

BENEMÉRITA UNIVERSIDAD AUTÓNOMA DE PUEBLA

Facultad de Ciencias de la Computación

Implantación de Métricas para la Evaluación de Algoritmos de Seguimiento de Objetos en Movimiento.

por

Gloria Hernández Vázquez

para obtener al grado de:

Licenciado en Ciencias de la Computación

PUEBLA PUEBLA
Junio de 2008

Resumen

Los algoritmos para seguimiento de objetos son de gran interés en los sistemas de vigilancia, ya que de ellos depende su buen funcionamiento.

Existe una gran variedad de algoritmos para seguimiento de objetos en movimiento, algunos de los problemas a los que se enfrenta cada uno de ellos son:

- **La frecuencia y complejidad de oclusiones dinámicas:** El proceso de oclusión se presenta cuando un objeto en movimiento se oculta detrás de otro objeto en movimiento y el sistema pierde al objeto que está rastreando.
- **Las características de cada objeto:** En este caso el seguimiento se centra en alguna característica del objeto, por ejemplo el color, tamaño, forma, etc.
- **Los cambios de iluminación o las condiciones climáticas:** Debido a éstos se pueden obtener distorsiones considerables en los objetos de las imágenes.

Debido a que cada sistema de vigilancia tiene diferentes necesidades, se vuelve compleja la elección del algoritmo adecuado. Por ello es necesario evaluar el desempeño de los algoritmos frente a un mismo problema y con el mismo conjunto de datos. Así, con los resultados obtenidos, se podrá elegir el algoritmo que se adapte más al sistema. Existen diversas métricas que facilitan dicha evaluación, éstas son capaces de evaluar los diferentes problemas a los cuales se enfrenta el seguimiento de objetos.

Las métricas pueden evaluar a los algoritmos de manera cualitativa (atributos) y cuantitativa (estadísticamente), en esta tesis se usarán las mediciones cuantitativas estipuladas en el proyecto ETISEO [15], ya que el laboratorio de visión requiere evaluar sus algoritmos para posteriormente participar en este proyecto.

El proyecto ETISEO requiere que los participantes utilicen un formato específico de datos, debido a esto, se diseñó un sistema que cuenta con 3 módulos: Algoritmos de seguimiento, Generar XML y Evaluación. Dentro del primer módulo se podrá seleccionar el algoritmo de seguimiento que se desee evaluar, en el segundo módulo se transforman los resultados obtenidos por el algoritmo al formato requerido (XML) y por último en el tercer módulo se aplican las métricas y se obtienen resultados. Cada uno de estos módulos se explica con detalle en el capítulo 3.

Se propone una metodología que evalúa de manera automática el resultado de los algoritmos, así como el análisis de cada métrica, para establecer las ventajas y desventajas de cada una de ellas.

Agradecimientos

Quiero expresar mi agradecimiento:

A Jehová por darme la vida y la fuerza necesaria para seguir adelante día con día.

A mi madre y a mis hermanos por apoyarme siempre en mis caminos.

Al Dr. Leopoldo Altamirano Robles, por confiar en mi al brindarme la oportunidad de participar en el laboratorio de visión por computadora del Instituto Nacional de Astrofísica Óptica y Electrónica.

A todos los doctores y maestros que colaboran en el laboratorio de visión y que se toman algunos minutos de su tiempo para ayudar a estudiantes inexpertos como yo. En especial al M. C. Iván Olivera que en muchas ocasiones compartió su conocimiento conmigo, a la Dra. Janeth Cruz que siempre tuvo la paciencia para resolver mis dudas y brindarme valiosos consejos, así como al M. C. Héctor Barrón que en distintas ocasiones me brindó su apoyo.

A todos mis maestros de la Benemérita Universidad Autónoma de Puebla que me apoyaron en mi formación académica, en especial al MC. José Alfonso Garcés Baéz por apoyarme siempre y brindarme su confianza.

A todos mis amigos y amigas que estuvieron siempre junto a mi para apoyarme, especialmente a mi compañera de carrera y amiga Adriana Cervantes Olvera, por haberme aconsejado acerca de este proyecto y nunca dejarme sola. GRACIAS.

Dedicatoria

A mi mamá
que se ha esforzado
toda su vida
porque mis hermanos
y yo tengamos
un buen futuro.

Sé que has
sacrificado muchas
cosas por sacarnos
adelante, pero quiero
que sepas que
el esfuerzo ha
valido la pena.
Te amo.

Índice

1. Introducción.	1
1.1. Antecedentes	3
1.1.1. Sistemas de Vigilancia.	3
1.1.2. Evolución de la Evaluación del Desempeño de Seguimiento de Objetos y Vigilancia. (PETS)	4
1.2. Motivación.	5
1.3. Problemática.	6
1.4. Objetivos	6
1.5. Organización.	6
2. Métricas para Evaluación del Desempeño de Algoritmos de Seguimiento de Objetos.	8
2.1. Seguimiento de Objetos.	8
2.1.1. Seguimiento de un blanco.	9
2.1.2. Seguimiento de múltiples blancos.	10
2.2. Características del Desempeño de Algoritmos para Seguimiento de Objetos.	12
2.3. Métricas	12
2.3.1. Detección imagen a imagen - Ligado entre dos imágenes.	15
2.3.2. Número de objetos detectados en un instante de tiempo.	16
2.3.3. Evaluación en tiempo de seguimiento.	16
2.3.4. Fragmentación ID.	17
2.3.5. Criterio de confusión para IDs.	17
2.3.6. Trayectorias 2D	18
2.4. Resultados de las métricas.	24
2.5. Métricas usadas comúnmente.	25
2.6. Trabajos Relacionados.	26

2.6.1.	Un Método de Evaluación del Desempeño para Seguimiento de Objetos en Video.	26
2.6.2.	Métricas y Estadísticas de Evaluación del Desempeño para Seguimiento Posicional.	26
2.6.3.	Métricas de Evaluación del Desempeño para Detección de Movimiento y Seguimiento de Objetos.	27
2.6.4.	Observación Parcial contra Seguimiento Ciego a través de Oclusiones.	27
3.	Herramientas para Análisis de Algoritmos de Seguimiento de Objetos.	29
3.1.	Modulo 1. Algoritmos de seguimiento.	29
3.2.	Módulo 2. Generar XML.	32
3.3.	Modulo 3. Evaluación.	35
4.	Resultados.	40
5.	Conclusiones.	72
5.1.	Limitaciones y Trabajos Futuros.	73
6.	Apéndices	74
6.1.	Apéndice A: Manual de usuario.	74
6.2.	Apéndice B: Manual Técnico.	76
7.	Referencias	88

Índice de figuras

1.	Elementos de un sistema de seguimiento de un solo blanco(STT).	9
2.	Elementos de un sistema de seguimiento de múltiples blancos (MTT).	10
3.	Área de interés.	14
4.	Intervalos de tiempo de un dato de referencia comparado con un candidato.	14
5.	Par de trayectorias.	18
6.	Comparación de trayectorias.	19
7.	Ejemplo de trayectorias separadas espacialmente.	21
8.	Ejemplo de trayectorias temporalmente separadas.	21
9.	Área entre trayectorias.	22
10.	Evaluación automática de algoritmos de seguimiento.	29
11.	Coordenadas de objeto.	31
12.	Formato de datos XML.	32
13.	Casos de traslape entre objetos para distancia D2.	35
14.	Objeto 0 de secuencia de imágenes número 1.	40
15.	Trayectoria de referencia de objeto 0.	40
16.	Objeto 1 de secuencia de imágenes número 1.	41
17.	Trayectoria de referencia de objeto 1.	41
18.	Objeto 4 de secuencia de imágenes número 1.	42
19.	Trayectoria de referencia de objeto 4.	42
20.	Objeto 5 de secuencia de imágenes número 1.	43
21.	Trayectoria de referencia de objeto 5.	43
22.	Objeto 0 de secuencia de imágenes número 2.	44
23.	Trayectoria de referencia de objeto 0.	44
24.	Objeto 1 de secuencia de imágenes número 2.	45
25.	Trayectoria de referencia de objeto 1.	45

26.	Objeto 2 de secuencia de imágenes número 2.	46
27.	Trayectoria de referencia de objeto 2.	46
28.	Objeto 0 de la secuencia de imágenes número 3.	47
29.	Trayectoria de referencia de objeto 0.	47
30.	Objeto 1 de secuencia de imágenes número 3.	48
31.	Trayectoria de referencia del objeto 1.	48
32.	Objeto 2 de la secuencia de imágenes número 3.	49
33.	Trayectoria de referencia del objeto 2.	49
34.	Objeto 3 de la secuencia de imágenes número 3.	50
35.	Trayectoria de referencia del objeto 3.	50
36.	Secuencia 1 comparada con resultados de métrica 2.3.2.	52
37.	Secuencia 1 comparada con resultados de métrica 2.3.2.	53
38.	Secuencia 2 comparada con resultados de métrica 2.3.2.	54
39.	Secuencia 2 comparada con resultados de métrica 2.3.2.	55
40.	Secuencia 2 comparada con resultados de métrica 2.3.2.	56
41.	Secuencia 3 comparada con resultados de la métrica 2.3.2.	57
42.	Secuencia 3 comparada con resultados de métrica 2.3.2.	58
43.	Secuencia 3 comparada con resultados de la métrica 2.3.2.	59
44.	Resultados de algoritmo 1 comparados con secuencia 1.	64
45.	Resultados de algoritmo 3 comparados con secuencia 1.	65
46.	Resultados de algoritmo 1 comparados con secuencia 2.	66
47.	Resultados de algoritmo 2 comparados con secuencia 2.	67
48.	Resultados de algoritmo 3 comparados con secuencia 2.	68
49.	Resultados de algoritmo 1 comparados con secuencia 3.	69
50.	Resultados de algoritmo 2 comparados con secuencia 3.	70
51.	Resultados de algoritmo 3 comparados con secuencia 3.	71
52.	Proceso general para ligado entre imágenes.	81

1. Introducción.

El objetivo de esta tesis es aplicar las métricas cuantitativas de seguimiento de objetos en movimiento del proyecto ETISEO para determinar cómo se comporta cada métrica, es decir, observar qué cualidades específicas se evalúan del algoritmo con cada una. Facilitando así la elección de métricas para evaluar el desempeño de un algoritmo de acuerdo a sus características.

ETISEO (Evaluation du Traitement et de Interprétation de Séquences Vidéo) es un proyecto de investigación patrocinado por el gobierno francés, con el objetivo de evaluar técnicas de visión para aplicaciones de video vigilancia. Trata de crear herramientas automáticas de evaluación de algoritmos, que permitan la comparación cuantitativa entre los resultados del algoritmo y los datos de referencia. Este proyecto tiene sus propias métricas de evaluación para algoritmos de seguimiento de objetos en movimiento [19].

Es importante conocer cuáles son las mediciones que toma en cuenta cada métrica y qué ventajas o desventajas tienen una sobre la otra. De esta manera se podrá realizar la evaluación del desempeño de algoritmos de seguimiento de objetos en movimiento de acuerdo a los resultados que éste proporciona y las características que se desea conocer de él, facilitando la elección de los algoritmos de acuerdo a las necesidades del sistema.

Según [15], los criterios y métricas se clasifican considerando las siguientes tareas:

1. Detección de objetos físicos de interés.
2. Localización de objetos físicos de interés.
3. Seguimiento de objetos físicos de interés.
4. Clasificación de objetos físicos de interés.
5. Reconocimiento de eventos.

Todo sistema de seguimiento puede ser evaluado en cada una de éstas etapas. El desarrollo de esta tesis se centrará en las métricas para el **seguimiento de objetos físicos de interés**, debido a que es el criterio que se desea evaluar.

Esta etapa de los sistemas de seguimiento cuenta con las 6 siguientes métricas que se programarán en esta tesis:

1. Detección imagen a imagen - Ligado entre dos imágenes.
2. Número de objetos detectados en un instante de tiempo.
3. Evaluación en tiempo de seguimiento.
4. Fragmentación ID.

5. Criterio de confusión para IDs.

6. Trayectorias 2D.

1.1. Antecedentes

Los sistemas de vigilancia han tomado una gran importancia a lo largo de la historia, debido a su gran evolución y confiabilidad. Una rama importante dentro de los sistemas de vigilancia es la visión por computadora, debido a que ésta hace posible mantener la vigilancia sin un operador humano, y aplicando los algoritmos necesarios es posible que el seguimiento de objetos se realice de manera automática. La visión por computadora está sustentada en algoritmos de análisis y procesamiento de imágenes, tiene como objetivo proporcionar una arquitectura artificial con percepción visual, debido a que esta característica permite a los seres vivos obtener una gran cantidad de información de forma eficiente a partir de su entorno.

Un sistema de visión por computadora está conformado por los siguientes elementos: dispositivos para capturar imágenes (sensores), sistemas de iluminación que aseguren la calidad en la imagen, computadoras y software que pueda hacer la manipulación de las imágenes adquiridas, de tal forma que se pueda hacer el reconocimiento de características en la imagen.

El seguimiento de objetos permite enfocar los recursos computacionales para el análisis de objetos de interés dentro de entornos dinámicos, por lo que es ampliamente utilizado en diversas aplicaciones. H. Barrón explica en [11] que los problemas que fundamentalmente envuelven a los sistemas autónomos que tienen que interactuar con ambientes reales no estructurados, están relacionados con el conocimiento incompleto del entorno y el decremento de la probabilidad de detección debido a ambientes dinámicos. Además, las características de los sensores seleccionados limitan el desempeño de los sistemas, debido a que existen sensores ineficientes, ruidosos, con fallas y con un campo de visión limitado. Lo cual significa que las decisiones no se basan en información confiable al cien por ciento. Por lo tanto el desarrollo de sistemas de visión es una tarea importante para crear aplicaciones robustas.

Un concepto muy importante dentro del área de la visión por computadora es el procesamiento digital. El procesamiento digital de imágenes se define como el proceso mediante el cual se toma una imagen y se produce una versión modificada de ella. Para las aplicaciones que involucran seguimiento de objetos en movimiento es de especial interés realizar un procesamiento digital de tal forma que se puedan aislar los objetos que experimentan cambios en su posición con respecto al fondo de la escena de aquellos elementos que no sufrieron desplazamiento dentro del área de visión de la cámara. Este proceso digital es conocido como segmentación de objetos en movimiento, que trata de identificar cuáles son los píxeles que pertenecen a un objeto en una imagen. Normalmente el resultado de este proceso es una imagen binaria constituida por píxeles blancos o negros que representan al objeto. Este proceso es comúnmente usado para la elaboración de sistemas de vigilancia por computadora.

1.1.1. Sistemas de Vigilancia.

Un sistema de vigilancia por computadora se caracteriza por 5 elementos necesarios:

1. Adquisición de la imagen.

2. Segmentación del objeto.
3. Seguimiento del objeto.
4. Clasificación del objeto.
5. Descripción de la escena.

La **adquisición de la imagen** puede hacerse mediante un sensor, que es un dispositivo que permite obtener información del medio ambiente, como una cámara.

La **segmentación de un objeto** ayuda a identificar y separar los objetos que son de interés de los que no lo son.

El **seguimiento de objetos** juega un papel muy importante dentro de cualquier sistema de vigilancia visual, y pueden presentarse diferentes algoritmos para diversas aplicaciones. Esta tesis se centrará en este punto vital durante su desarrollo, se tratará el tema con mayor detenimiento en el siguiente capítulo.

1.1.2. Evolución de la Evaluación del Desempeño de Seguimiento de Objetos y Vigilancia. (PETS)

En la actualidad existen grupos de trabajo denominados PETS (Performance Evaluation of Tracking and Surveillance), que en [5] se definen como un conjunto de grupos de trabajo enfocados al estudio de las diferentes técnicas y metodologías para la evaluación del desempeño de los algoritmos utilizados por sistemas de vigilancia, éstos contribuyen a un desarrollo de la visión por computadora en el campo de la vigilancia. Consisten en un número de presentaciones de los principales investigadores que trabajan en el área de la vigilancia visual, y una discusión de panel. Estos grupos de trabajo son los únicos en donde todos los participantes prueban sus algoritmos en los mismos conjuntos de datos.

En estos grupos de trabajo se analizan los algoritmos para y se definen pautas que cada algoritmo debe cumplir para ser óptimo en su desempeño. Los principales objetivos de los grupos de trabajo en PETS son:

- Reunir a investigadores interesados en el área de la vigilancia visual.
- Aplicar diversos algoritmos al mismo conjunto de datos.
- Evaluar las diferencias entre los diversos modelos y algoritmos.
- Discutir qué criterios se deben utilizar para la evaluación objetiva de dichos algoritmos.
- Discutir cómo documentar el funcionamiento (incluyendo exactitud) de los algoritmos de vigilancia.
- Discutir el desarrollo de una metodología para probar los algoritmos.

- Discutir el desarrollo en curso de pautas para realizar pruebas comparativas empíricas.

Los grupos de trabajo en PETS se han llevado a cabo año tras año desde el 2000. Es difícil realizar comparaciones estándar entre algoritmos si son probados con diferentes conjuntos de datos, por ello, en el 2000 se creó un conjunto de datos estándar para que los investigadores apliquen sus algoritmos sobre ellos. El siguiente grupo de trabajo se realizó en el año 2001, en el que se continuó con el tema del año anterior, debido a su gran impacto. Para el año 2002 se cambió el tema por "seguimiento de personas y reconocimiento de gestos". Las áreas de aplicación incluyeron: clasificación estática de la postura de la mano, seguimiento de personas y conteo. Posteriormente, los PETS 2003 se basaron en modelado de escenas, aprendizaje, modelado del cuerpo humano, estimación de actitud, clasificación y seguimiento de objetos. El tema para PETS 2004 fue observando la actividad humana. Se analizaron diferentes videos, en los que se incluyeron personas caminando solas, conviviendo con otras, peleando y dejando un paquete en un lugar público. Lo cual implica el empleo de algoritmos con mayor robustez a las oclusiones entre objetos y particiones de objetos. El enfoque para el 2005 fue detección y seguimiento de un objeto.

Muchos de estos grupos de trabajo han dejado un antecedente muy fuerte en cuanto a la evaluación del desempeño de algoritmos. Xu y Ellis en [9], dividen la evaluación del desempeño de algoritmos en:

- **Desempeño cualitativo:** Permite evaluar al algoritmo de acuerdo a ciertas cualidades propias del algoritmo (tiempo de ejecución, robustez, etc).
- **Desempeño cuantitativo:** Permite cuantificar específicamente algunas características de los algoritmos (nivel de exactitud, errores, etc).

Este proyecto se enfocará en el desempeño cuantitativo como base para la selección de las métricas a aplicar, ya que se desea medir la exactitud de los algoritmos.

1.2. Motivación.

El seguimiento de alta exactitud permite el incremento de la confiabilidad en sistemas de seguridad y vigilancia que incluyen detección de movimiento inteligente, análisis de eventos y seguridad doméstica, por lo que es vital que las mediciones sean exactas y confiables para una correcta toma de decisiones. Ya que los sistemas de vigilancia dependen de los algoritmos para seguimiento de objetos en movimiento, es necesario evaluar el desempeño de los mismos, y para ello es útil saber con qué métricas se cuenta y cuales son las ventajas que ofrece una sobre la otra, dependiendo de las necesidades del sistema.

Dentro de la Coordinación de Ciencias Computacionales, en el Laboratorio de Visión por Computadora del Instituto Nacional de Astrofísica Óptica y Electrónica (INAOE), se requiere de un

sistema que realice la evaluación del desempeño de los algoritmos usados para el seguimiento de objetos de manera automática. Es por ello que se hace necesaria la programación de determinadas métricas para el seguimiento de objetos, así como un análisis de las mismas.

1.3. Problemática.

Existen métricas enfocadas a la evaluación del desempeño de algoritmos de seguimiento de objetos en movimiento, que facilitan el proceso de comparación.

Los algoritmos de seguimiento de objetos en movimiento entregan como resultado una trayectoria, que es generada mediante el seguimiento efectuado por el algoritmo sobre un determinado objeto. De esta forma si se desea evaluar el desempeño de dichos algoritmos es necesario contar con una "trayectoria de referencia", la cual representa el camino que recorrió el objeto realmente. Así, se puede realizar una comparación directa entre ambas trayectorias (la resultante del algoritmo y la de referencia) y saber que tan bueno es el desempeño del algoritmo.

Para probar la eficacia de las métricas en cuanto a evaluación, se proveerán los mismos conjuntos de datos para cada algoritmo, se implantarán las métricas a cada uno de los algoritmos de seguimiento de objetos en movimiento, y se medirá su desempeño. Además se realizará un análisis de los resultados arrojados por métrica y por último se analizarán las ventajas y desventajas de las mismas.

1.4. Objetivos

Objetivo General:

Programar y evaluar las métricas para la evaluación del desempeño de algoritmos de seguimiento de objetos en movimiento propuestas por ETISEO.

Objetivos Específicos:

1. Aplicar métricas estándar para la evaluación del desempeño en algoritmos de seguimiento de objetos.
2. Realizar análisis y comparación de resultados obtenidos por las métricas.
3. Analizar las ventajas y desventajas de cada métrica.

1.5. Organización.

El resto de la tesis se encuentra organizada en los siguientes capítulos:

-
- En el capítulo 2 se presenta el estado del arte del seguimiento de objetos, así como los conceptos relacionados con el tema. Además se explica con detalle cuáles son los elementos necesarios para la evaluación de los algoritmos. También se dan a conocer las métricas a programar y los avances más significativos en el área.
 - En el capítulo 3 se explica con detenimiento cada uno de los módulos que componen al sistema, así como su elaboración.
 - En el capítulo 4 presentan los resultados obtenidos y se realiza un análisis de los mismos.
 - El capítulo 5 se dan a conocer las conclusiones del trabajo elaborado y se proponen alternativas para la extensión del mismo.

2. Métricas para Evaluación del Desempeño de Algoritmos de Seguimiento de Objetos.

2.1. Seguimiento de Objetos.

El seguimiento de objetos es un elemento esencial para los sistemas de vigilancia. Permite generar inferencias acerca del movimiento de un objeto mediante la estimación de su estado. Esto es posible obteniendo una secuencia de imágenes a través de uno o varios sensores sobre plataformas que pueden ser fijas o presentar movimiento.

En el seguimiento basado en imágenes, es necesario monitorear los cambios espaciales y temporales que sufre el objeto en la secuencia de video; por lo que, este proceso se realiza al resolver el problema de correspondencia temporal entre dos imágenes del mismo objeto, las cuales representan el estado del objeto en distintos instantes de tiempo. El seguimiento basado en imágenes es más amplio en alcance que la estimación ya que no solo hace uso de herramientas de estimación sino además de teorías de decisión estadísticas.

