fukunaga, L.D. Hostetler. The estimation of the gradient of a density function, with applications in pattern recognition. IEEE Trans. Information Theory, vol. 21, pp. 32-40, 1975.

[23] Y. Cheng. Mean Shift, Mode Seeking, and Clustering. IEEE Trans. Pattern Analysis And Machine Intelligence. Vol. 17, No. 8, August 1995.

[24] D. Comaniciu. Mean Shift: A Robust Approach Toward Feature Space Analysis. IEEE Trans. Patern On Analysis And Machine Intelligence, Vol 24, No. 5, May 2002.

[25] G. Tian, R. Hu, Z. Wang, L. Zhu. Object Tracking Algorithm Based on Meanshift Algorithm Combining with Motion Vector Analysis. First International Workshop on Education Technology and Computer Science, vol. 1, 2009.

[26] R. Xu and D.Wunsch. Survey of clustering algorithms. IEEE Transactions on Neural Networks, 2005.

[27] D. Gatica-Perez, C. Gu, M. T. Sun. Semantic video object extraction using four-band watershed and partition lattice operators. IEEE Trans. Circuits Syst. Video Technol., vol. 11, pp. 603–618, May 2001.

[28] J. Miano. Compressed image file formats: JPEG, PNG, GIF, XBM, BMP. USA, Addison Wesley Longman, 1999, 264p.

[29] D. Salomon. Data Compression The Complete Reference. USA, Springer, 2007. 1092p.

[30] D. Comaniciu, P. Meer. Mean Shift Analysis and Applications. The Proceedings of the Seventh IEEE International Conference on Computer Vision, Vol. 2, 1999.

[31] G. Bradski. Real Time Face and Object Tracking as a Component of a Perceptual User Interface. IEEE Workshop Applications of Computer Vision. Princeton, 1998.

[32] K. Alsabti, S. Ranka, V. Singh. An Efficient k-means Clustering Algorithm. Proc. First Workshop High Performance Data Mining, Mar. 1998.

[33] J. Bezdek. Pattern Recognition with Fuzzy Objective Function Algorithms. Plenum Press, New York. 1981.

[34] A. Baraldi, P. Blonda. A survey of fuzzy clustering algorithms for pattern recognition – Part I and II. IEEE Trans. Systems Man Cybernet. Pt. B 29, 1999.