El seguimiento es una tecnología que tiene diversas aplicaciones. En [14] se definen tres tópicos dominantes dentro de la misma:

Seguimiento Vehicular Sistemas capaces de reportar congestionamientos de tráfico, accidentes y movimientos peligrosos o ilegales por parte de los conductores.

Vigilancia Sistemas capaces de reportar qué es lo que las personas hacen, usualmente con el objetivo de observar a personas que presenten comportamientos sospechosos.

Interacción Humano-Computadora Sistemas que usan acciones de personas para controlar varios dispositivos.

Un sistema de seguimiento colecciona un conjunto de datos a partir de sensores, los cuales representan atributos de uno o más objetos de interés. A este conjunto de datos recolectados se le conoce como mediciones u observaciones. Se llama blanco al objeto que se está siguiendo a partir de las mediciones. Internamente, en el sistema de seguimiento, un blanco es representado por su estado. El estado de un objeto es el conjunto de parámetros que definen su comportamiento, tal como la posición, velocidad de movimiento, tamaño, color; el estado puede ser actualizado con cada nueva medición. Los sistemas de seguimiento de un solo blanco tratan de mantener en la línea de vista al objeto de interés. La diferencia entre la línea de vista y las mediciones sirve de entrada a un filtro predictivo el cual estima la posición del objeto y permite mantener al objeto de interés en seguimiento.

Básicamente existen dos tipos de sistemas para llevar a cabo el seguimiento de objetos, de acuerdo al número de objetos en seguimiento, aquellos que realizan el seguimiento de un solo objeto y los que hacen el seguimiento de múltiples objetos en el campo de visión del sensor.

2.1.1. Seguimiento de un blanco.

Un sistema que realiza el seguimiento de un solo blanco (STT) lleva a cabo funciones tales como conservar la línea de vista dirigida al objeto de interés, minimizar el error que existe entre el desplazamiento del blanco y el ángulo de orientación del sensor. En sistemas que realizan el seguimiento de múltiples blancos se consideran todos los elementos que están presentes en el campo de visión del sensor. Por lo tanto, esta clase de sistemas no requieren mantener algún objeto en línea de seguimiento, excepto que se quiera designar la atención del sistema a un solo objeto de interés [7].

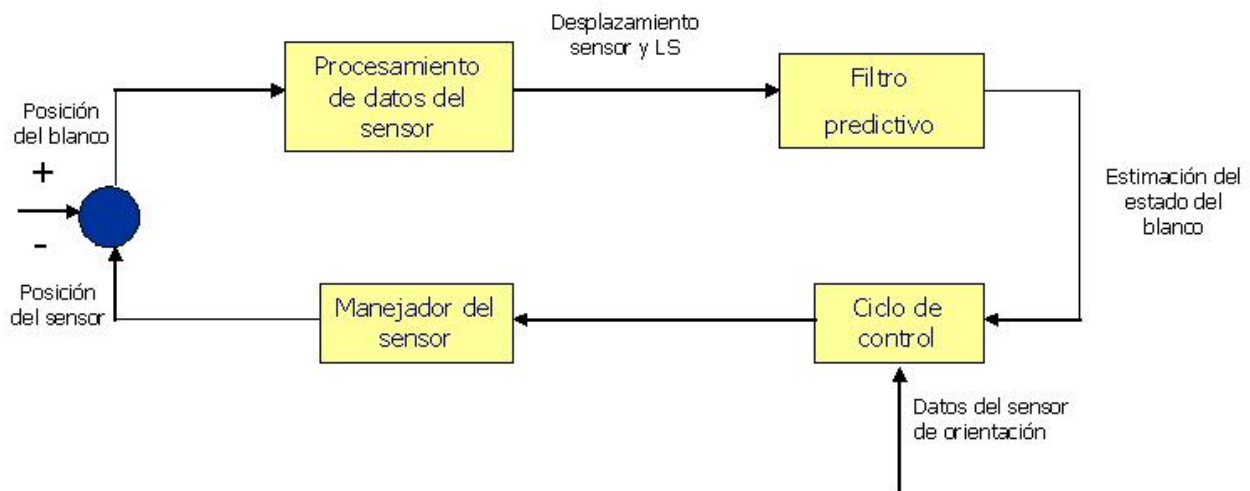


Figura 1: Elementos de un sistema de seguimiento de un solo blanco (STT).

En la figura 1 se puede observar el ciclo del proceso de seguimiento de un blanco, que se lleva a cabo de la siguiente manera: se tiene como entrada la posición del blanco, es decir, las coordenadas que el objeto tiene en la muestra que el sensor obtiene, además se tiene la posición actual del sensor en ese instante de tiempo, el cual puede estar sobre una plataforma que puede realizar movimiento. Estos datos son ingresados al primer bloque del ciclo que corresponde al procesamiento de datos del sensor, de tal forma que se procesa la información obtenida para conocer la posición actual del objeto en seguimiento, información que es combinada con los datos de entrada. El resultado obtenido es el desplazamiento que existe entre el objeto en seguimiento y la línea de vista del sensor. El proceso de filtrado tiene como objetivo estimar la posición del blanco relativa al ángulo de orientación del sensor, la cual es usada como una entrada al ciclo de control que intenta guardar la orientación del sensor con respecto al blanco. El bloque de control es el encargado de controlar el movimiento del sistema donde está colocado el sensor. Este bloque es de gran importancia, ya que de este depende que la línea de vista del sensor realmente esté orientada al objeto en seguimiento. Finalmente, se tiene el bloque que manipula la posición del sensor, donde se lleva a cabo la actualización de la posición del dispositivo de posicionamiento del sensor con respecto a la línea de vista del mismo. Los sistemas que realizan el seguimiento de un solo blanco requieren que la orientación del sensor con respecto al blan-

co sea muy precisa para mantener al objeto en la línea de vista. Frecuentemente se requieren técnicas de estabilización del sensor, ciclos de control, procesamiento de los datos que proporciona el sensor y tecnologías de seguimiento de objetos para conseguir que el blanco de interés permanezca en estado de seguimiento.

2.1.2. Seguimiento de múltiples blancos.

El objetivo de un sistema de seguimiento de múltiples blancos (MTT) es formar y mantener las trayectorias de los objetos de interés a partir de las mediciones que proveen los sensores. Un sistema MTT incrementa su dificultad en el seguimiento de objetos con movimientos estratégicos y en la presencia de ambientes con múltiples objetos en desorden[7].

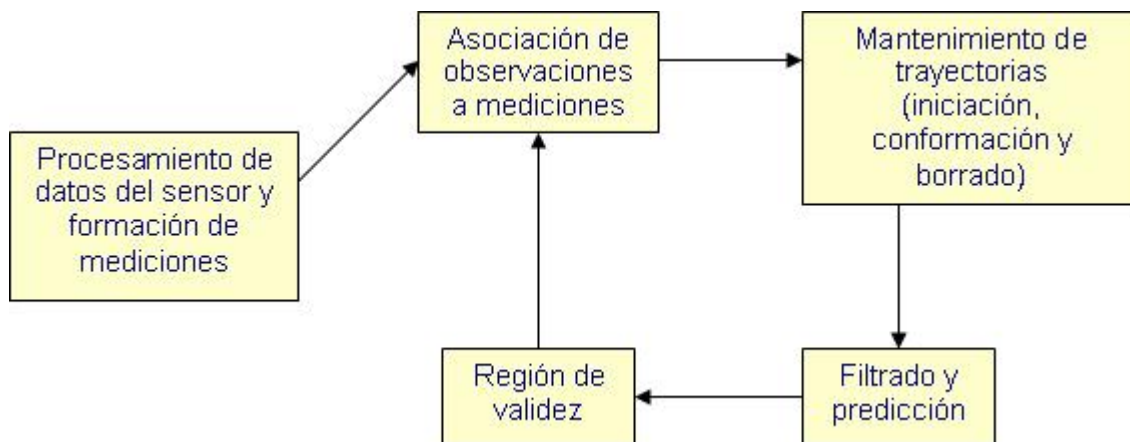


Figura 2: Elementos de un sistema de seguimiento de múltiples blancos (MTT).

Como se observa en la figura 2, el ciclo del proceso de seguimiento de múltiples blancos se lleva a cabo de la siguiente manera: Como entrada se tiene el bloque de procesamiento de datos y formación de mediciones donde se lleva a cabo la captura de datos con el sensor y el procesamiento de la información proporcionada por el sensor. El resultado de este bloque es información cinemática de los objetos de interés, la cual provee, al sistema de seguimiento, información acerca de la existencia y localización del blanco, por ejemplo posición y velocidad, además de otros datos que identifiquen a estos objetos. Los bloques de asociación de observaciones a mediciones, mantenimiento de mediciones y región de validez corresponden a la parte de asociación lógica de datos, la cual tiene como entrada el resultado que se obtiene en el bloque de filtro de predicción. La selección de blancos se utiliza como una técnica que sirve para determinar cuáles de las observaciones son válidas para realizar el seguimiento. Esta función también es ejecutada para reducir el tiempo de procesamiento a los bloques de asociación de observaciones y mantenimiento. La asociación de observaciones a los seguidores es una función que realiza la asignación de parejas obtenidas del proceso de selección, de tal forma que esta asignación es única, es decir, esta función asegura que la observación será asignada a una medición dada y esta asignación

será para adaptar la información de un solo seguidor. La función de mantenimiento tiene como objetivo actualizar la información de los objetos en seguimiento, también para dar de alta objetos nuevos que son candidatos a ser blancos en seguimiento. Finalmente, el paso de filtrado predictivo sirve para hacer la predicción de las posiciones de los objetos en seguimiento a partir de la observación actual. A través de estas predicciones se definen las vecindades de búsqueda para los objetos en seguimiento, donde las coordenadas encontradas por el filtro predictivo son de gran importancia ya, que definen el centro de las vecindades de búsqueda.

2.2. Características del Desempeño de Algoritmos para Seguimiento de Objetos.

El objetivo de la evaluación algorítmica es determinar el desempeño de un algoritmo dados diferentes parámetros y los datos de entrada. Estas métricas cuantifican la ocurrencia, la estabilidad, la robustez y la complejidad del algoritmo.

La ocurrencia es medida en términos de p_e , error de posición, que representa la desviación de la trayectoria resultante del algoritmo respecto a la "trayectoria de referencia". El error de posición es la distancia euclidiana entre el centro de masa del área mínima que contiene al blanco detectado y la trayectoria de referencia.

El tamaño del error, s_e , se expresa en términos de Falsos Positivos y Falsos Negativos:

$$s_e = \frac{(f_p + f_n)}{(A_g + A_t)} \quad (1)$$

Donde f_p y f_n son el número de Falsos Positivos y Falsos Negativos respectivamente; A_g y A_t son el número de pixeles en la "trayectoria de referencia" y la trayectoria del algoritmo respectivamente.

2.3. Métricas

El proyecto ETISEO cuenta con diferentes métricas para la evaluación de algoritmos de seguimiento de objetos. En este capítulo se explicarán con mayor detalle, para tener una mayor comprensión de las mismas, es necesario definir algunos conceptos previamente.

Trayectoria de referencia: Es una trayectoria dada por un operador humano, la cual describe los resultados obtenidos en el mundo real por la secuencia de imágenes con que será evaluado el algoritmo. Esta trayectoria debe ser única y corresponder con los requerimientos del usuario final, esta información puede contener errores y puede ser escrita en un archivo con formato XML o MPEG7 [20].

Una trayectoria de referencia contiene datos que se consideran fidedignos y así poder compararlos con los resultados obtenidos por un algoritmo, es decir, es la trayectoria **real**. Existen diversas formas de generar la trayectoria de referencia, una de las más comunes es **manualmente**.

En [3] las trayectorias de referencia son evaluadas de acuerdo a las siguientes métricas:

- **Coherencia de ruta:** Hace la suposición de que la trayectoria derivada del objeto seguido debe ser un objeto liso en dirección y movimiento. Las medidas son penalizadas por menor consistencia con respecto a dirección y velocidad, mientras que son recompensadas por la situación inversa.

$$\varepsilon_{pc} = \frac{1}{N-2} \sum_{k=2}^{N-1} w_1 \left(1 - \frac{\overline{X_{k-1}X_k} \bullet \overline{X_kX_{k+1}}}{\|\overline{X_{k-1}X_k}\| \|\overline{X_kX_{k+1}}\|} \right) + W 2 \left(1 - \frac{\sqrt{\|\overline{X_{k-1}X_k}\| \|\overline{X_kX_{k+1}}\|}}{\|\overline{X_{k-1}X_k}\| + \|\overline{X_kX_{k+1}}\|} \right) \} \quad (2)$$

Donde $\overline{X_{k-1}X_k}$ es el vector posicional de cambios del objeto seguido entre el imagen k y k-1.

- **Coherencia de color:** Esta métrica mide la distancia promedio en el histograma del objeto seguido. Supone que el histograma del objeto debe permanecer constante entre imágenes. El histograma normalizado es generado usando el espacio de color (r,g). Esta métrica tiene valores pequeños si el objeto segmentado tiene atributos de color similares, y valores más altos cuando los atributos de color son diferentes. Cada histograma contiene 8x8 bins para los componentes de color normalizado.

$$\varepsilon_{cc} = \frac{1}{N-1} \sum_{k=2}^N \sqrt{1 - \sum_{u=1}^M P_{k-1}(u)P_k(u)} \quad (3)$$

Donde $P_k(u)$ es histograma de color normalizado del objeto seguido en el imagen k, el cual tiene M bins, y N es el número de imágenes sobre los cuales el objeto fue seguido. Esta métrica es una medida de color muy popular empleada por varios algoritmos de seguimiento.

- **Coherencia de forma:** Indica el nivel de acuerdo entre la posición del objeto seguido y el primer plano

Área mínima que contiene al objeto: Es la caja mínima formada por coordenadas 2D en la que se encierra a un objeto en una imagen. Es representado por las coordenadas (h,w), el ancho y el alto.

Se definen dos conjuntos principales usados con propósitos de asociación, áreas e intervalos de tiempo[15]:

1. **RD:** intervalo de tiempo o área de interés de un dato de referencia (dato tomado de la trayectoria de referencia). Se puede observar en la figura 3.
2. **C:** (figura 4) Intervalo de tiempo o área de interés de un dato candidato (dato resultante del algoritmo de seguimiento).

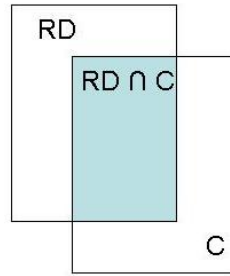


Figura 3: Área de interés.

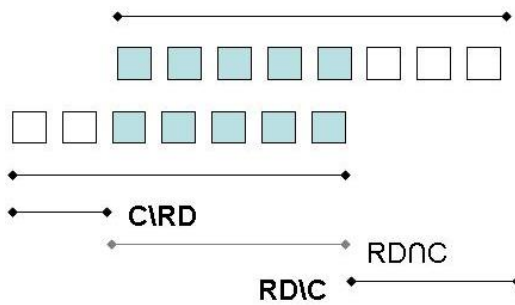


Figura 4: Intervalos de tiempo de un dato de referencia comparado con un candidato.

Para evaluar la asociación entre un dato candidato y un dato de referencia, se usará alguna de las siguientes distancias:

- **D1 - Doble coeficiente:**

$$\frac{2 * card(RD \cap C)}{(card(RD) + card(C))} \quad (4)$$

- **D2 - Traslape :**

$$\frac{card(RD \cap C)}{card(RD)} \quad (5)$$

- **D3 - Distancia de Bertozzi :**

$$\frac{(card(RD \cap C))^2}{card(RD) * card(C)} \quad (6)$$

- **D4 - Desviación máxima :**

$$Max \left\{ \frac{card(C \setminus RD)}{card(C)}, \frac{card(RD \setminus C)}{card(RD)} \right\} \quad (7)$$

Conociendo lo anterior, se definen las métricas que se implantaron en esta tesis [15].

2.3.1. Detección imagen a imagen - Ligado entre dos imágenes.

Esta parte de la evaluación estima si la liga entre dos objetos físicos detectados en dos instantes consecutivos de tiempo es calculada correctamente o no.

Definición Número de asociaciones entre objetos obtenidas en los resultados del algoritmo comparado con el número de asociaciones en la trayectoria de referencia. Esta métrica usa la siguiente información de comparación:

1. El área mínima que contiene al objeto detectado por el algoritmo en un tiempo t y $t+1$, se relaciona con el mismo dato de referencia usando una de las distancias $D1$ a $D4$ y un umbral adecuado.
2. Existe una liga entre los objetos detectados en un tiempo t y $t+1$, y una liga existente en los datos de la trayectoria de referencia.

Usando esta métrica podemos obtener los siguientes resultados:

- Seguimiento Correcto (GT) = Una liga existente en el dato de referencia detectada por el algoritmo.
- Seguimiento Falso (FT) = Una liga entre dos objetos de los resultados del algoritmo no asociada a una liga en la trayectoria de referencia.
- Seguimiento Perdido (MT1) = Rechazo de información de comparación 1.
- Seguimiento Perdido 2 (MT2) = Rechazo de información de comparación 1 y 2.
- $OT = GT + FT$ (todas las ligas detectadas por el algoritmo).
- $RT1 = GT + MT1$.
- $RT2 = GT + MT2$.
- Precisión: número de GT/Número de OT.
- Sensitividad 1: Número de GT/Número de RT1.
- Sensitividad 2: Número de GT/Número de RT2.
- F-contador1: $2 * \text{Precisión} * \text{Sensitividad1} / (\text{Precisión} + \text{Sensitividad1})$.
- F-contador2: $2 * \text{Precisión} * \text{Sensitividad2} / (\text{Precisión} + \text{Sensitividad2})$.

2.3.2. Número de objetos detectados en un instante de tiempo.

Este criterio mide la habilidad global del sistema para detectar y seguir un objeto a través del tiempo.

Definición Número de objetos de la trayectoria de referencia detectados por el algoritmo de acuerdo a sus intervalos de presencia. Definiendo un umbral adecuado (D1 a D4) podemos extraer lo siguiente:

- GT = Dato de referencia asociado al objeto detectado por el algoritmo.
- FT = Objeto detectado por el algoritmo que no tiene suficiente asociación (dependiendo del umbral utilizado en el traslape) con algún dato en la trayectoria de referencia.
- MT = Dato en la trayectoria de referencia que no tiene suficiente asociación con algún objeto detectado.
- $OT = GT + FT$. Número de objetos detectados por el algoritmo.
- $RT = GT + MT$. Número de objetos en la trayectoria de referencia.
- Precisión: Número de GT / Número de OT . Del total de objetos detectados por el algoritmo, cuántos se detectaron correctamente.
- Sensitividad: Número de GT / Número de RT . Del total de objetos en la trayectoria de referencia, cuántos fueron correctamente detectados por el algoritmo.
- F-contador: $2 * \text{Precisión} * \text{Sensitividad} / (\text{Precisión} + \text{Sensitividad})$.

Esta métrica pone en evidencia la persistencia de los objetos a través del tiempo.

2.3.3. Evaluación en tiempo de seguimiento.

Este criterio mide el porcentaje de tiempo durante el cual un objeto presente en la trayectoria de referencia ha sido observado y seguido por el algoritmo de seguimiento con un mismo identificador a través del periodo de seguimiento.

Definición: La asociación entre un dato en la trayectoria de referencia (RD) y un objeto en los resultados del algoritmo (C) es hecho mediante su "área mínima que contiene al objeto" (usando las distancias D1 a D4 y el umbral adecuado) y con la construcción de un identificador a través del tiempo. Con la asociación, obtenemos múltiples candidatos. Se elige uno de ellos siguiendo alguno de éstos criterios:

1. El candidato es el primer objeto físico que corresponde con el dato de referencia.
2. El candidato tiene la intersección más grande con el intervalo de tiempo del dato de referencia.

Una vez elegido un candidato se podrá calcular el tiempo medio durante el cual un dato de referencia tiene un “seguimiento correcto”:

$$T_{seguido} = \frac{1}{NB_{objRef}} \sum_{objref} \frac{card(RD \cap C)}{card(RD)} \quad (8)$$

Donde:

- NB_{objRef} es el número de objetos en la trayectoria de referencia.
- $\frac{card(RD \cap C)}{card(RD)}$ es el traslape obtenido entre el objeto en la trayectoria de referencia y el objeto detectado por el algoritmo.

2.3.4. Fragmentación ID.

Esta métrica sirve para evaluar la persistencia de los identificadores que puede obtener un objeto. Es decir, mide cuantos objetos o regiones en los resultados del algoritmo son asociados a un objeto en la trayectoria de referencia (fragmentación ID).

Definición: Número de identificadores (IDs) diferentes que puede tomar un objeto en la trayectoria de referencia ($NumObjetoID_{objref}$). Los diferentes IDs de los objetos detectados son identificados mediante el área mínima que contiene al objeto y con el hecho de que sus intervalos de intersección con el objeto de referencia son disjuntos.

$$Persistencia = \frac{1}{NB_{objref}} \sum_{objref} \frac{1}{NumObjetoID_{objref}} \quad (9)$$

Donde:

- NB_{objref} es el número objetos en la trayectoria de referencia.

Nota: La persistencia más alta es la mejor de las persistencias para un ID.

2.3.5. Criterio de confusión para IDs.

Este criterio se calcula de manera similar al anterior, pero ahora de manera inversa se identifica cuantos objetos en la trayectoria de referencia son asociados a un objeto en los resultados del algoritmo. Un ejemplo de confusión puede ser el de una persona que se encuentra con otra y al estar muy cerca sus identificadores se intercambian.

Definición: Número de IDs para un dato de referencia que puede tomar un objeto detectado

($NumIDRef_{objAlg}$). El ID diferente del dato de referencia es identificado con el área mínima que contiene al objeto y con el hecho de que sus intervalos de intersección con los objetos de los resultados del algoritmo son disjuntos. Para transformar el resultado en un porcentaje, se calcula la inversa:

$$Confusión = \frac{1}{NB_{objAlg \text{ as } objref}} \sum_{objAlg \text{ as } objref} \frac{1}{NumIDRef_{ObjAlg}} \quad (10)$$

Donde:

- $NB_{objAlg \text{ as } objref}$ es el número e objetos detectados por el algoritmo asociados a un objeto en la trayectoria de referencia.

2.3.6. Trayectorias 2D

Esta métrica estima si dos trayectorias de objetos son detectadas correctamente durante su intervalo de presencia en la escena o no. Para tener una buena comprensión de esta métrica será necesario conocer algunos conceptos primero:

Definición de Trayectoria.

Una Trayectoria es una secuencia de puntos a través del tiempo. La definición general de Trayectoria T es una secuencia de posiciones (x_i, y_i) y tiempos correspondientes t_i . Se puede observar un ejemplo en la figura 5.

De esta forma podemos representar una trayectoria como:

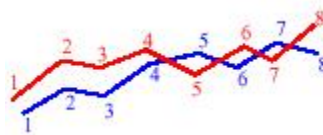


Figura 5: Par de trayectorias.

$$T = (x_1, y_1, t_1), (x_2, y_2, t_2), \dots, (x_n, y_n, t_n).$$

En el dominio de la visión por computadora, esta ecuación se transforma en:

$$T = (x_1, y_1), (x_2, y_2), \dots, (x_n, y_n)$$

i.e. la trayectoria T es una secuencia de posiciones (x_i, y_i) en un tiempo i . Como se ilustra en la figura 5, las rutas se distinguen de las trayectorias mediante la definición de una ruta como una trayectoria no parametrizada por tiempo.

Para evaluar el desempeño del seguimiento, las métricas necesitan comparar dos trayectorias. Tenemos dos trayectorias T_A y T_B las cuales representan la trayectoria obtenida por el algoritmo de seguimiento, y su trayectoria de referencia. Las métricas comparan las trayectorias permitiendo identificar qué tan similares o diferentes son entre si.

Comparación de Trayectorias.

Una forma usada comúnmente para comparar trayectorias es medir la distancia entre una trayectoria y otra. Considere dos trayectorias compuestas de posiciones en 2D en una secuencia de tiempo. Supongamos una trayectoria $T_A = (x_i, y_i)$ y otra como $T_B = (p_i, q_i)$, para cada tiempo i . La figura 6 muestra dos trayectorias e identifica la distancia entre posiciones.

Las distancias entre las posiciones en tiempo i están dadas por d_i :

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

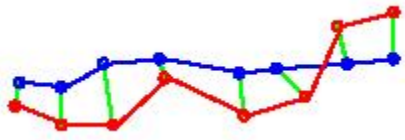


Figura 6: Comparación de trayectorias.

Una métrica usada comúnmente para la evaluación de trayectorias es la media de sus distancias. Se sugiere llamarla m_1 .

$$m_1 = \mu(d_i) = \frac{1}{n} \sum_{i=1}^n d_i \quad (11)$$

Donde:

- n es el número de puntos en la trayectoria.
- d_i es la distancia obtenida entre el punto de la trayectoria A y el punto de trayectoria B en tiempo i .

Esta fórmula provee la distancia promedio entre posiciones en cada intervalo de tiempo.

Otras estadísticas proveen información acerca de la distribución. Aquí se identifica la media, mediana, desviación estándar, valores máximos y mínimos para el conjunto de datos. Éstas permiten definir $D(T_A, T_B)$ para ser el conjunto de distancias d_i entre las trayectorias A y B.

Las estadísticas anteriores pueden ser aplicadas a este conjunto:

$$\text{Media : } \mu(D(T_A, T_B)) = \frac{1}{n} \sum_{i=1}^n d_i \quad (12)$$

$$\text{Mediana : } \text{mediana}(D(T_A, T_B)) = d_{\frac{n+1}{2}} = \frac{1}{2}(d_{\frac{n}{2}} + d_{\frac{n}{2}+1}) \quad (13)$$

$$\text{Desviación estándar : } \sigma(D(T_A, T_B)) = \sqrt{\frac{1}{n} \sum_{i=1}^n (d_i - \mu(d_i))^2} \quad (14)$$

$$\text{Mínimo : } \min(D(T_A, T_B)) = d_i \text{ más pequeña} \quad (15)$$

$$\text{Máximo : } \max(D(T_A, T_B)) = d_i \text{ más grande} \quad (16)$$

Trayectorias Separadas Espacialmente.

Algunas trayectorias pueden ser muy similares, excepto por una diferencia constante en un determinado tiempo, se observa con mayor claridad en la figura 7.

Se define una métrica que calcula la relación entre trayectorias: dadas dos trayectorias T_A y T_B , es posible calcular la traslación espacial óptima \hat{d} de T_A sobre T_B . \hat{d} es el desplazamiento promedio entre trayectorias, y se calcula como:

$$\hat{d} = \mu(d_i) = \frac{1}{n} \sum_{i=1}^n d_i \quad (17)$$

Ahora se puede definir $D(T_A + \hat{d}_i, T_B)$ para fijar las distancias entre una trayectoria trasladada T_A (por \hat{d}) y T_B . Se pueden aplicar las mismas estadísticas a este conjunto, $D(T_A + \hat{d}_i, T_B)$, para describir las distancias. $\mu(D(T_A + \hat{d}_i, T_B)) < \mu(D(T_A, T_B))$ en todos los casos excepto cuando las trayectorias ya son óptimas y espacialmente alineadas.

Cuando $\mu(D(T_A + \hat{d}_i, T_B))$ es significativamente más pequeño que $\mu(D(T_A, T_B))$, esto puede significar un error de seguimiento en una diferencia espacial consistente entre la posición verdadera del objeto y la posición obtenida por el seguimiento.

Trayectorias Temporalmente Separadas.

En este caso las trayectorias son muy similares excepto por una diferencia constante de tiempo. Se define una métrica que revela la relación entre dos trayectorias, como se observa en la figura 8. Dadas dos trayectorias T_A y T_B , es posible calcular la traslación temporal óptima j de T_A sobre T_B .

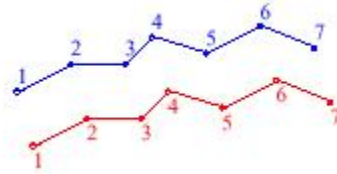


Figura 7: Ejemplo de trayectorias separadas espacialmente.

Cuando el tiempo-variante j es positivo $T_{A,i}$ es mejor asociada con $T_{B,i+j}$, y cuando j es negativo $T_{A,i+j}$ es mejor asociada con $T_{B,i}$. Y j es calculado por:

$$j = \arg \min_k \left(\frac{i}{n - |k|} \sum_{i=Q}^R |(p_{i+k}, q_{i+k}) - (x_i, y_i)| \right) \quad (18)$$

Donde:

$$\begin{cases} Q = 0 & , \text{ si } k \geq 0 \\ Q = -k & , \text{ en caso contrario.} \end{cases} \quad (19)$$

$$R = Q + n - |k|$$

Ahora se puede definir $D(T_A, T_B, j)$ para ser el conjunto de distancias entre una trayectoria temporalmente trasladada T_A o T_B , dependiendo del signo de j . Las mismas estadísticas pueden ser aplicadas a este conjunto, $D(T_A, T_B, j)$, para describir las distancias. Cuando $\mu(D(T_A, T_B, j)) < \mu(D(T_A, T_B))$ en todos los casos, excepto cuando las trayectorias ya son óptimas y espacialmente alineadas.

Cuando $\mu(D(T_A, T_B, j))$ es significativamente más pequeño que $\mu(D(T_A, T_B))$, esto puede significar un error de seguimiento en una diferencia espacial consistente entre la posición verdadera del objeto y la posición obtenida por el seguimiento.

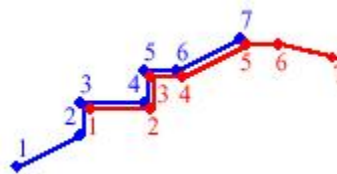


Figura 8: Ejemplo de trayectorias temporalmente separadas.

Trayectorias Separadas Espacio-Temporalmente.

Se define $D(T_A + \hat{d}', T_B, j)$ como el conjunto de distancias entre trayectorias óptimamente alineadas espacial y temporalmente, donde $\hat{d}' = \hat{d}(T_A, T_B, j)$ es el cambio espacial óptimo entre trayectorias temporalmente trasladadas (por j instantes de tiempo).

El procedimiento para definir este conjunto es similar al anterior: calcular el j óptimo para el cual la distancia media de las posiciones entre espacio(traslación de \hat{d}') y tiempo(cambio de j) es minimizado, usando una búsqueda exhaustiva. El primer j debe ser calculado, el conjunto de distancias $D(T_A + \hat{d}', T_B, j)$ puede ser formado y se pueden usar las estadísticas usuales.

Cuando las trayectorias están espacial y temporalmente alineadas, el valor medio, $\mu(D(T_A + \hat{d}', T_B, j))$ es menor o igual que el valor medio de otros tres conjuntos de distancias, cuando las trayectorias son espacialmente alineadas o temporalmente alineadas.

Área entre Trayectorias.

Dadas dos trayectorias A y B, el área entre ellas se calcula mediante el conjunto de puntos donde la trayectoria A se intersecta con la trayectoria B. Éstos puntos se usan para definir un conjunto de regiones. Esto se observa con mayor claridad en la figura 9.



Figura 9: Área entre trayectorias.

El área entre trayectorias puede ser calculada suponiendo regiones separadas. Se calcula suponiendo cada región como un polígono con n -lados. Con los puntos (x_i, y_i) para $i = 1, \dots, n$, donde el primer punto es el punto de intersección entre las trayectorias y el siguiente punto sigue al primer punto de intersección en la trayectoria A.

Trazando el polígono el área bajo cada segmento se calcula como un trapezoide, cada una de éstas áreas se suma o resta del total, dependiendo de su signo, los cuales resultan del cálculo de:

$$A = \frac{(x_{i+1} - x_i)(y_i + y_{i+1})}{2} \quad (20)$$

como el área entre (x_i, y_i) y (x_{i+1}, y_{i+1}) . Después obtenemos la siguiente ecuación:

$$A_{region} = \left| \frac{1}{2} \left(\left(\sum_{i=1}^{n-1} x_{i+1}y_i \right) + x_1y_n \right) - \left(\left(\sum_{i=1}^{n-1} x_1y_{i+1} \right) + x_ny_1 \right) \right| \quad (21)$$

La suma de las áreas de cada una de las regiones provee el área total entre las rutas, tiene dimensionalidad L^2 . Para obtener un valor útil para la métrica del área, el área calculada es

normalizada mediante el tamaño promedio de las trayectorias. Lo que representa el promedio de la distancia independiente del tiempo entre dos trayectorias y es una **distancia promedio continua**.

Definición de la métrica: Número de trayectorias detectadas por el algoritmo comparadas con las trayectorias en los datos de referencia. La asociación entre dos trayectorias se realiza mediante el cálculo de la distancia máxima entre ellas y la definición de un umbral adecuado. De esta manera podemos obtener los siguientes datos:

- GT = Trayectorias de datos de referencia asociadas a trayectorias de objetos físicos.
- FT = Trayectorias de objetos físicos no asociadas a alguna trayectoria de datos de referencia.
- MT = Trayectorias de datos de referencia no encontrados.
- $OT = GT + FT$
- $RT = GT + MT$
- Precisión: $\text{Número de } GT / \text{Número de } OT$
- Sensitividad: $\text{Número de } GT / \text{Número de } RT$
- F-contador: $2 * \text{Presición} * \text{Sensitividad} / (\text{Presición} + \text{Sensitividad})$.

2.4. Resultados de las métricas.

Cuando medimos la habilidad de un algoritmo para enfrentar cierto problema, se usan métricas que pueden devolver como salida un conjunto de localizaciones de error en 2D, un intervalo de tiempo entre la aparición del objeto en la **Trayectoria de Referencia** y su detección por el sistema, en estos casos las métricas son calculadas estadísticamente (media, desviación, valores mínimos y máximos, etc.)

También se puede obtener un número de éxitos o fallas del algoritmo frente a diferentes situaciones. En este caso se usan definiciones estándar:

- **Los Verdaderos Positivos (TP):** El sistema detecta una situación real. (Existente en la realidad y en el resultado del algoritmo)
- **Los Falsos Negativos (FN):** Una situación real ha sido omitida por el sistema. (Existe solo en la realidad pero el sistema no la detectó)
- **Los Falsos Positivos (FP):** El sistema ha detectado una situación que no es real. (Existente únicamente en los resultados del algoritmo)
- **Los Verdaderos Negativos (TN):** Una entidad que no se encuentra en la realidad y tampoco en los resultados del algoritmo.

Estos datos pueden proporcionar mayor información en cuanto a las características del algoritmo, como:

1. La precisión = $\frac{TP}{(TP+FP)}$
2. La sensibilidad = $\frac{TP}{(TP+FN)}$
3. La especificación = $\frac{TN}{(FP+TN)}$
4. El F-contador = $\frac{2 * precisión * sensibilidad}{(precisión + sensibilidad)}$

Según [21], dichos criterios y métricas se organizan considerando las siguientes tareas:

- Detección de objetos físicos de interés.
- Localización de objetos físicos de interés.

- Seguimiento de objetos físicos de interés.
- Clasificación de objetos físicos de interés.
- Reconocimiento de eventos.

2.5. Métricas usadas comúnmente.

Esta parte de la evaluación estima si la correspondencia entre dos objetos físicos detectados en dos instantes de tiempo es calculado correctamente por el algoritmo. Para ello se cuenta con diferentes métricas definidas. Las siguientes se encuentran en [3]:

1. Tasa de Detección de Seguimiento

$$(TRDR) = \frac{\text{Total Verdaderos Positivos}}{\text{Número Total de Puntos en la Trayectoria de Referencia}} \quad (22)$$

2. Tasa de Falsa Alarma

$$(FAR) = \frac{\text{Total Falsos Positivos}}{\text{Total Verdaderos Positivos} + \text{Total Falsos Positivos}} \quad (23)$$

3. Tasa de Detección de Trayectoria

$$(TDR) = \frac{\text{Número de Verdaderos Positivos por objeto Seguido}}{\text{Número total de puntos en la Trayectoria de Referencia por objeto}} \quad (24)$$

4. Error de Seguimiento de Objeto

$$(OTE) = \frac{1}{N_{rg}} \sum_{\exists ig(t_i) \wedge r(t_i)} \sqrt{(xg_i - xr_i)^2 + (yg_i - yr_i)^2} \quad (25)$$

Donde:

- N_{rg} es el número de imágenes que tienen en común la trayectoria de referencia y la trayectoria obtenida con el algoritmo de seguimiento.
- $(xg_i, yg_i), (xr_i, yr_i)$ es la localización del punto en la trayectoria de referencia y el punto resultante del algoritmo en la imagen i respectivamente.

5. Tasa de Oclusiones Ocurridas (OSR)

$$\frac{\text{Numero de oclusiones exitosas}}{\text{Numero total de oclusiones}} \quad (26)$$

6. Tasa de Seguimiento Ocurrido (TSR)

$$\frac{\text{Numero de objetos seguidos no fragmentados}}{\text{Numero total de objetos en la trayectoria de referencia}} \quad (27)$$

2.6. Trabajos Relacionados.

Existen diferentes trabajos desarrollados anteriormente, algunos de los más actuales son los siguientes:

2.6.1. Un Método de Evaluación del Desempeño para Seguimiento de Objetos en Video.

En [3] se propone una metodología para evaluar el desempeño de sistemas de video vigilancia usando videos pseudo-sintéticos, que emplean datos capturados en línea y almacenados en una base de datos de vigilancia.

Las trayectorias se toman directamente de la base de datos, después se genera la **Trayectoria de referencia**. Mediante el análisis de la misma se puede cuantificar la calidad de los algoritmos de seguimiento de objetos. Las métricas propuestas para éste proyecto son las siguientes:

- Número de Objetos Seguidos en un Tiempo (NOST)
- Evaluación del Tiempo de Seguimiento (ETS)
- Fragmentación (TDR)
- Criterio de Confusión (CC)
- Trayectorias de Objetos 2D (TO2D)

Este sistema fue evaluado con 200 secuencias de video diferentes, con un total de 300,000 imágenes. Lo cual equivale a 800 minutos de video.

2.6.2. Métricas y Estadísticas de Evaluación del Desempeño para Seguimiento Posicional.

Chris J. Needham y Roger D. Boyle muestran en [17] un análisis de diferentes conceptos para la evaluación posicional del seguimiento, es decir, las métricas propuestas en este trabajo permiten

determinar qué tan buena es la posición de un objeto detectado por el algoritmo de seguimiento. Los datos se tomaron del seguimiento de un jugador de futbol , y el seguimiento de un vehículo.

El objetivo específico de este trabajo fue comparar dos trayectorias; el artículo muestra formas específicas que permiten comparar dos trayectorias con diferentes métricas estadísticas, como el área entre dos trayectorias, la distancia, etc, lo cual da como resultado la precisión de una trayectoria con respecto a otra.

2.6.3. Métricas de Evaluación del Desempeño para Detección de Movimiento y Seguimiento de Objetos.

En [18] se proporciona una metodología para la evaluación operacional de un sistema de video vigilancia, en términos de robustez y confiabilidad. Se propone la creación de secuencias de video semi-sintéticas y completamente sintéticas, es decir, secuencias de video diseñadas específicamente con variaciones de escenarios y eventos entre objetos. Todo esto se evalúa cuantitativamente mediante métricas de error.

Las métricas de error utilizadas para la evaluación del desempeño del seguimiento son las siguientes:

1. **Tasa acertada:** Número de objetos detectados exitosamente en una imagen con respecto al número de objetos en la trayectoria de referencia.
2. **Tasa perdida:** Número de objetos no detectados en una imagen con respecto al número de objetos en la trayectoria de referencia. Es complementaria a la tasa acertada:
Tasa perdida = 1 - Tasa acertada.
3. **Falsos intentos:** Número de hipótesis del seguimiento que no corresponden a ningún objeto en la trayectoria de referencia para el número total de imágenes en la secuencia.
4. **Cambios correspondientes:** Número de cambios correspondientes con respecto al número de trayectorias en la trayectoria de referencia.

2.6.4. Observación Parcial contra Seguimiento Ciego a través de Oclusiones.

Ming Xu y Tim Ellis proponen dos métodos de evaluación cuantitativos. El primero mide el error del seguimiento usando la siguiente fórmula:

$$e_k = \|z_k - \hat{z}_k\| \quad (28)$$

Donde z_k es la medida actual y \hat{z}_k es la siguiente medida.

El segundo método es conocido como coherencia de ruta, definido de la siguiente manera:

$$\Phi_k = w_1 \left\{ 1 - \frac{|s_k * s_{k+1}|}{\|s_k\| \|s_{k+1}\|} \right\} + w_2 \left\{ 1 - 2 \frac{\sqrt{\|s_k\| \|s_{k+1}\|}}{\|s_k\| + \|s_{k+1}\|} \right\} \quad (29)$$

Donde:

- s_k es el segmento entre el centroide del área mínima que contiene al objeto estimado en dos imágenes consecutivas.
- $w_1 = 0,5$ y $w_2 = 0,5$
- $\Phi_k \in [0, 1]$

3. Herramientas para Análisis de Algoritmos de Seguimiento de Objetos.

El proceso general que sigue el sistema propuesto para la evaluación de un algoritmo de seguimiento de objetos en movimiento se describe en la siguiente figura, como se observa existen 3 módulos principales que definen el comportamiento del sistema (círculo rojo):

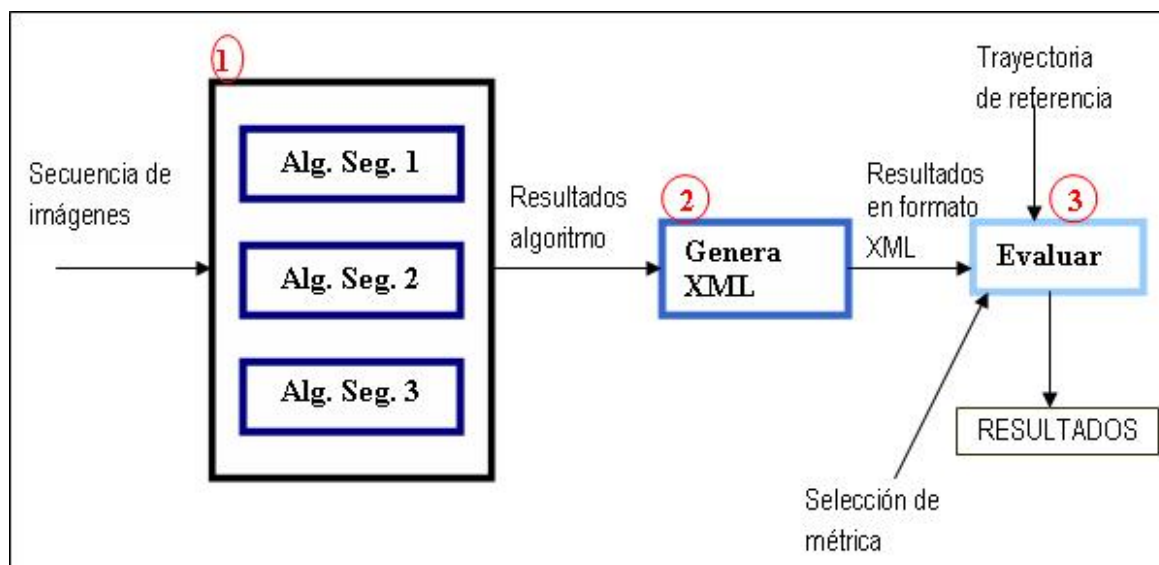


Figura 10: Evaluación automática de algoritmos de seguimiento.

La comunicación entre estos módulos se hizo mediante llamados a procesos ejecutables (.exe), con la ayuda de la instrucción WinExec de Visual c++. Los parámetros para cada módulo se listan después del nombre del proceso, por ejemplo:

WinExec("Evaluar trayectoria, metrica")

El nombre del proceso a ejecutar es *Evaluar* y sus parámetros son *trayectoria* y *metrica*. Posteriormente en la ejecución del proceso se hace una lectura de parámetros mediante apuntadores y así se conocen los parámetros necesarios.

3.1. Modulo 1. Algoritmos de seguimiento.

Este primer módulo recibe como entrada una secuencia de imágenes proporcionada por el proyecto CAVIAR[13]. Para probar el sistema se seleccionaron 3 secuencias, las cuales tienen formato jpg y un tamaño de 384x288.

Una vez seleccionada la secuencia de imágenes el usuario debe elegir uno de los tres algoritmos de seguimiento que se tienen disponibles para evaluar:

- **Algoritmo 1 (Correlación).** Este algoritmo usa los valores de los píxeles de cada imagen para representar el objeto de interés. En este tipo de algoritmos, se comparan conjuntos o regiones rectangulares de píxeles de dos imágenes de $M \times N$, para cada región de la imagen de la derecha, se busca una región de correspondencia en la imagen de la izquierda usando una medida de similitud. La correlación cruzada es una propuesta estadística comúnmente usada para el problema de correspondencia o reconocimiento de patrones. La medida de similitud de correlación cruzada se define una solución estadística debido a que utiliza el criterio del error cuadrático medio, está dada por la siguiente ecuación:

$$CCS(u, v) = \frac{\sum_x \sum_y T(x, y)I(x - u, y - v)}{[\sum_x \sum_y I^2(x - u, y - v)]^{\frac{1}{2}}} \quad (30)$$

donde x, y y u, v son las posiciones y el desplazamiento entre las imágenes. El valor obtenido de esta ecuación está en un intervalo de -1 a 1, donde $c(u, v) = 1$, indica una correlación perfecta positiva.

- **Algoritmo 2 (Diferencia de imágenes).** La detección de bordes es un proceso que trata de obtener propiedades significativas de los objetos de interés en la imagen. Las propuestas más comunes de detección de bordes incluyen tres operaciones: diferenciación, suavizado y etiquetado. La diferenciación consiste en evaluar las derivadas de la imagen, el suavizado consiste en reducir el ruido en la imagen y finalmente, el etiquetado corresponde a la localización de bordes e incremento de la razón señal a ruido con la finalidad de eliminar falsos bordes.
- **Algoritmo 3 (Sustracción de fondo adaptable).** El principal objetivo de la sustracción del fondo es detectar a todos los objetos que presentan movimiento dado un conjunto de imágenes obtenidas a partir de una cámara fija. El proceso de sustracción del fondo consiste en calcular una imagen de fondo, obtener la diferencia absoluta de cada imagen de entrada con la imagen de referencia y finalmente segmentar el resultado. El resultado de este proceso es una segmentación binaria de la imagen con regiones de objetos que presentan movimiento.

$$|I_i - F_i| > Umbral \quad (31)$$

donde, I_i es la imagen de entrada en el tiempo i y F es la imagen de referencia o fondo en el mismo tiempo y $Umbral$ es el valor de nivel de gris para la segmentación.

Algunas propuestas que existen para obtener el fondo de manera automática y adaptable a los cambios en el escenario son el promedio o la media de n imágenes [16], que requiere mucha memoria. Cabe mencionar que este algoritmo es para seguimiento de múltiples blancos.

Se encuentran mayores detalles de cada algoritmo en [7].

Los algoritmos arrojan sus resultados en un archivo de texto denominado Reporte.txt que debe tener un formato como el que se muestra en el siguiente ejemplo:

Imagen: 30

Identificador	Fila	Columna	X-MIN	Y-MIN	X-MAX	Y-MAX
1	188.114029	84.913742	131.0	357.0	205.0	397.0
2	208.064102	162.59936	144.0	192.0	190.0	224.0
3	246.848007	149.52800	137.0	235.0	164.0	261.0
4	253.917496	83.705002	63.0	240.0	103.0	267.0
5	277.140625	292.94271	264.0	257.0	318.0	286.0

Imagen: 32

Identificador	Fila	Columna	X-MIN	Y-MIN	X-MAX	Y-MAX
1	189.065567	86.116756	132.0	359.0	207.0	402.0
2	206.728699	160.21565	141.0	191.0	186.0	222.0
3	246.180348	149.10619	137.0	235.0	164.0	261.0
4	254.244553	82.796608	67.0	243.0	99.0	270.0
5	277.163025	291.82464	264.0	257.0	318.0	286.0
6	241.319260	218.33210	201.0	226.0	244.0	258.0
7	257.778748	115.13274	105.0	248.0	125.0	266.0

Donde Identificador es el número que identifica al objeto que se está siguiendo, Fila es la posición en el eje x en que se detectó al objeto, Columna es la posición en el eje y en que se detectó al objeto, X-MIN es la coordenada x mínima del área mínima que contiene al objeto, Y-MIN es la coordenada y mínima del área mínima que contiene al objeto, X-MAX es la coordenada x máxima del área mínima que contiene al objeto, Y-MAX es la coordenada x máxima del área mínima que contiene al objeto. Esto se observa claramente en la siguiente figura:

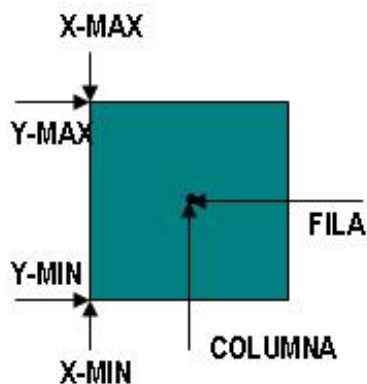


Figura 11: Coordenadas de objeto.

3.2. Módulo 2. Generar XML.

Este módulo recibe como entrada el reporte generado en el módulo anterior, y a partir de estos datos genera un archivo XML con el formato requerido por el proyecto ETISEO [15]:

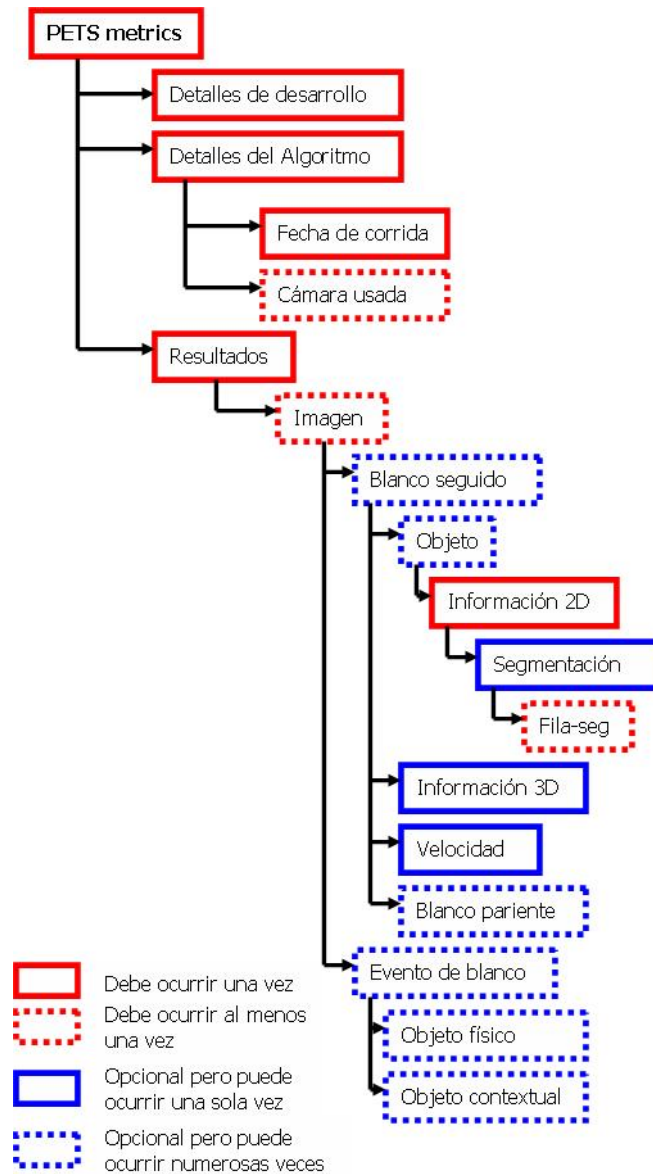


Figura 12: Formato de datos XML.

Un ejemplo del formato anterior escrito en XML es el siguiente:

```
<EXTENDED_PETS_METRICS
  xmlns="http://www.cvg.cs.reading.ac.uk/PETSMETRICS"
  xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
  xsi:schemaLocation = "http://www.cvg.cs.reading.ac.uk/PETSMETRICS
  http://www.cvg.cs.reading.ac.uk/PETSMETRICS/XML/Metrics.xsd">
  <developer_details autor="" email="" institucion="">
  </developer_details>
  <algorithm_details version="" tarea="" conjunto_datos="" numero_conjunto=""
  resultdos="1">
    <run_date dia="" mes="" ano="">
    </run_date>
    <camera_used camera_id = "1">
    </camera_used>
  </algorithm_details>
  <results>
    <frame frame_id="0">
    </frame>
    <...>
    <frame frame_id="151">
      <tracked_target target_id="1" start_frame_track="150"
      end_frame_track="151" status="0">
        <object object_id="1" classification="4">
          <info_2d xmin="234" xmax="285" x_center="259" ymin="574" ymax="507"
          y_center="540" viewed_in_camera_id="1">
          </info_2d>
        </object>
      </tracked_target>
      <tracked_target target_id="2" start_frame_track="150"
      end_frame_track="151" status="0">
      <...>
      </tracked_target>
      <event_target target_id="10" name="inside" start_frame="150"
      end_frame="350" status="0">
      </event_target>
    </frame>
  </results>
</EXTENDED_PETS_METRICS>
```

El proceso general que se realiza en este módulo es el siguiente:

- Se hace la lectura línea por línea del Reporte.txt generado en el módulo anterior. Se pueden obtener tres tipos de línea:

1. Número de imagen en la que se encuentra el algoritmo:

Imagen: 30

2. Encabezado para los datos:

Identificador	Fila	Columna	X-MIN	Y-MIN	X-MAX	Y-MAX
---------------	------	---------	-------	-------	-------	-------

3. Datos obtenidos por el algoritmo para cada objeto:

1	247.226761	87.323418	72.0	231.0	100.0	262.0
---	------------	-----------	------	-------	-------	-------

- Se divide la línea obtenida por tokens y se guardan los datos necesarios. El tipo de línea número 2 es ignorada debido a que no proporciona información relevante.
- Se envían los datos a los métodos encargados de generar el archivo XML. Estos métodos están hechos con vectores que son creados de acuerdo al número de datos obtenidos en el Reporte.txt.

3.3. Modulo 3. Evaluación.

En este módulo se realiza todo el trabajo referente a las métricas, recibe como entradas el reporte generado en formato XML, la Trayectoria de referencia en formato XML (proporcionada por el proyecto CAVIAR) y el tipo de métricas con que se desea evaluar el algoritmo.

El primer paso es realizar una lectura de ambos archivos XML almacenando sus datos en una estructura. Como ya se mencionó anteriormente se implementaron seis métricas y para esto se hizo uso de la distancia D2 mencionada en la sección 2.3, esta distancia mide la cardinalidad del traslape que hay entre dos objetos. Para ello se tuvieron que analizar diferentes casos entre objetos:

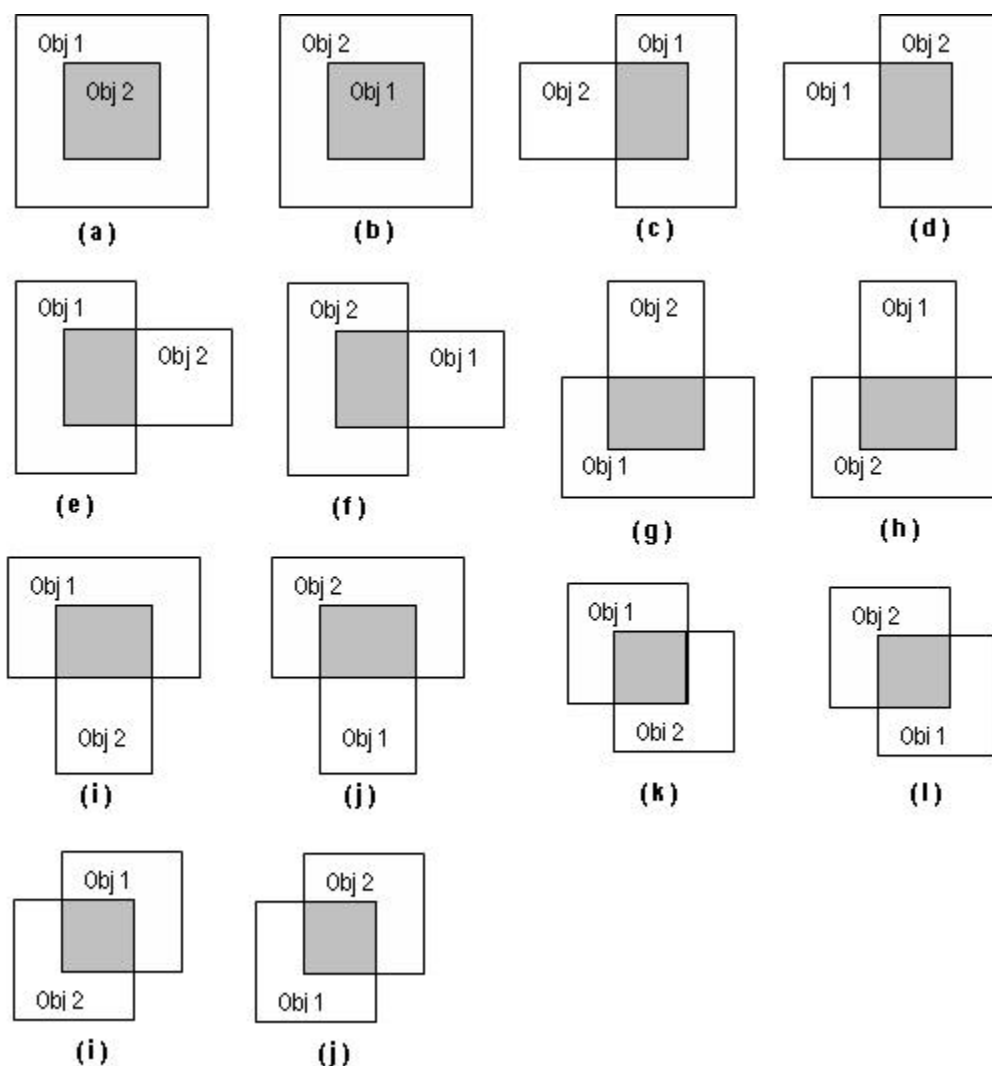


Figura 13: Casos de traslape entre objetos para distancia D2.

Estos casos representan todas las maneras posibles en que puede existir una intersección no vacía entre el área mínima que contiene a los objetos en la trayectoria de referencia y el área mínima que contiene a los objetos en los resultados del algoritmo, En cada caso se debe calcular el traslape de manera diferente, para ello es necesario conocer los siguientes datos:

- \mathbf{T} = Traslape
- $\mathbf{x1x}$ = Coordenada x máxima del objeto 1.
- $\mathbf{x1m}$ = Coordenada x mínima de objeto 1.
- $\mathbf{y1x}$ = Coordenada y máxima del objeto 1.
- $\mathbf{y1m}$ = Coordenada y mínima de objeto 1.
- $\mathbf{w1}$ = Anchura del objeto 1.
- $\mathbf{h1}$ = Altura de objeto 1.
- $\mathbf{x2x}$ = Coordenada x máxima de objeto 2.
- $\mathbf{x2m}$ = Coordenada x mínima de objeto 2.
- $\mathbf{y2x}$ = Coordenada y máxima del objeto 2.
- $\mathbf{y2m}$ = Coordenada y mínima de objeto 2.
- $\mathbf{w2}$ = Anchura del objeto 2.
- $\mathbf{h2}$ = Altura de objeto 2.

Así se obtiene:

Caso (a)

$$T = w2 * h2$$

Caso (b)

$$T = w1 * h1$$

Caso (c)

$$T = (x2x - x1m) * h2$$

Caso (d)

$$T = (x1x - x2m) * h1$$

Caso (e)

$$T = (x1x - x2m) * h2$$

Caso (f)

$$T = (x2x - x1m) * h2$$

Caso (g)

$$T = (y1x - y2m) * w2$$

Caso (h)

$$T = (y2x - y1m) * w1$$

Caso (i)

$$T = (y2x - y1m) * w2$$

Caso (j)

$$T = (y1x - y2m) * w1$$

Caso (k)

$$T = (y2x - y1m) * (x1x - x2m)$$

Caso (l)

$$T = (y1x - y2m) * (x2x - x1m)$$

Caso (m)

$$T = (y2x - y1m) * (x2x - x1m)$$

Caso (n)

$$T = (y1x - y2m) * (x1x - x2m)$$

La primera métrica definida fue la de “Liga entre imágenes”, cuyo objetivo es verificar qué tan bien se realiza el seguimiento imagen por imagen, es decir, si existen asociaciones entre los objetos de una imagen en tiempo t y los de la imagen en tiempo $t + 1$.

Para ello se toman los datos de los objetos de la imagen en la trayectoria de referencia en tiempo t , se busca la imagen correspondiente (t) en los resultados del algoritmo, se toman los datos de sus objetos y se checa que la cardinalidad del traslape entre ellos sea menor que un umbral establecido (en este caso se usó un octavo de la cardinalidad del área mínima que contiene al objeto en la trayectoria de referencia), si este umbral no es excedido, se obtiene un seguimiento correcto (GT). Si el umbral es menor que la cardinalidad, se puede tener uno de dos casos, el primero da como resultado un seguimiento perdido 1 (MT1) que consiste en el rechazo del segundo caso definido en dicha métrica (una liga existente entre objetos detectados en tiempo $t + 1$ y t y una liga existente solo en los datos de referencia) y el segundo es un seguimiento perdido 2 (MT2) que consiste en rechazar ambos casos.

Y para los falsos seguimientos simplemente se checa cuantos objetos fueron asociados a la trayectoria de referencia, y se hace una resta de los objetos existentes en los resultados del algoritmo y los asociados ($FT = OT - GT$).

La métrica 2.3.2 (Número de objetos detectados en un instante de tiempo) se realizó haciendo una comparación de las coordenadas de los objetos imagen por imagen. Se realizó un ciclo que comienza desde cero y termina cuando se llega al número de imágenes que tiene la trayectoria de referencia. Dentro del ciclo, se busca la imagen correspondiente en el tiempo de los resultados del algoritmo y se comparan los datos de acuerdo a la cardinalidad del traslape de los objetos. En este caso se tomó como umbral la mitad de la cardinalidad del área mínima que contiene al objeto en la trayectoria de referencia, si esta no se excede, entonces se toma como un objeto en la trayectoria asociado a un objeto detectado por el algoritmo. Para este proceso también se usa una bandera, que se enciende cuando hubo asociación de algún objeto en la imagen, y si esta no se enciende, se toma el número de objetos en la trayectoria de referencia como objetos que no tuvieron la suficiente asociación con ningún objeto detectado por el algoritmo (MT).

También puede darse el caso en el que se encuentre un objeto detectado por el algoritmo que no tuviera suficiente asociación con ningún objeto en la trayectoria de referencia (FT). Para detectar este tipo de resultado, se hizo uso de un contador, el cual lleva el número de asociaciones que se hace entre objetos de la trayectoria de referencia y objetos detectados por el algoritmo. De esta

forma al terminar con el número de objetos por imagen en la trayectoria de referencia, se hace una resta del número de objetos detectados por el algoritmo y el número de objetos asociados, obteniendo así el resultado deseado.

Para la programación de la métrica 2.3.3 (Evaluación en tiempo de seguimiento), se hizo necesaria la elaboración de un método que devuelve como resultado la siguiente tupla:

Objeto	Imagen Inicial	Imagen Final	Persistencia	Asociación
1	30	228	199	1
2	70	228	159	-1
3	150	172	23	2
4	184	228	45	-1

donde “Objeto” es el identificador del objeto detectado por el algoritmo, “Imagen Inicial” es el número de imagen en que apareció por primera vez dicho objeto, “Imagen Final” es el número de imagen en que aparece por última vez, “Persistencia” es el número de imágenes en que está presente el objeto y “Asociación” es el identificador del objeto en la trayectoria de referencia al que fue asociado el objeto, si este número es -1, quiere decir que no hubo asociación de dicho objeto con ninguno en la trayectoria de referencia. Una vez obtenida esta tupla, se procede a elegir uno de dos posibles candidatos:

1. El candidato es el primer objeto detectado que se asocia a la trayectoria de referencia.
2. El candidato que tiene la intersección más grande de tiempo con el dato de la trayectoria de referencia.

En este caso elegimos al segundo candidato, debido a que este nos da una mayor confiabilidad a lo largo del tiempo. Partiendo de la tupla obtenida, se puede saber qué objeto tiene una mayor persistencia, y obteniendo sus datos, se procede a calcular lo siguiente:

$$T_{seguido} = \frac{1}{NB_{objtray}} \sum_{objtray} \frac{card(RD \cap C)}{card(RD)} \quad (32)$$

Donde:

- $NB_{objtray}$ es el número de objetos en la trayectoria de referencia.
- $card(RD \cap C)$ es la cardinalidad de la intersección entre el objeto detectado y el existente en la trayectoria de referencia.
- $card(RD)$ es la cardinalidad del objeto en la trayectoria de referencia.

Esta métrica es la base para la métrica 2.3.4 (Fragmentación ID), ya que esta consiste en tomar el número de identificadores diferentes que puede tomar un objeto en la trayectoria de referencia, denominado $NumObjID_{objtray}$ y se procede a calcular la persistencia dad por:

$$P = \frac{1}{NB_{objtray}} \sum_{objref} \frac{1}{NumObjID_{objref}} \quad (33)$$

Esta ecuación da como resultado un número entre 0 y 1, es decir, el porcentaje de persistencia.

Para la métrica “criterio de confusión para IDs” se toma el mismo concepto que la métrica anterior, pero ahora se toma en cuenta el número de identificadores diferentes que puede tomar un objeto detectado por el algoritmo ($NumRefID_{objAlg}$) y se procede a calcular el criterio de confusión mediante:

$$C = \frac{1}{NB_{objAlg \text{ as } objref}} \sum_{objAlg \text{ as } objret} \frac{1}{NumRefID_{objAlg}} \quad (34)$$

donde $NB_{objAlg \text{ as } objref}$ es el número de objetos detectados por el algoritmo asociados al objeto en la trayectoria de referencia.

También se obtiene un porcentaje de esta ecuación.

4. Resultados.

La primera de las secuencias seleccionadas es tomada del primer conjunto de secuencias de CAVIAR, con una cámara de lente ancho en la entrada del lobby de los laboratorios INRIA en Grenoble, Francia. Tiene 610 imágenes, en las que se observa el pasillo interno de los laboratorios, a lo largo de la secuencia se distinguen cuatro objetos etiquetados con los identificadores 0, 1, 4, y 5, así se obtienen 4 trayectorias de referencia, la primera pertenece al objeto 0 y se muestra en la fig 15, donde se puede observar que es continua, en la imagen superior se muestra el objeto, que es una persona caminando. Esta secuencia es identificada por CAVIAR como “Una persona caminando-línea recta .”



Figura 14: Objeto 0 de secuencia de imágenes número 1.

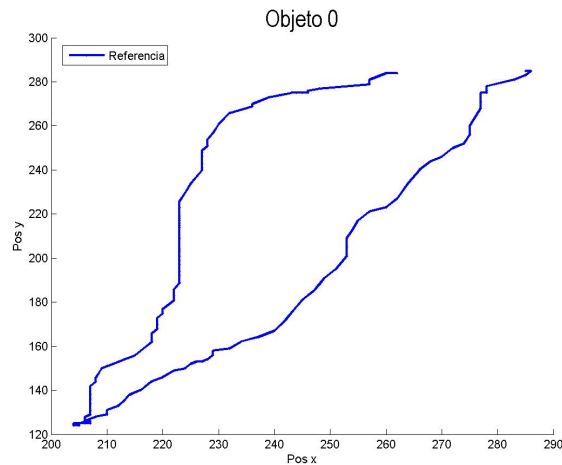


Figura 15: Trayectoria de referencia de objeto 0.

En las figuras siguientes observamos el objeto identificado con el número 1, que es una persona que aparece en la imagen 236 y desaparece en la 511; y su trayectoria respectivamente.



Figura 16: Objeto 1 de secuencia de imágenes número 1.

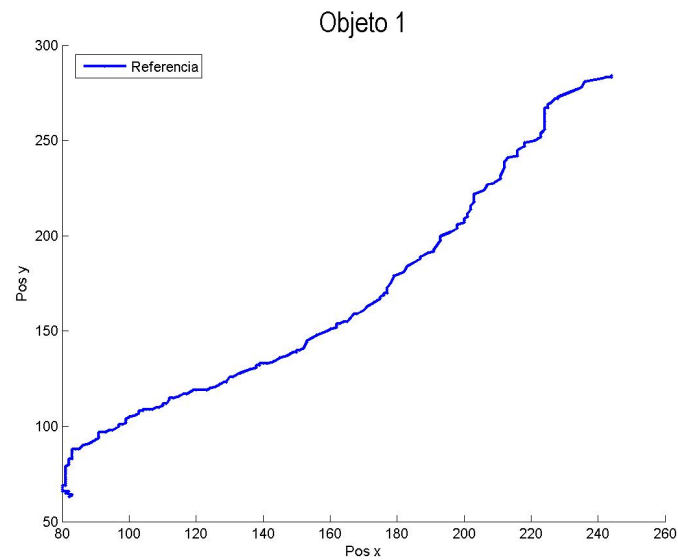


Figura 17: Trayectoria de referencia de objeto 1.

Como se observa con facilidad, la trayectoria de referencia es continua en todo el intervalo de presencia del objeto.



Figura 18: Objeto 4 de secuencia de imágenes número 1.

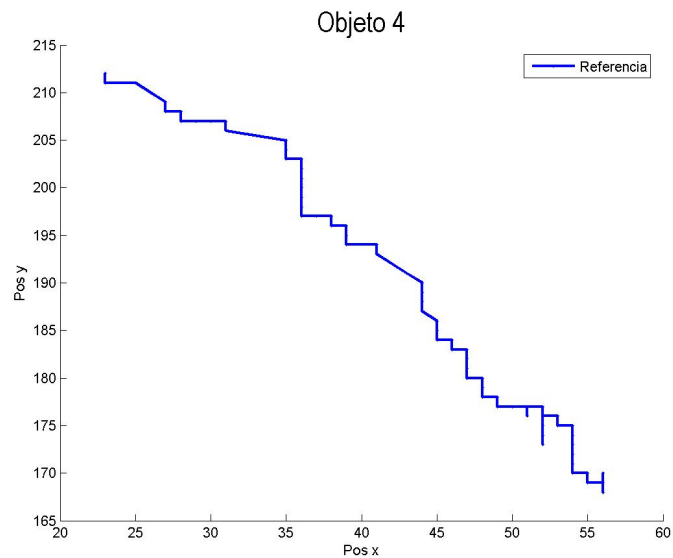


Figura 19: Trayectoria de referencia de objeto 4.

En las figuras 18 y 19 se muestra el objeto identificado con el número 4 y su trayectoria de referencia respectivamente; este objeto parece en la secuencia a partir de la imagen 39 y corresponde a una persona que sale del edificio.

El objeto número 5 (figura 20) es otra persona que se sale del edificio, se observa a partir de la imagen número 63 hasta el final de las secuencia, y su trayectoria de referencia muestra algunos movimientos bruscos, pero en general es continua.



Figura 20: Objeto 5 de secuencia de imágenes número 1.

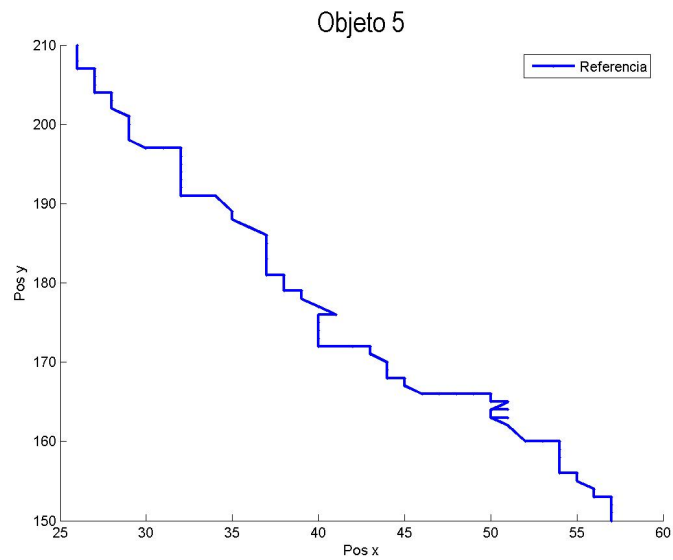


Figura 21: Trayectoria de referencia de objeto 5.

Las siguientes 2 secuencias son tomadas del segundo grupo de secuencias de CAVIAR, en los cuales durante la captura y digitalización hubo algunos errores de segmentación, lo que significa que algunas imágenes fueron duplicadas, así que ocasionalmente dos imágenes consecutivas son iguales. La segunda secuencia es identificada como “Person goes outside store-Three person walking together”, contiene 294 imágenes y en ella se observa el corredor de un centro comercial en Portugal, se identifican tres objetos, que se muestran a continuación junto con sus respectivas trayectorias de referencia.



Figura 22: Objeto 0 de secuencia de imágenes número 2.

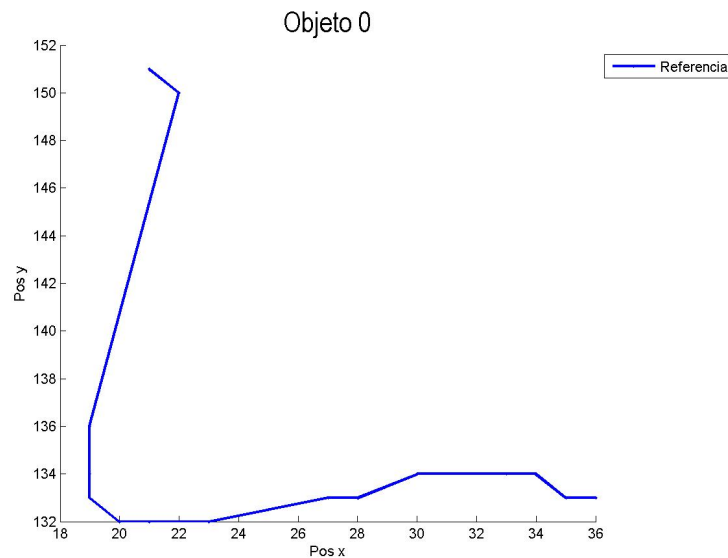


Figura 23: Trayectoria de referencia de objeto 0.

El primer objeto es identificado con el ID 0, éste tiene una presencia corta, pues solo aparece durante el intervalo 0 a 19; representa a una persona caminando a lo largo del pasillo.

A continuación se muestra a una persona dentro del centro comercial que aparece desde la primera imagen de la secuencia, y se pierde en la número 241. Como se observa en su trayectoria de



Figura 24: Objeto 1 de secuencia de imágenes número 2.

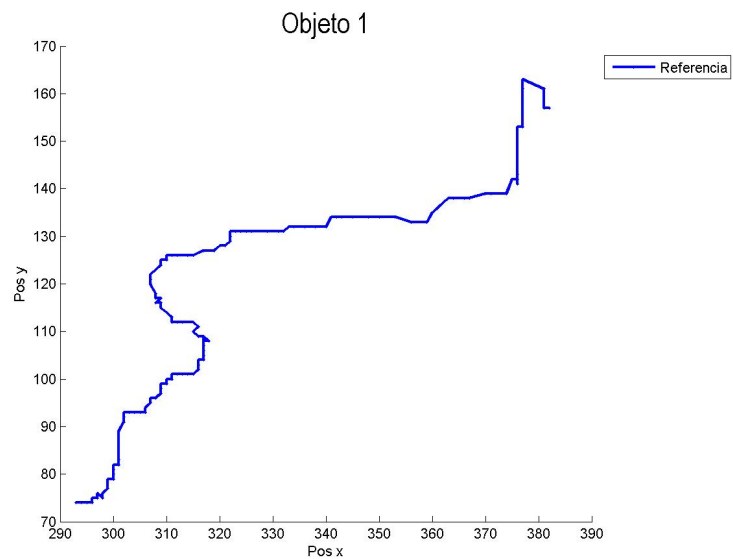


Figura 25: Trayectoria de referencia de objeto 1.

referencia, su movimiento es continuo.

Se observa al último objeto de esta secuencia de imágenes identificado con el número 2, este representa a una persona dentro del centro comercial, que aparece en la imagen número 119 y se pierde en la 294. El intervalo de presencia de este objeto es muy pequeño por lo que su



Figura 26: Objeto 2 de secuencia de imágenes número 2.

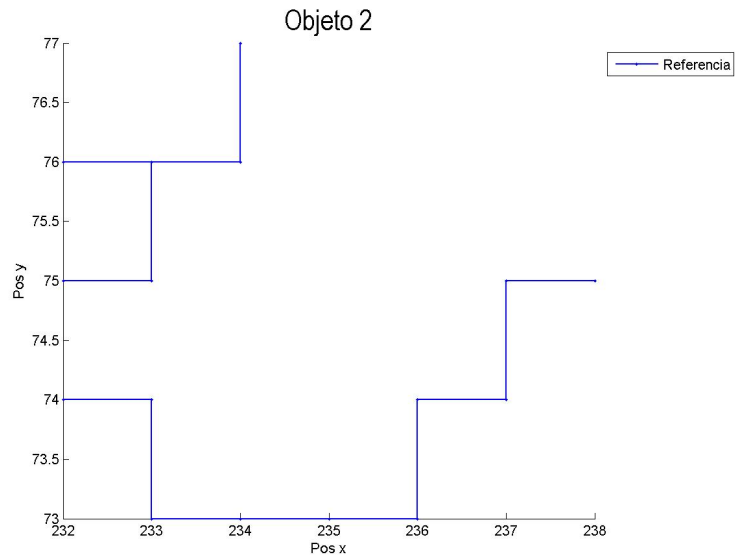


Figura 27: Trayectoria de referencia de objeto 2.

trayectoria de referencia parece ser no continua, pero si se observa el intervalo de presencia, se notará con claridad que la trayectoria de referencia es continua.

La tercera secuencia se identifica como “Dos personas cruzan trayectorias a la entrada de la tienda, pareja caminando en el corredor vista frontal”, conformada por 382 imágenes, en las cuales se distinguen cuatro objetos. Esta secuencia fue tomada por una cámara colocada frente al corredor del centro comercial.

El objeto número 1 se identificó en la trayectoria de referencia con el ID 0, su intervalo de presencia es corto, ya que aparece en la primera imagen y desaparece en la imagen 23, tiene una trayectoria de referencia continua.



Figura 28: Objeto 0 de la secuencia de imágenes número 3.

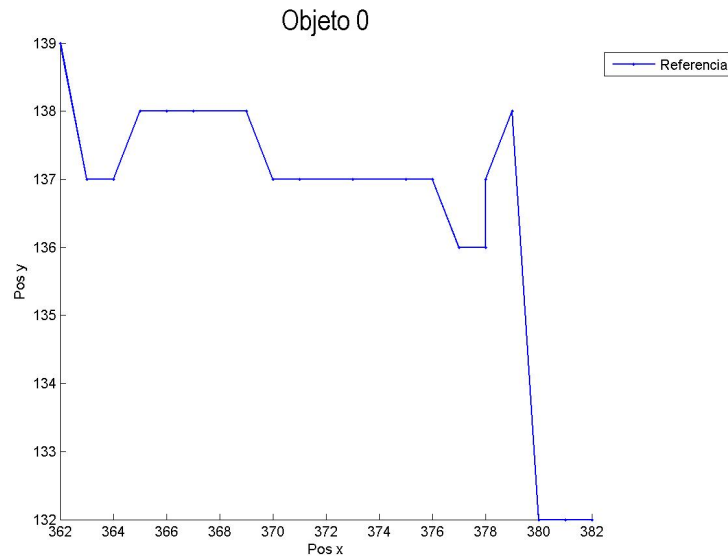


Figura 29: Trayectoria de referencia de objeto 0.

El siguiente objeto tiene un ID 1 y se muestra en la figura 27, aunque es muy difícil de identificar ya que está muy próximo al objeto anterior y su color es el mismo. Al igual que el objeto anterior, éste tiene una presencia corta pues aparece en la primera imagen y se pierde en la imagen número 26. Su trayectoria de referencia es pequeña pero continua.



Figura 30: Objeto 1 de secuencia de imágenes número 3.

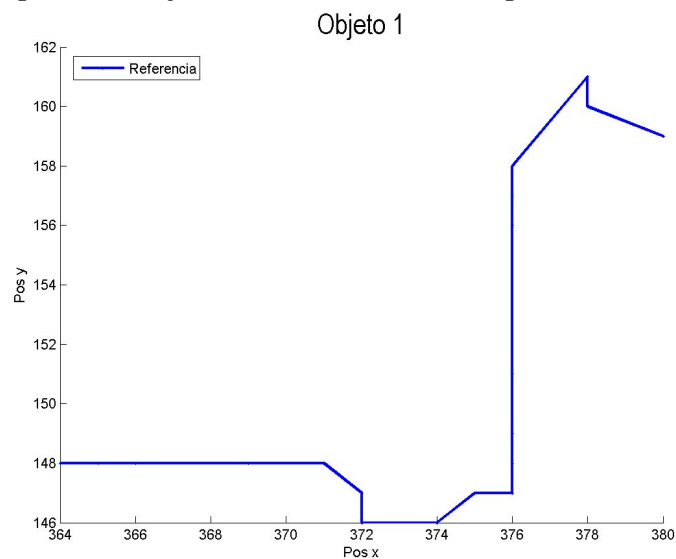


Figura 31: Trayectoria de referencia del objeto 1.

El objeto con ID 2 aparece en la imagen 94, solo se observa una parte de él en el interior del centro comercial, se observa en la siguiente imagen. Debido a lo anterior, su trayectoria de referencia (fig. 32) muestra cambios bruscos en el eje y.



Figura 32: Objeto 2 de la secuencia de imágenes número 3.

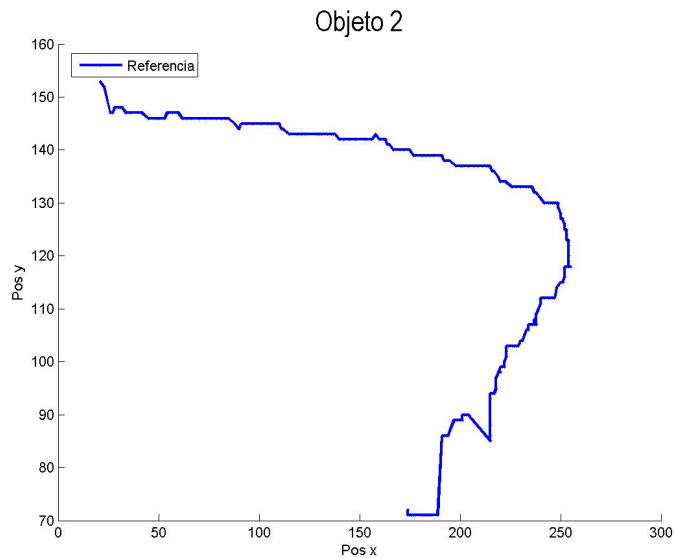


Figura 33: Trayectoria de referencia del objeto 2.

En la imagen número 130 se observa el objeto número 4 (con ID 3) de la secuencia, que identifica a una persona caminando a lo largo del corredor, y desaparece en la última imagen. Su trayectoria de referencia tiene un intervalo de presencia mayor a los objetos anteriores, es uniforme y continua.



Figura 34: Objeto 3 de la secuencia de imágenes número 3.

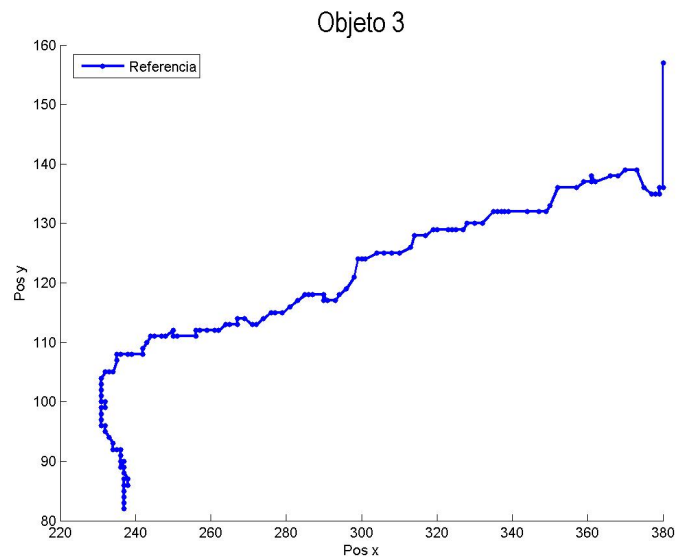


Figura 35: Trayectoria de referencia del objeto 3.

Al realizar la evaluación de los algoritmos de seguimiento antes mencionados con las métricas se obtuvieron los siguientes resultados:

Liga entre imágenes

Se detectó que los algoritmos son intermitentes, es decir, que detectan objetos en una imagen y en la siguiente no, por lo tanto los resultados de esta métrica para todas las secuencias son ceros:

GT	FT	MT1	MT2
0	0	0	1604

Tabla 1.Liga entre imágenes sec. 1 algoritmos 1 y 3.

GT	FT	MT1	MT2
0	0	0	431

Tabla 2.Liga entre imágenes sec. 2 algoritmos 1, 2 y 3.

GT	FT	MT1	MT2
0	0	0	584

Tabla 3.Liga entre imágenes sec. 3 algoritmos 1, 2 y 3.

Como se observa el único dato que se obtiene es el seguimiento perdido de tipo 2 (MT2), debido a que el algoritmo no detectó ninguna de las ligas existentes en la trayectoria de referencia; así que no hubo ninguna buena detección (GT) y tampoco existen falsos seguimientos debido a que no hay ninguna liga entre imágenes detectada. Así los cálculos que dependen de estos datos (OT, RT1, RT2, Precisión, Sensitividad 1, Sensitividad 2, etc.) también se convierten en cero.

Número de objetos detectados en un instante de tiempo

GT	FT	MT	OT	RT	Precisión	Sensitividad	F-Contador
364	82	1193	446	1557	0.816144	0.233783	0.363455

Tabla 4. Resultados de métrica 2.3.2 con secuencia 1 algoritmo 1.

Como se observa en los resultados, la precisión de este algoritmo es de 82 %, esto se debe a que de un total de 446 (OT) objetos detectados por el algoritmo se asociaron 364 de manera correcta (GT). Sin embargo la sensitividad es pequeña, debido a que de los 1557 objetos (RT) que se tienen en la trayectoria de referencia el algoritmo detectó únicamente 364.

En la gráfica se observa con mayor claridad que es en pequeños intervalos donde el algoritmo detecta todos los objetos presentes por imagen. Se graficó el error obtenido de la diferencia entre los objetos de la trayectoria de referencia y los objetos detectados por el algoritmo en cada imagen.

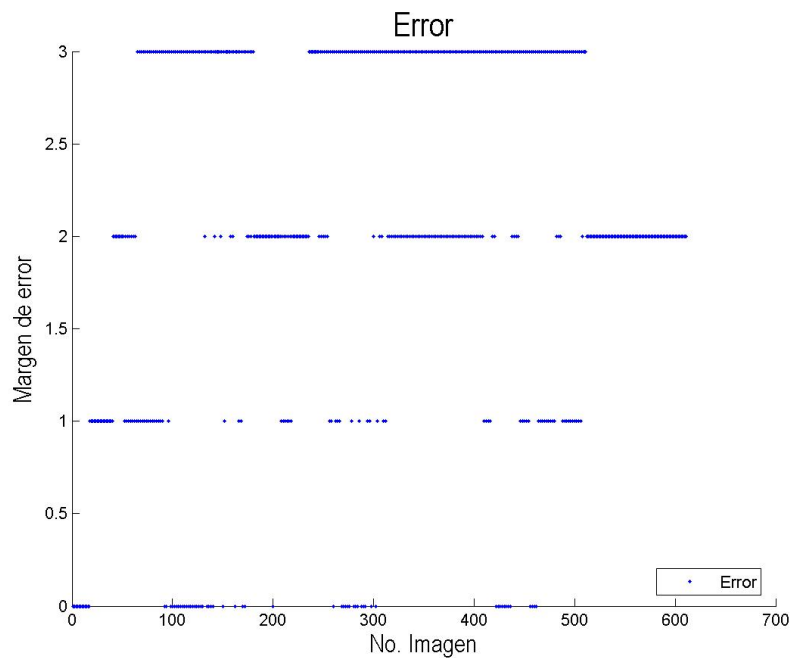


Figura 36: Secuencia 1 comparada con resultados de métrica 2.3.2.

GT	FT	MT	OT	RT	Precisión	Sensitividad	F-contador
333	32	1224	365	1557	0.912329	0.213873	0.346514

Tabla 5. Resultados de métrica 2.3.2 con secuencia 1 algoritmo 3.

En este caso se observa que la precisión aumentó un 9% respecto al algoritmo 1, esto se debe a que el número de objetos que detectó el algoritmo disminuyó, mientras que los seguimientos correctos (GT) disminuyeron, por lo tanto la sensibilidad disminuyó.

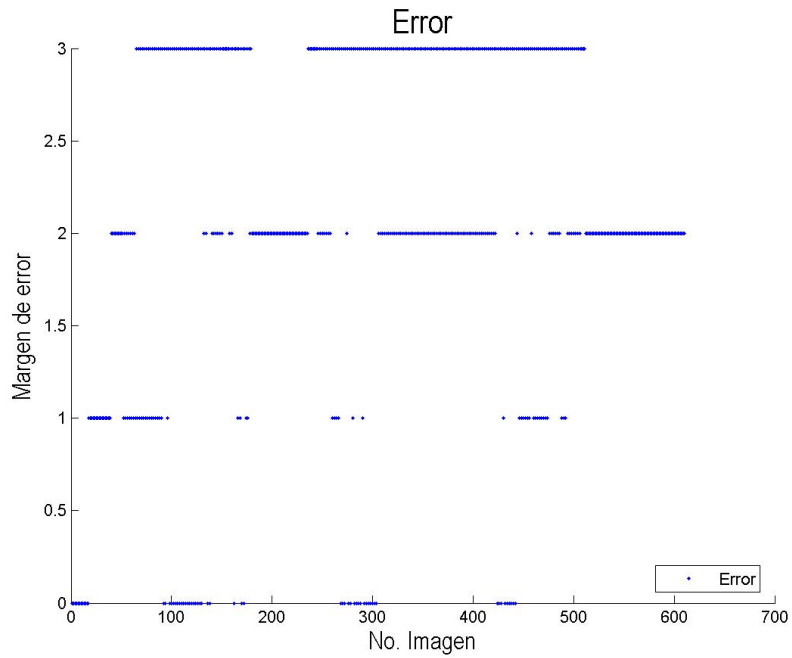


Figura 37: Secuencia 1 comparada con resultados de métrica 2.3.2.

GT	FT	MT	OT	RT	Precisión	Sensitividad	F-contador
177	62	312	239	489	0.740586	0.361963	0.486264

Tabla 6. Resultados de métrica 2.3.2 con secuencia 2 algoritmo 1.

En este caso la sensibilidad es de 31 %, pues al observar los datos se nota que se detectaron 177 objetos de un total de 489 objetos.

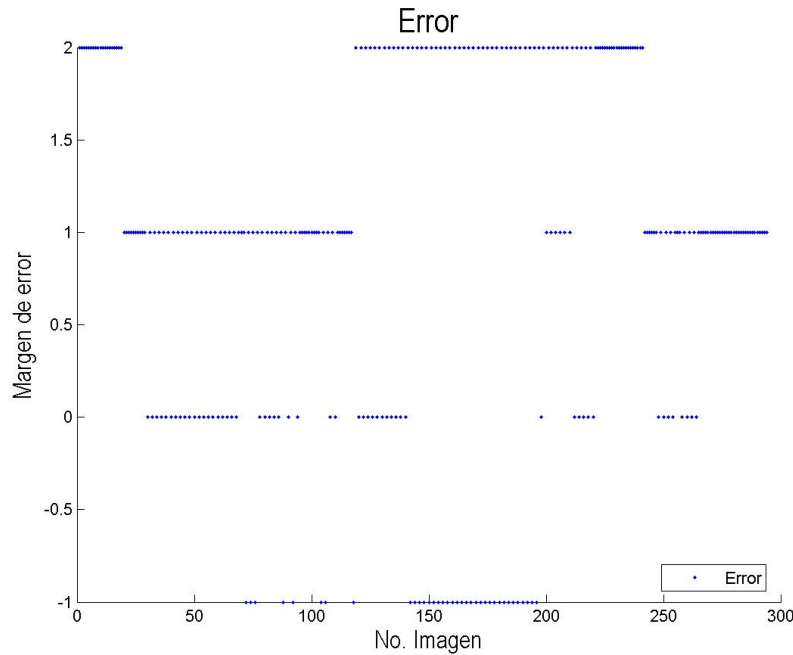


Figura 38: Secuencia 2 comparada con resultados de métrica 2.3.2.

Al ver la gráfica se puede notar que existen imágenes en las que el algoritmo detecta más objetos de los existentes en la trayectoria de referencia y éstos se toman como seguimientos correctos, la razón de esto es el umbral que se tomó para asociar a los objetos. Existen falsos seguimientos (FT) que pueden ser tomados como correctos si es que no exceden el umbral fijado. Para aumentar la certeza de esta métrica únicamente hay que disminuir el umbral*.

* Se puede consultar mayor información acerca del umbral definido en el apéndice B.

GT	FT	MT	OT	RT	Precisión	Sensitividad	F-contador
64	50	373	114	437	0.561404	0.146453	0.232305

Tabla 7. Resultados de métrica 2.3.2 con secuencia 2 algoritmo 2.

En este caso la sensibilidad es muy pequeña pues únicamente se detectaron correctamente 64 objetos y existen 50 objetos que se detectaron falsamente. Esto se observa mejor en la gráfica, ya que el grado de error es elevado.

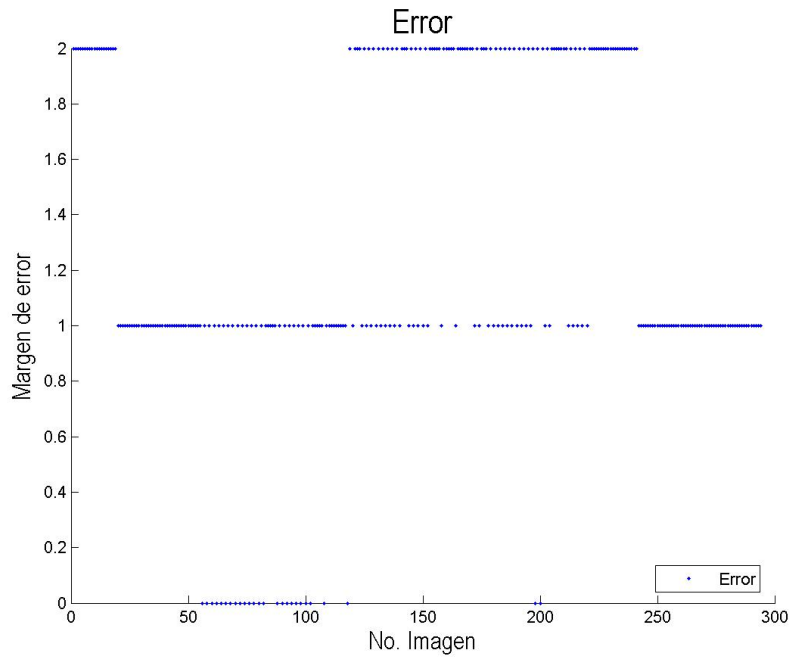


Figura 39: Secuencia 2 comparada con resultados de métrica 2.3.2.

GT	FT	MT	OT	RT	Precisión	Sensitividad	F-contador
150	65	343	215	493	0.697674	0.304260	0.423729

Tabla 8. Resultados de métrica 2.3.2 con secuencia 2 algoritmo 3.

En estos resultados se tiene el mismo problema que en los del algoritmo 1, pues se toman algunos objetos que no existen en la trayectoria de referencia. Sin embargo la persistencia aumentó un 13 % debido a que los seguimientos correctos aumentaron.

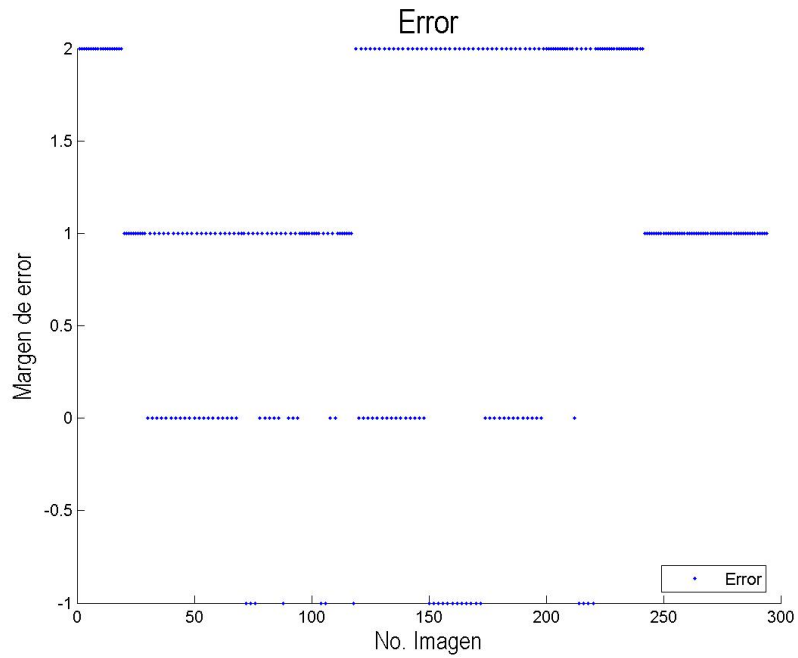


Figura 40: Secuencia 2 comparada con resultados de métrica 2.3.2.

GT	FT	MT	OT	RT	Precisión	Sensitividad	F-contador
159	264	391	423	550	0.375887	0.289091	0.326824

Tabla 9. Resultados de métrica 2.3.2 con secuencia 3 algoritmo 1.

Lo que se desea obtener es una estabilidad entre la precisión y la sensibilidad, la cual está dada por el F-Contador entre más se acerque a 1, habrá un mejor desempeño del algoritmo. En este caso sólo se tiene un 32%.

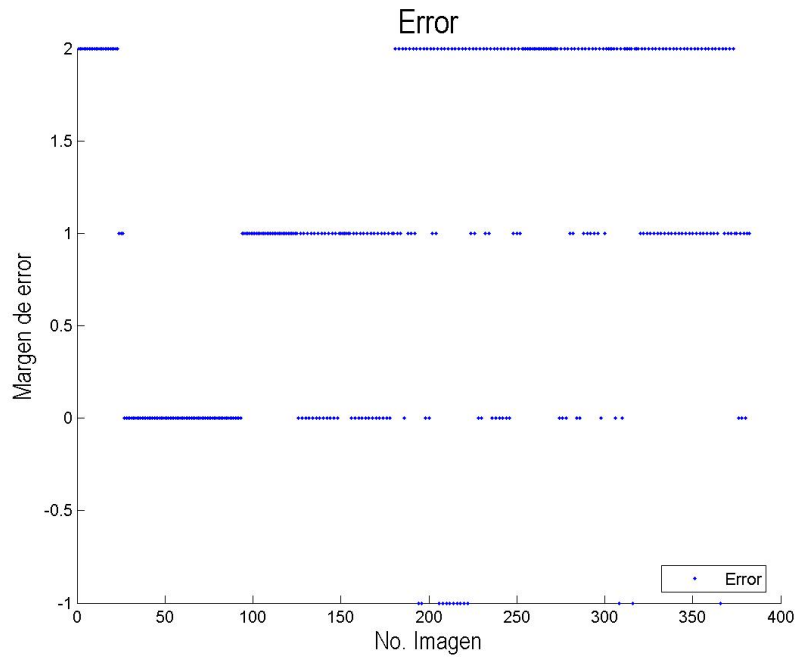


Figura 41: Secuencia 3 comparada con resultados de la métrica 2.3.2.

GT	FT	MT	OT	RT	Precisión	Sensitividad	F-contador
159	264	391	423	550	0.375887	0.289091	0.326824

Tabla 10. Resultados totales de métrica 2.3.2 con secuencia 3 algoritmo 2.

Se obtuvieron los mismos resultados que con el algoritmo anterior.

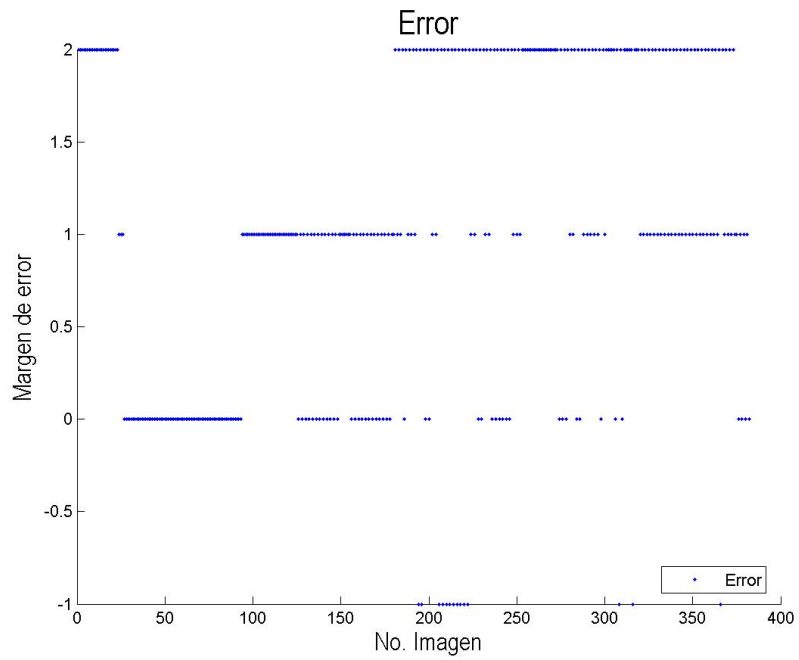


Figura 42: Secuencia 3 comparada con resultados de métrica 2.3.2.

GT	FT	MT	OT	RT	Precisión	Sensitividad	F-contador
159	210	377	369	536	0.430894	0.296642	0.351381

Tabla 11. Resultados totales de métrica 2.3.2 con secuencia 3 algoritmo 3.

Al tomar en cuenta el desempeño general de este algoritmo y compararlo con los dos anteriores, se tiene que el algoritmo 3 resultó mejor para esta secuencia y al observar la gráfica es fácil notar que existe un menor grado de error que en los anteriores.

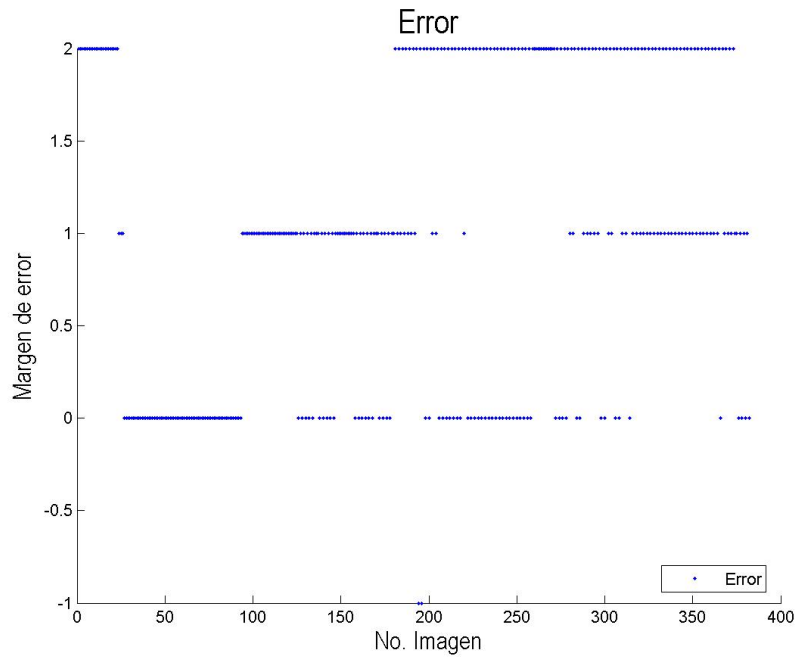


Figura 43: Secuencia 3 comparada con resultados de la métrica 2.3.2.

Evaluación en tiempo de seguimiento.

Para esta métrica se evalúa la siguiente fórmula:

$$T_{seguido} = \frac{1}{NB_{objRef}} \sum_{objref} \frac{card(RD \cap C)}{card(RD)}$$

Donde:

- NB_{objRef} es el número de objetos en la trayectoria de referencia.
- $\frac{card(RD \cap C)}{card(RD)}$ es el traslape obtenido entre el objeto en la trayectoria de referencia y el objeto detectado por el algoritmo.

Así se obtienen los siguientes datos:

Objeto	Traslape	Cardinalidad total	Sumatoria	T-seguido
0	57137	1207301	0.047326	0.000763
4	19170	5010129	0.003826	0.000023
1	113355	25941564	0.004370	0.000017
Totales	189662	252028	0.752543	0.188136

Tabla 12. Resultados de métrica 2.3.3 con secuencia 1 algoritmo 1.

Donde “Objeto” es el objeto en la trayectoria de referencia, “Traslape” es el traslape existente entre el objeto en la trayectoria de referencia y el objeto detectado por el algoritmo en el intervalo de presencia del objeto detectado. La “Cardinalidad total” es la suma de las cardinalidades obtenidas en el intervalo de presencia, la “Sumatoria” es el resultado de la división entre la “Cardinalidad total” y la cardinalidad obtenida en el intervalo de presencia del objeto en la trayectoria de referencia.

En algunas tablas resultantes de esta métrica, se puede observar que hay identificadores de objetos repetidos, esto se debe a que el algoritmo perdió en algún momento al objeto en seguimiento, y lo volvió a detectar posteriormente, por lo tanto se tiene más de un intervalo de presencia en los resultados del algoritmo para un mismo objeto en la trayectoria de referencia.

Objeto	Traslape	Cardinalidad total	Sumatoria	T-seguido
0	58690.750	1203708	0.048758	0.000786
4	16590.50	4536643	0.003657	0.000027
1	113144.50	26282564	0.004305	0.000017
5	7668.50	2588698	0.002962	0.000064
Totales	196094.250	254059	0.771845	0.192961

Tabla 13. Resultados de métrica 2.3.3 con secuencia 1 algoritmo 3.

Objeto	Traslape	Cardinalidad total	Sumatoria	T-seguido
1	67377	3656405	0.018427	0.000091
2	5277	3038430	0.001737	0.000026
Totales	72654	90144	0.805977	0.268659

Tabla 14.Resultados de métrica 2.3.3 con secuencia 2 algoritmo 1.

Objeto	Traslape	Cardinalidad total	Sumatoria	T-seguido
1	56530.250	2631057	0.021486	0.000125
2	1260	296524	0.004249	0.000708
Totales	57790.250	74779	0.772814	0.257605

Tabla 15.Resultados de métrica 2.3.3 con secuencia 2 algoritmo 2.

Objeto	Traslape	Cardinalidad total	Sumatoria	T-seguido
1	66427.50	3468198	0.019153	0.000097
2	1699	960690	0.001769	0.000080
Totales	68126.50	81174	0.839265	0.279755

Tabla 16.Resultados e métrica 2.3.3 con secuencia 2 algoritmo 3.

Objeto	Traslape	Cardinalidad total	Sumatoria	T-seguido
2	30556.750	1087644	0.028094	0.000281
2	36016.50	3884447	0.009272	0.000087
3	36854	4026128	0.009154	0.000131
3	32266	8037628	0.004014	0.000038
Totales	135693.250	184401	0.735860	0.183965

Tabla 17.Resultados de métrica 2.3.3 con secuencia 3 algoritmo 1.

Objeto	Traslape	Cardinalidad total	Sumatoria	T-seguido
2	30556.750	1087644	0.028094	0.000281
2	36016.50	3884447	0.009272	0.000087
3	36854	4026128	0.009154	0.000131
3	32266	8037628	0.004014	0.000038
Totales	135693.250	184401	0.735860	0.183965

Tabla 18.Resultados de métrica 2.3.3 con secuencia 3 algoritmo 2.

Objeto	Traslape	Cardinalidad total	Sumatoria	T-seguido
2	3084.50	25863	0.119263	0.005963
2	21853.750	841532	0.025969	0.000351
2	33396.750	3134375	0.010655	0.000102
3	36298.250	6032770	0.006017	0.000056
Totales	94633.250	132497	0.714229	0.178557

Tabla 19.Resultados totales de métrica 2.3.3 con secuencia 3 algoritmo 3.

Fragmentación ID

En esta métrica se obtienen los siguientes resultados:

Objeto de GT	ID asociado
0	3
4	3
5	0
1	1
Persistencia:	0.035714

Tabla 20.Resultados métrica 2.3.4 secuencia 1 Algoritmo 1.

Donde Objeto de GT es el identificador del objeto en la trayectoria de referencia, “ID asociado” es el número de identificadores diferentes en los resultados del algoritmo asociados al mismo objeto en la trayectoria de referencia. Así la persistencia se obtiene de la siguiente manera:

$$Persistencia = \frac{1}{NB_{objref}} \sum_{objref} \frac{1}{NumObjetoID_{/objref}}$$

Donde:

- NB_{objref} es el número objetos en la trayectoria de referencia.
- $NumObjetoID_{/objref}$ es el número de identificadores diferentes en los resultados del algoritmo asociados al mismo objeto en la trayectoria de referencia.

Objeto de GT	ID asociado
0	3
4	3
5	1
1	1
Persistencia:	0.031250

Tabla 21.Resultados métrica 2.3.4 secuencia 1 Algoritmo 3.

Objeto de GT	ID asociado
0	0
1	1
2	3
Persistencia:	0.083333

Tabla 22.Resultados métrica 2.3.4 secuencia 2 Algoritmo 1.

Objeto de GT	ID asociado
0	0
1	2
2	1
Persistencia:	0.111111

Tabla 23.Resultados métrica 2.3.4 secuencia 2 Algoritmo 2.

Objeto de GT	ID asociado
0	0
1	1
2	1
Persistencia:	0.166667

Tabla 24.Resultados métrica 2.3.4 secuencia 2 Algoritmo 3.

Objeto de GT	ID asociado
0	0
1	0
2	2
3	2
Persistencia:	0.062500

Tabla 25.Resultados métrica 2.3.4 secuencia 3 Algoritmo 1.

Objeto de GT	ID asociado
0	0
1	0
2	2
3	2
Persistencia:	0.062500

Tabla 26.Resultados métrica 2.3.4 secuencia 3 Algoritmo 2.

Objeto de GT	ID asociado
0	0
1	0
2	3
3	1
Persistencia:	0.062500

Tabla 27.Resultados métrica 2.3.4 secuencia 3 Algoritmo 3.

Criterio de confusión para IDs

En el caso de esta métrica todos los algoritmos probados con cualquier secuencia de imágenes, presentan un porcentaje de confusión de 0 %.

Trayectorias 2D.

GT	FT	MT	OT	RT	Precisión	Sensitividad	F-contador
330	212	2508	542	2838	0.608856	0.116279	0.195266

Tabla 28. Resultados de métrica 2.3.6 sec. 1 algoritmo 1.

En base a los resultados obtenidos por esta métrica se obtiene la siguiente gráfica que representa el error obtenido entre la trayectoria de referencia y los resultados obtenidos por el algoritmo.

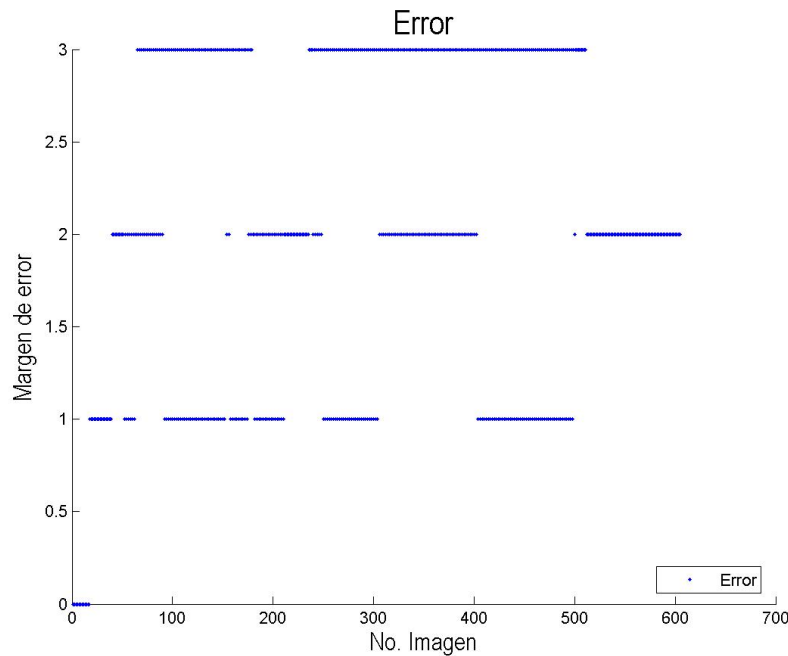


Figura 44: Resultados de algoritmo 1 comparados con secuencia 1.

Como se observa claramente, no existe ningún momento en que el algoritmo detecte por completo los objetos en la trayectoria de referencia, es por ello que el número de seguimientos perdidos (MT) es muy elevado, lo que indica que el algoritmo no es confiable. El dato OT muestra el número total de puntos detectados por el algoritmo en la trayectoria.

GT	FT	MT	OT	RT	Precisión	Sensitividad	F-contador
292	127	2742	419	3034	0.696897	0.096243	0.169128

Tabla 29. Resultados de métrica 2.3.6 con secuencia 1 algoritmo 3.

La gráfica que se muestra a continuación muestra más claramente el desempeño de este algoritmo.

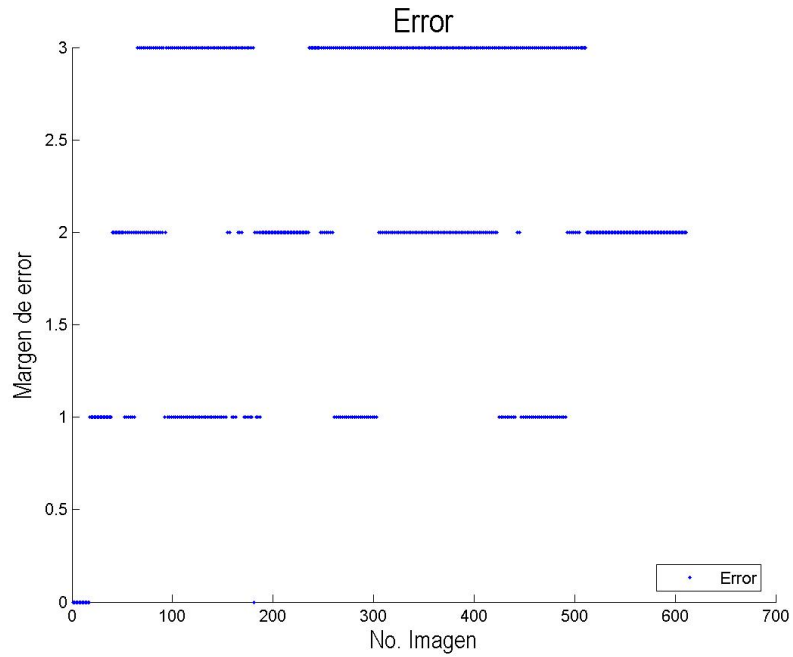


Figura 45: Resultados de algoritmo 3 comparados con secuencia 1.

Como se observa fácilmente, este algoritmo tiene un número mayor de seguimientos perdidos (MT), por lo tanto el algoritmo 1 resulta mejor.

La precisión aumenta un 8% con respecto al algoritmo anterior, pero la sensibilidad disminuye.

GT	FT	MT	OT	RT	Precisión	Sensitividad	F-contador
141	185	372	326	513	0.432515	0.274854	0.336114

Tabla 30. Resultados de métrica 2.3.6 con secuencia 2 algoritmo 1.

Al observar la gráfica generada por los resultados de este algoritmo evaluado con la secuencia número 2, queda claro que en este caso existe una mayor exactitud por el algoritmo que en los casos anteriores, es por ello que la sensibilidad aumenta a un 27%, ya que esta indica la relación que hay entre los objetos asociados (algoritmo, trayectoria de referencia) y los datos de la trayectoria de referencia.

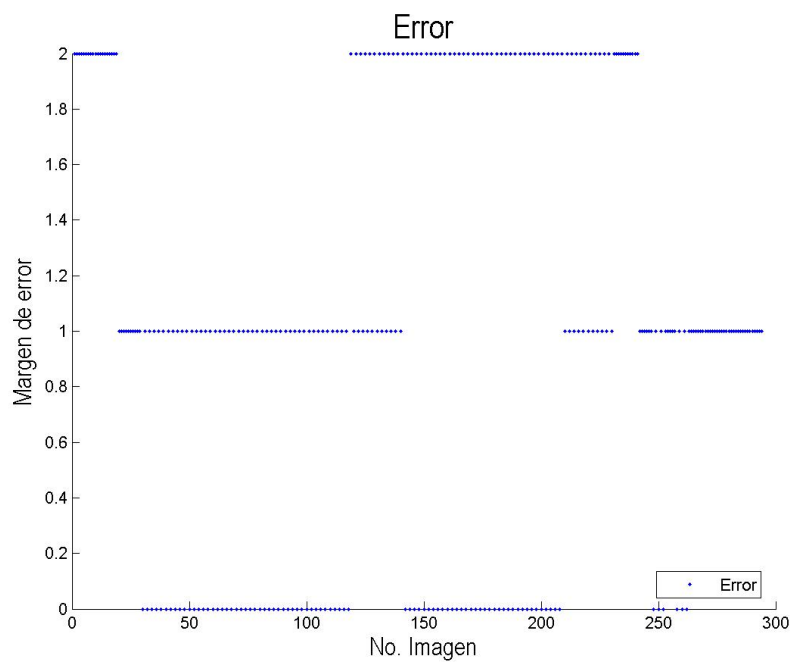


Figura 46: Resultados de algoritmo 1 comparados con secuencia 2.

GT	FT	MT	OT	RT	Precisión	Sensitividad	F-contador
96	28	433	124	529	0.774194	0.181474	0.294028

Tabla 31. Resultados de métrica 2.3.6 con secuencia 2 algoritmo 2.

Para el caso de este algoritmo se observa con facilidad en la tabla de resultados y en la gráfica que el número de GT aumenta considerablemente, lo que hace al algoritmo 1 más confiable.

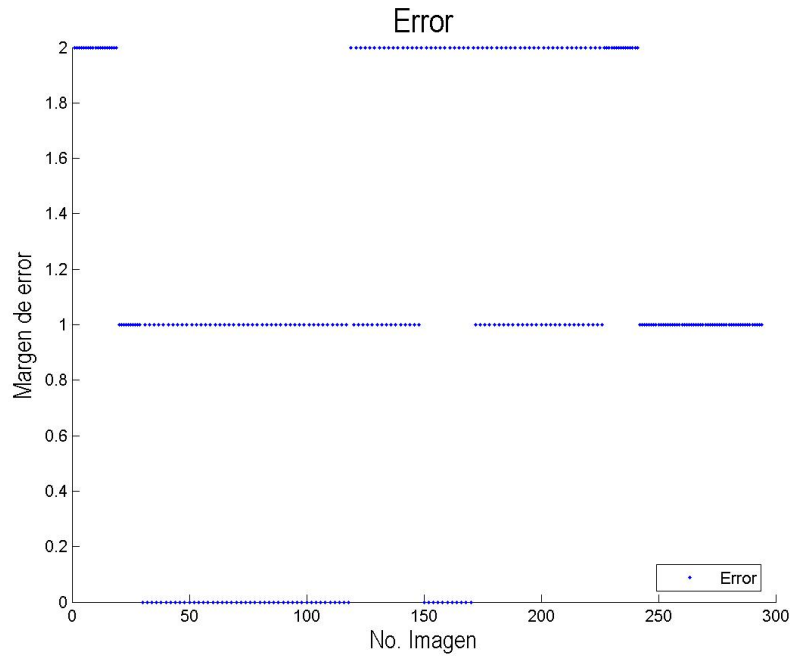


Figura 47: Resultados de algoritmo 2 comparados con secuencia 2.

GT	FT	MT	OT	RT	Precisión	Sensitividad	F-contador
110	204	409	314	519	0.350318	0.211946	0.264106

Tabla 32. Resultados de métrica 2.3.6 con secuencia 2 algoritmo 3.

En el caso de este algoritmo, el número de seguimientos perdidos disminuye con respecto al algoritmo anterior, pero no es mejor que el resultado del algoritmo 1, debido a que existe un número mayor de seguimientos correctos.

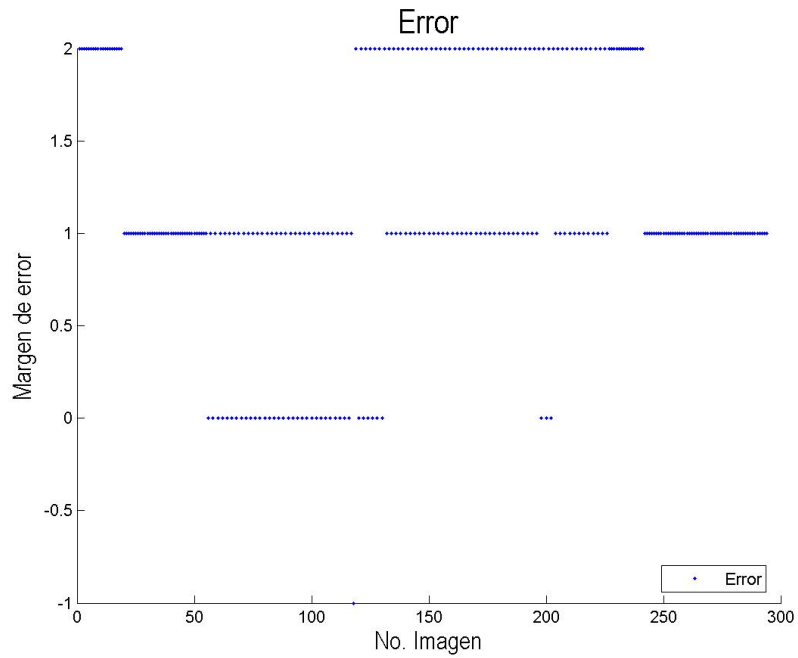


Figura 48: Resultados de algoritmo 3 comparados con secuencia 2.

GT	FT	MT	OT	RT	Precisión	Sensitividad	F-contador
198	437	516	635	714	0.311811	0.277311	0.293551

Tabla 33. Resultados de métrica 2.3.6 con secuencia 3 algoritmo 1.

Al observar la gráfica de resultados de este algoritmo probado con la secuencia 3, queda claro que únicamente algunos puntos corresponden con la trayectoria de referencia y esto se nota más claramente al comparar los datos, pues de un total de 714 objetos en la trayectoria de referencia (RT), el algoritmo solo detectó 198 (GT), lo cual implica que perdió 516 objetos. Además se observa que el número de falsos seguimientos (FT) es muy elevado.

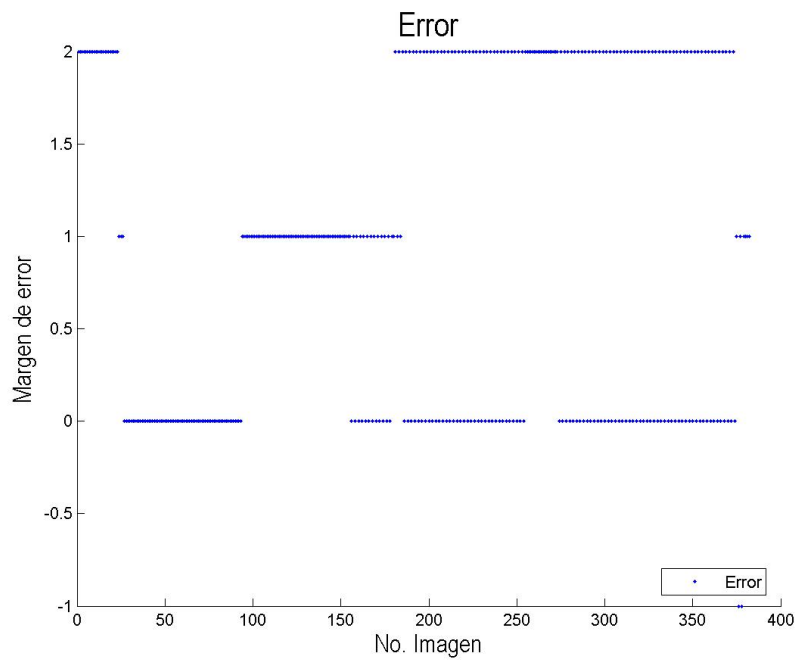


Figura 49: Resultados de algoritmo 1 comparados con secuencia 3.

GT	FT	MT	OT	RT	Precisión	Sensitividad	F-contador
198	437	516	635	714	0.311811	0.277311	0.293551

Tabla 34. Resultados de métrica 2.3.6 con secuencia 3 algoritmo 2.

En base a la tabla de resultados y la gráfica, se deduce que este algoritmo y el algoritmo 2 presentan el mismo desempeño.

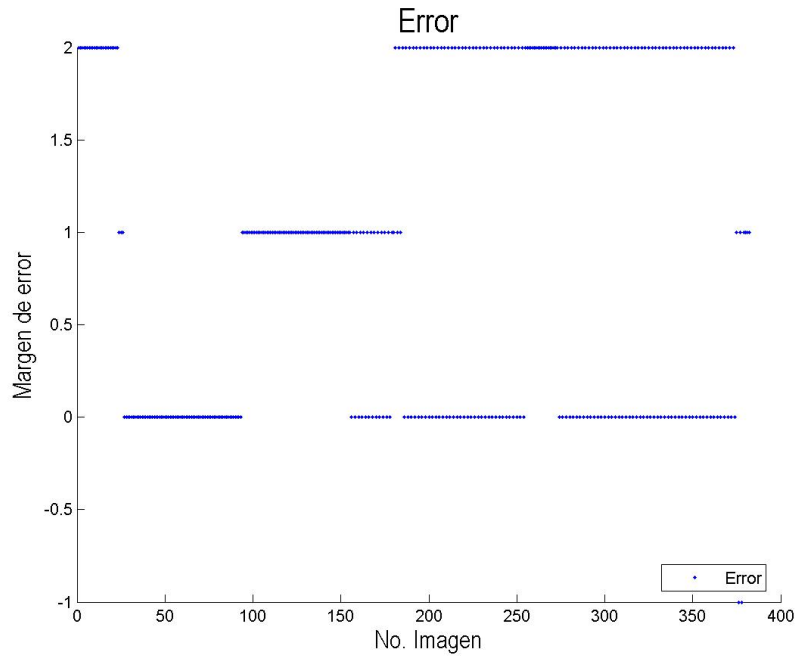


Figura 50: Resultados de algoritmo 2 comparados con secuencia 3.

GT	FT	MT	OT	RT	Precisión	Sensitividad	F-contador
153	415	584	568	737	0.269366	0.207598	0.234483

Tabla 35. Resultados de métrica 2.3.6 con secuencia 3 algoritmo 3.

Se puede notar que el desempeño de este algoritmo disminuye, pues el número de seguimientos correctos (GT) disminuye a 153, pero por otro lado los falsos seguimientos también disminuyen.

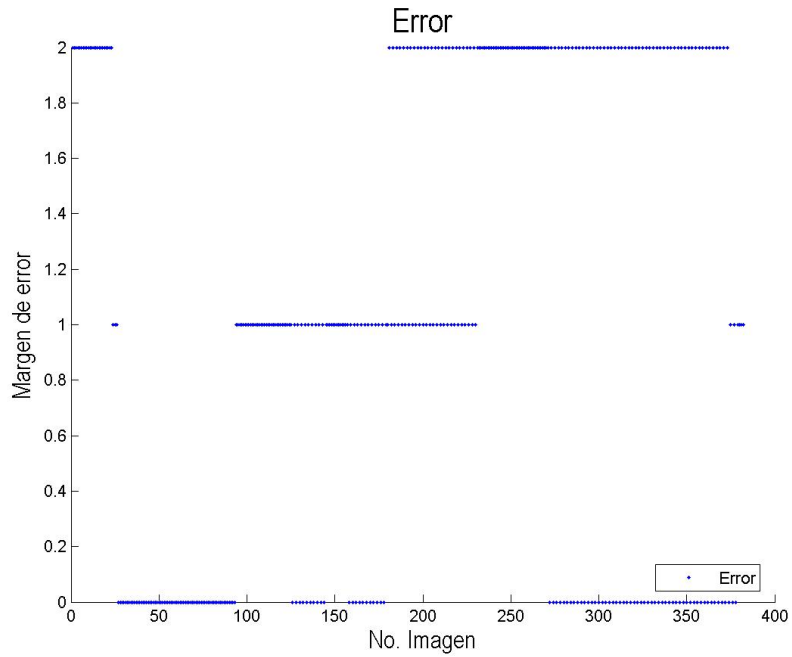


Figura 51: Resultados de algoritmo 3 comparados con secuencia 3.

5. Conclusiones.

Se desarrolló un sistema que evalúa algoritmos de seguimiento de objetos en movimiento que cuenta con 6 diferentes formas de medir el desempeño cuantitativo de los mismos, cada una de las medidas debe seleccionarse de acuerdo a las características que se deseen evaluar en el algoritmo.

En base a los resultados obtenidos en esta tesis, se puede concluir que una de las métricas que resulta más útil es la de “Ligado entre dos imágenes” que indica si el algoritmo realiza el seguimiento entre imágenes de manera correcta y constante, para el caso de los algoritmos evaluados se detectó que ambos son intermitentes, es decir, que detectan objetos en una imagen y en la siguiente no, por lo que su trayectoria no es continua. Además la métrica “Trayectorias 2D” proporciona información importante en cuanto al seguimiento, ya que la precisión indica la relación que existe entre los objetos asociados y los detectados por el algoritmo; la sensibilidad muestra la relación existente entre los objetos asociados y los objetos en la trayectoria de referencia. Por lo tanto entre mayor sea la sensibilidad habrá una mayor correspondencia con la trayectoria de referencia, y a su vez la precisión también debe ser lo más cercana posible a uno, ya que esto indicaría que hubo un número muy pequeño de falsos seguimientos.

Todos estos datos también son proporcionados por la métrica “Número de objetos detectados en un instante de tiempo”. El desempeño general de cada algoritmo se sabe al verificar el dato F-Contador, que indica la relación que existe entre la precisión y la sensibilidad, por lo tanto, entre más se acerque este dato a uno, mayor es el desempeño del algoritmo en cuanto al seguimiento de los objetos.

Además de las tres métricas mencionadas anteriormente, también existen otras que sirven para evaluar cosas específicas de cada algoritmo, como el manejo que se da a cada identificador asociado a un objeto en el algoritmo (mediante la métrica criterio de confusión de IDs). También se puede evaluar el tiempo en que se mantuvo constante un identificador (con la métrica evaluación en tiempo de seguimiento), o saber cuál es la persistencia de los objetos (métrica fragmentación ID), la selección de las métricas va a depender de las características del mismo que se deseen evaluar. Cabe señalar que los algoritmos fueron diseñados para seguir objetos rígidos, sin embargo se evaluaron con secuencias de imágenes que presentan objetos no rígidos.

Basándose en la información proporcionada en los trabajos relacionados vistos en la sección 2.6, podemos concluir que existen métricas muy similares entre sí, cuyo objetivo es medir el desempeño de los algoritmos de seguimiento de objetos en movimiento, pero difieren en la manera en que comparan los datos, en los artículos mencionados, solo se obtuvo una métrica idéntica a la de Trayectorias 2D propuesta por ETISEO, que es llamada “Error de Seguimiento de Objeto” en [4], esta métrica utiliza la distancia euclidiana para comparar los datos de la trayectoria de referencia y los del algoritmo. Sin embargo esto sucede únicamente con la métrica “Trayectorias 2D”, las demás métricas de ETISEO son completamente diferentes, ya que, aunque su objetivo es comparar las trayectorias punto por punto, varía la manera de comparar los datos, por ejemplo, en la sección 2.6.2 se usan métodos estadísticos y matemáticos, que son utilizados por las métricas propuestas por ETISEO pero se valen también de otro tipo de información

como son los identificadores de cada objeto y sus intervalos de presencia.

5.1. Limitaciones y Trabajos Futuros.

Una de las limitantes más significativas del sistema es que el número de objetos soportado en una imagen es pequeño, debido a la estructura de datos definida. Es decir, hay ocasiones en las que el algoritmo detecta más de 7 objetos en una imagen y el sistema tiene capacidad solamente para 7. Este problema puede resolverse a futuro si los datos de cada objeto son almacenados en una base de datos, así se tendría capacidad para soportar tantos objetos como detecte el algoritmo.

Para el caso de la métrica “Criterio de confusión de IDs” también se propone una mejora, que consiste en verificar las posibles confusiones empleando distancias entre objetos, es decir, medir la distancia que hay entre los objetos de una misma imagen y así saber cuando existe un posible cambio de identificadores (cuando dos objetos estén muy cerca, sus trayectorias podrían cruzarse y haber confusión). Esta medida ayudaría mucho al buen desempeño de la métrica.

El programa que realiza la construcción del archivo con formato XML tiene la capacidad de recibir información más detallada por parte del algoritmo, como son:

- **Clasificación de objetos.-** Una persona, un grupo de personas (contiene de 2 a 7 personas), una multitud (más de 7 personas), un objeto portable, un auto o múltiples autos.
- **Reconocimiento y clasificación de eventos.-** Por ejemplo: persona 3 corriendo, persona 2 caminando, etc.

En base a la mejora de los algoritmos para que proporcionen este tipo de datos, se pueden proponer nuevos métodos de programación para cada métrica, e incluso se pueden proponer nuevas métricas que se basen en esta información.

6. Apéndices

Los siguientes apéndices contienen los manuales necesarios para que el sistema sea operado de manera correcta, y para que su codificación sea comprendida fácilmente.

6.1. Apéndice A: Manual de usuario.

Para que el sistema trabaje adecuadamente, se debe contar con los siguientes requerimientos mínimos:

- Procesador Intel Celeron 902 MHz.
- 120 Mb de RAM.
- Sistema operativo Windows XP.
- Halcon 7.0.
- Secuencias de imágenes en C:\ Base de Imagenes
- Una carpeta llamada Reportes en la unidad C.

Al ejecutar el programa podrá observarse una ventana de diálogo dividida en tres partes:

- **Manejo de secuencias de imágenes.**
 1. Ventana de despliegue de imágenes: Ventana que permite observar las secuencias de imágenes después de seleccionarse.
 2. Barra combo para selección de secuencia de imágenes: Barra que permite seleccionar las secuencias de imágenes disponibles.
 3. Botón “Comenzar”: Al presionar este botón se desplegará la secuencia de imágenes seleccionada en la ventana de despliegue.
 4. Botón “Detener”: Cuando se quiere detener la secuencia de imágenes en alguna en específico se presiona este botón.
 5. Botón “Avanzar”: Este botón se usa para reanudar la secuencia de imágenes que ha sido detenida.
 6. Botón “Abrir Video”: Este botón permite abrir un video disponible de cada secuencia.
- **Corrida de algoritmo y Generación de formato XML.**
 1. Área de selección de algoritmo: Esta área muestra los algoritmos disponibles para evaluar, se permite seleccionar únicamente uno de ellos.

2. Botón “Ejecutar Algoritmo”: Al presionar este botón verá el algoritmo seleccionado ejecutándose con la secuencia de imágenes seleccionada. Los resultados se dejan en una archivo denominado Reporte.txt
3. Botón “Generar XML”: Este botón permite crear una archivo con extensión xml a partir de los datos arrojados por el algoritmo.

■ **Selección de métricas y evaluación.**

1. Área de selección de tipo de métrica: Esta área muestra el tipo de métrica disponible para evaluar el algoritmo (segmentación y seguimiento).
2. Botón “Elegir Tipo”: Al presionar este botón se podrá observar una lista detallada de las métricas disponibles dependiendo del tipo seleccionado. Estas métricas pueden seleccionarse o deshabilitarse de acuerdo al interés de evaluación.
3. Botón “Evaluar”: Este botón toma los resultados obtenidos por el algoritmo en formato xml y los evalúa, genera un archivo de texto por cada métrica seleccionada (cada uno con el nombre de la métrica), en cada archivo se muestra el resultado de la evaluación de dicha métrica.

El cuadro de diálogo del sistema también cuenta con dos menús en su parte superior:

1. Archivo: Este menú solamente cuenta con la opción para salir del sistema.
2. ?: En este menú se encuentra la “Ayuda de las Métricas”, que despliega un recuadro que muestra los pasos a seguir para la evaluación de un algoritmo.

6.2. Apéndice B: Manual Técnico.

El sistema está hecho con el lenguaje Visual C++ versión 6.0, además se creó una interfaz con Halcon 7.0 para la lectura de las imágenes; consta de un programa principal y cuatro programas independientes que realizan tareas específicas. La comunicación entre el programa principal y los programas componentes se realiza mediante el llamado a los procesos ejecutables de cada uno como sea conveniente.

1. Programa Principal.

Dentro del programa principal se define un cuadro de combo (combo box) al que se agregan los nombres de las secuencias con la instrucción `AddString`, para ello se creó un arreglo de estructuras de datos denominado `secuencia`, la cual consta de los siguientes elementos:

1. `CString sec`: Cadena en la que se almacena la ruta de la secuencia de imágenes `Walking Secuencia3 Walk3`
2. `int n`: Entero que indica el número de imágenes con que cuenta la secuencia.
3. `int cero`: Entero que indica el número de ceros que contiene el nombre de la imagen.
4. `CString gt`: Cadena que indica la ruta y nombre de la trayectoria de referencia.

Dentro del método `OnCloseupCombo()`, se verifica qué secuencia fue seleccionada y con los datos almacenados en la estructura se crea un ciclo que lee las imágenes desde Halcon y las despliega en la ventana. Debido a que Halcon crea su propia variable de control y el cuadro de combo también necesita una, es necesaria la creación de un hilo que maneje la lectura de imágenes. Así cuando se da click en el botón `Comenzar`, se prende la bandera de inicio (`iniciar`) y se corre el hilo, que se detiene apagando la bandera cuando se da click en el botón `Detener`, el proceso del botón `Avanzar` es similar al del botón `Comenzar`, pero en este caso se inician los datos con los que se tenían al `Detener` la secuencia. El botón `Abrir Video` * simplemente hace un llamado al proceso que permite abrir un video con la siguiente instrucción:

```
WinExec("PlayWnd", SW_NORMAL);
```

donde `PlayWnd` indica el nombre del programa, y `SWNORMAL` indica que se desplegará en una ventana normal de visual C++.

Hasta este punto se han definido los elementos necesarios para la lectura de las secuencias de imágenes. Posteriormente, en el método correspondiente al botón `Ejecutar Algoritmo`, se crean tres variables tipo `CButton`, que permiten verificar si se ha seleccionado alguno de los botones de radio disponibles para cada algoritmo, y dependiendo del que se halla seleccionado, se llama al proceso que ejecuta el algoritmo correspondiente, veamos un ejemplo:

* Si se desea modificar este programa se tiene que acudir al módulo correspondiente.

```
WinExec("FusionDecision ruta,nombre,ceros,numImg", SW_NORMAL);
```

Como en la instrucción anterior, el nombre del programa es FusionDecision *, pero ahora va acompañado por los parámetros que necesita conocer como son la ruta de la secuencia (ruta), el nombre de la secuencia (nombre), el número de ceros que preceden el nombre de cada imagen (ceros) y el número de imágenes que conforman la secuencia (numImg). La lectura de estos parámetros se hace en el programa FusionDecision mediante apuntadores.

En el botón “Generar XML” se realiza un proceso similar para identificar qué algoritmo ha sido seleccionado y así llamar al proceso que realiza esta tarea:

```
WinExec("PetsMetricsXMLWriteData ruta", SW_NORMAL);
```

Como se observa el nombre de este programa es PetsMetricsXMLWriteData * y recibo como parámetro únicamente la ruta donde se encuentra el archivo txt generado por el algoritmo.

En la parte de selección de métricas y evaluación, cuando se da click sobre el botón “Elegir Métrica”, se checa qué tipo de métrica se ha seleccionado y de acuerdo a ello se despliega uno de dos cuadros de diálogo, cada uno muestra el nombre de las métricas disponibles. Una vez seleccionadas las métricas deseadas, se verifica cuáles de ellas han sido elegidas y de acuerdo a ello se crea un archivo de texto denominado m.txt que contiene los números correspondientes a las métricas seleccionadas:

- 1.- 1 ⇒ métrica 2.3.1 Ligado entre dos imágenes.
- 2.- 2 ⇒ métrica 2.3.2 Número de objetos detectados en un instante de tiempo.
- 3.- 3 ⇒ métrica 2.3.3. Evaluación en tiempo de seguimiento.
- 4.- 4 ⇒ métrica 2.3.4. Fragmentación ID.
- 5.- 5 ⇒ métrica 2.3.5. Criterio de confusión para IDs.
- 6.- 6 ⇒ métrica 2.3.6. Trayectorias 2D.

Finalmente cuando se da click sobre el botón “Evaluar”, se llama al proceso que contiene las métricas programadas con la siguiente instrucción:

```
WinExec("XmlMenuDemo GT,rutaRep", SW_NORMAL);
```

Como se observa, el nombre del programa es XmlMenuDemo * y recibe dos parámetros, la ruta de la trayectoria de referencia y la ruta del reporte generado por el algoritmo en formato xml.

2. Algoritmos de seguimiento.

Estos programas fueron diseñados en Visual C++ con interfaz a Halcon 7.0 por la Doctora Janeth Cruz [7], so programa original toma el resultado de tres algoritmos y los fusiona para obtener un mejor resultado en el seguimiento, en este caso se separaron los tres algoritmos, y se modificaron para que leyeran los parámetros necesarios en su ejecución:

* Si se desea modificar este programa se tiene que acudir al módulo correspondiente.

```
CWinApp* pApp2=AfxGetApp ();    // Apuntador del cuadro de diálogo
n=pApp2->m_lpCmdLine;           // CString que contiene los parámetros
```

De acuerdo a lo anterior, se tienen los parámetros en la cadena n, que simplemente tiene que dividirse en cadenas separadas por comas y así obtener los parámetros. Esto se realiza mediante uno o varios ciclos como el siguiente:

```
Para i=0 hasta tam hacer
  si n[j]!=','
    concatena dato
  sino
    guarda indice
    rompe ciclo
  fin sino
fin si
fin para
```

donde tam es el tamaño de la cadena que contiene los parámetros.

Cada uno de estos algoritmos crea un archivo de texto con el formato especificado en la página número 39, y lo nombra dependiendo del algoritmo que se haya ejecutado, por ejemplo, si se ejecutó el algoritmo 1, el reporte se llamará Reporte0.txt, si se ejecuta el algoritmo 2, el reporte se llamará Reporte1.txt y si se ejecuta el algoritmo 3, su reporte será llamado Reporte2.txt.

2. Generación de archivo XML.

Como ya se explicó anteriormente, este programa recibe como parámetro el nombre del reporte generado en el módulo anterior, después de obtener el nombre, hace una lectura del archivo generado y toma sus datos:

```
1  Lee(Reporte.txt)
2  c = Lee primera línea;
3  int seguido=0;
4  mientras c != NULL
5      int id=0;
6      si !band
7          band=1;
8          ptr1= toma primer token de línea
9          mientras divide_token!= NULL
10             fr=convierte_a_entero(ptr1)
11             fin mientras
12         fin si
13         sino
```

```
14     mientras divide_token!= NULL
15         fr=convierte_a_entero(ptr1)
16     fin mientras
17 fin sino
18 c = Lee linea
19 c = Lee linea
20 ptr = Toma primer token de linea
21 cambio = ptr
22 mientras cambio=="Imagen:" y c!=NULL
23     id = convierte_a_entero(ptr)
24     ptr = Toma siguiente token
25     yc= convierte_a_entero(ptr)
26     ptr = Toma siguiente token
27     xc= convierte_a_entero(ptr)
28     ptr = Toma siguiente token
29     xmin= convierte_a_entero(ptr)
30     ptr = Toma siguiente token
31     ymin= convierte_a_entero(ptr)
32     ptr = Toma siguiente token
33     xmax= convierte_a_entero(ptr)
34     ptr = Toma siguiente token
35     ymax = convierte_a_entero(ptr)
36     ptr = Toma siguiente token
37     Asinga_info2d(xmin,xmax,xc,ymin,ymax,yc)
38     c = Toma siguiente línea
39     ptr = Toma primer token de linea
40     cambio = ptr
41     ptr1=ptr
42 fin mientras
43     Asigna_info(info2d,fr)
44 fin mientras
```

Para dividir las líneas obtenidas en la lectura (líneas 8, 20, 24, 26, 28, 30, 32, 34, 36 y 39), se usa la instrucción de Visual C++ `strtok(cadena, "separador")`, que separa la cadena recibida por tokens de acuerdo al separador indicado. En las líneas 37 y 43, se hace referencia a métodos de asignación de información, éstos se refieren a los métodos diseñados para la creación del archivo XML. Se creó una clase llamada `PetsMetricsXMLObject`, que contiene varios métodos que ayudan a la construcción del archivo, entre ellos se encuentra el método `PetsMetricsXmlInfo2d`, que recibe los siguientes parámetros:

```
PetsMetricsXmlInfo2d(int xmin, int xmax, float x_center, int ymin, int ymax,
                    float y_center, int camara_id)
```

Este método es el encargado de asignar los datos obtenidos por cada objeto en la imagen, y

una vez que se han obtenido todos los datos de los objetos en una imagen, se llama al siguiente método:

```
PetsMetricsXmlFrame (targets,fr)
```

Este método recibe un vector de objetos tipo `PetsMetricsXmlTrackedTarget`, el cual contiene la información 2D de cada objeto en la imagen, y el número de imagen en el que se encuentra. Se usan vectores debido a que no se sabe con certeza cuántos objetos se tiene por imagen ni cuántas imágenes se tiene por secuencia; los vectores se crean de acuerdo a las necesidades del sistema. El resultado final de este programa es un reporte que contiene los mismos datos arrojados por el algoritmo pero en formato XML. El archivo resultante es llamado `Reporte.xml`.

2. Evaluación.

Es en este programa que se realiza todo el trabajo referente a las métricas, recibe como parámetros la ruta del archivo xml que tiene los datos de la trayectoria de referencia y la ruta del reporte xml generado por el módulo anterior. La primera actividad que se realiza en este programa es la lectura y almacenamiento de los datos de cada archivo xml, para ello se crearon dos estructuras de datos de la siguiente manera:

```
estructura objetos inicio:
    entero id
    flotante h,w,xc,yc,xmin,ymin,xmax,ymax,umbralh,umbralw
    entero g,nt
fin objeto
```

```
estructura imagenes inicio:
    entero no,imagen
    objetos obj[7]
fin coo[1010],coo1[1010]
```

Como se observa la estructura de imagen es capaz de almacenar los datos de 7 objetos, es decir, se puede tener un máximo de 7 objetos por imagen, así se crean dos estructuras, una almacena la información contenida en la trayectoria de referencia y la otra la información generada por el algoritmo de seguimiento. Después de almacenar los datos, se procede a la lectura del archivo `m.txt` generado en el programa principal, y dependiendo de los datos obtenidos se llama al método o métodos correspondientes.

1. Para la primera métrica se creó un método llamado `LigaEntreFrames()`, dentro del cual se crea un ciclo que recorre toda la estructura de la trayectoria de referencia y al mismo tiempo obtiene los datos correspondientes arrojados por el algoritmo. El proceso seguido se comprende mejor al observar la siguiente imagen:

La figura tiene marcado con número rojos los 3 procesos que deben seguirse, primero se

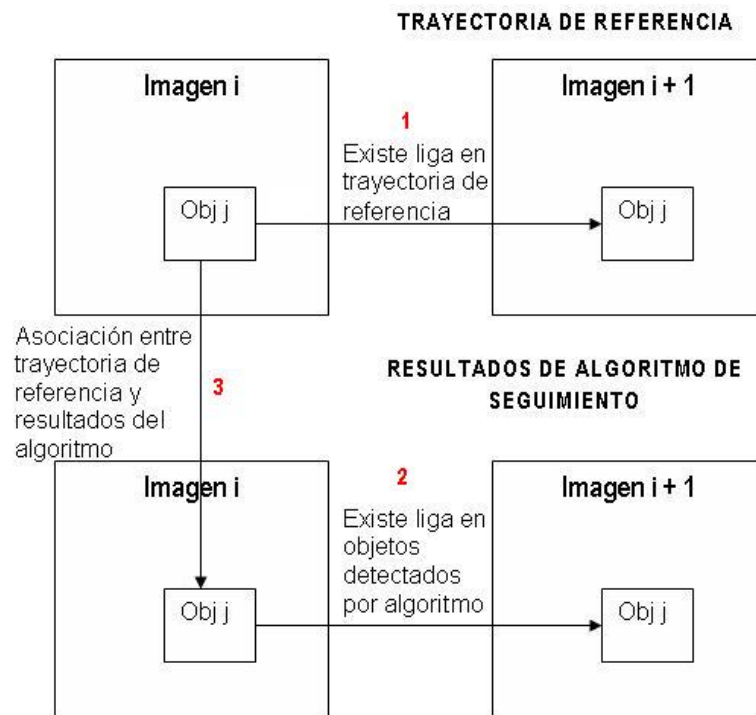


Figura 52: Proceso general para ligado entre imágenes.

debe verificar que exista una liga entre un objeto de la imagen i y otro de la imagen $i+1$ en la trayectoria de referencia, si esto sucede, se verificará que exista también una liga entre un objeto en la imagen i con otro de la imagen $i+1$ en los datos del algoritmo y si esto también se cumple, ahora se verificará que exista una asociación entre los objetos de la trayectoria de referencia y los objetos detectados por el algoritmo.

Así el pseudocódigo se describe a continuación:

```

1  para i=0 hasta fin_referencia hacer:
2      obj2 = toma_datos_de_referencia(i)
3      obj3 = toma_datos_de_referencia(i+1)
4      card = D2(xm3,xm2,ym3,ym2,xx3,xx2,yx3,yx2,h3,h2,w3,w2);
5      card1 = w2*h2
6      si card > card1/8
7          ligasGT = ligasGT+1
8          para j=0 hasta num_obj_referencia hacer:
9              obj1=toma_datos_obj(j)
10             card=D2(xm1,xm2,ym1,ym2,xx1,xx2,yx1,yx2,h1,h2,
11                    w1,w2)
11             card1=h2*w2;
12             si card >= card1/8
13                 para m=0 hasta num_obj_alg hacer:
14                     obj=toma_datos_obj(m)

```

```

15         card=D2(xm1,xm,ym1,ym,xx1,xx,yx1,yx,h1,h0,
                w1,w0)
16         card1=w0*h3
17         si card >= card1/8
18             card=D2(xm3,xm,ym3,ym,xx3,xx,yx3,yx,h3,
                    h0,w3,w0)
29             card1=h3*w3
20             si card >= card1/4
21                 GT = GT+1
22                 asociadas = asociadas+1
23                 GTT = GTT+1
24             fin si
25         fin si
26     fin para
27     fin si
28     fin para
29     fin si
30 fin para

```

Como se observa, en este caso se fijó un umbral igual a un octavo de la cardinalidad del área mínima que contiene al objeto en la trayectoria de referencia (líneas 6,13 y 18) para la liga entre una imagen y otra, y para verificar la asociación entre ligas (liga en trayectoria de referencia y liga en resultados de algoritmo) se usó un umbral equivalente a un cuarto de la cardinalidad del área mínima que contiene al objeto en la trayectoria de referencia. Esto es porque debe haber un traslape mayor al asociar objetos entre referencia y algoritmo para tener mayor exactitud. Esta métrica arroja resultados imagen por imagen y totales, es por ello que se manejan dos variables, GT es para mostrar los resultados obtenidos por imagen y GTT es para mostrar los resultados obtenidos al término de la secuencia. La manera de obtener los datos para los falsos seguimientos (FT) y seguimientos perdidos (MT) es similar a la de los seguimientos correctos (GT). Después de tener estos datos, los demás pueden calcularse en términos de éstos:

```

RT1 = GT + MT1
RT2 = GT + MT2
OT = GT + FT
si OTT != 0
    P = GT/OT
si RT1 != 0
    S1 = GT/RT1
si RT2 !=0
    S2 = GT/RT2
si (P+S1) != 0
    F1 = (2*P*S1)/(P+S1)
si (P+S2) !=0

```

$$F1=(2*P*S2)/(P+S2)$$

Los resultados de esta métrica son arrojados en un archivo de texto llamado “Liga entre frames.txt”.

2. Para la métrica número 2 (Número de objetos detectados en un instante de tiempo), se creó un método llamado NumeroObjetos(), cuyo objetivo es verificar la presencia de los objetos por intervalos de presencia sin tomar en cuenta su identificador. A continuación se muestra el algoritmo en pseudocódigo:

```

inicio
1  para i=0; i<fin_referencia hacer
2      f=imagen_algoritmo
3      si f != -1
4          para j=0 j<fin_referencia hacer:
5              obj2 = Toma_datos_de_referencia(i,j)
6              band=0
7              para k=0 k<fin_algoritmo hacer:
8                  obj1=toma_datos_de_algoritmo(f,k)
9                  card=D2(xm1,xm2,ym1,ym2,xx1,xx2,yx1,yx2,h1,h2,w1,w2)
10                 card1=h2*w2
11                 si card >= card1/2
12                     ban=1
13                     asociados = asociados+1
14                     GT = GT+1
15                     GT1 = GT1+1
16                 fin si
17             fin para
18             si ban!=1
19                 MT = MT+1
20                 MT1 = MT1+1
21             fin si
22         fin para
23         FT = num_obj_alg - asociados
24         asociados=0;
25         FT1+=FT;
26         OT=GT+FT;
27         OT1+=OT;
28         RT=GT+MT;
29         RT1+=RT;
30         GT=0;
31         MT=0;
32         FT=0;
33     fin si
34     sino

```

```

35         MT1+=num_obj_ref
36     fin sino
37 fin para
38 ban=0
39 OT1=GT1+FT1
40 RT1=GT1+MT1
41 if(OT1!=0) P=GT1/OT1
42 if(RT!=0) S=GT1/RT1
43 if((P+S)!=0) FS=(2*P*S)/(P+S)
fin

```

En la línea 5 se hace referencia a obtener las coordenadas del objeto j en la imagen i de la trayectoria de referencia, ocurre de manera similar en la línea 8 pero en los datos del algoritmo. El umbral establecido en esta ocasión para asociar objetos es la mitad de la cardinalidad total del cuadro de referencia del objeto en la trayectoria de referencia (línea 11), si esta condición se cumple, entonces se cuentan el número de asociaciones que se realizan en la misma imagen, para facilitar posteriormente el conteo de los falsos seguimientos (FT).

En la línea 18 se verifica si la bandera de asociación no ha sido prendida, ya que esto significaría que el objeto en la trayectoria de referencia no fue asociado a ninguno en el algoritmo, por lo tanto se trata de un seguimiento perdido (MT). La línea 23 muestra cómo obtener el número de falsos seguimientos (FT), simplemente se toma el número de objetos existentes en la imagen del algoritmo y se le resta el número de objetos que fueron asociados. El resto de los datos dependen de estos.

En este caso los datos se pueden observar en el archivo “Numero de objetos.txt”.

3. La tercera métrica (Evaluación en tiempo de seguimiento), necesitó de la elaboración de un método llamado `ComparaIDs()`, que a su vez hace uso de `ChecaAsociacion(indimg,indobj)`, cuyo objetivo es devolver la tupla que se observa en la página número 46, a continuación se presenta el pseudocódigo para ello:

```

    entero ChecaAsociacion (entero ind_img,entero id_obj)
1  inicio
2  numImg=frame_algoritmo(ind_img)
3  para m=0 m<num_obj_alg(numImg) hacer:
4      si(id_obj==obj[m])
5          para i=0 i<fin_referencia hacer:
6              si imagen(i) == numImg
7                  para j=0 j<num_obj_referencia hacer
8                      obj2 = toma_datos_referencia(i,j)
9                      obj1 = toma_datos_algoritmo(ind_img,m)
10                     card=D2(xm1,xm2,ym1,ym2,xx1,xx2,yx1,yx2,h1,h2,w1,w2)
11                     card1=h2*w2
12                     si card >= (card1/2)

```

```

13             ob=obj[j]
14             terminar ciclo
15         fin si
16     fin para
17     fin si
18     fin para
19     regresa ob
20     fin si
21     fin para
22     regresa -1
fin

```

En la línea número 2 se obtiene el número de la imagen dependiendo del índice en la estructura del algoritmo, después se obtienen los datos necesarios para verificar la cardinalidad de su traslape, en este caso se fijó un umbral igual a la mitad de la cardinalidad total del área mínima que contiene al objeto en la trayectoria de referencia (líneas 11 y 12). Por último este método regresa el objeto asociado o -1 en el caso de no haber asociación. La tupla es almacenada en una matriz de la siguiente forma:

- obj[0] Almacena el identificador del objeto detectado por el algoritmo.
- obj[1] Almacena el número de imagen donde aparece por primera vez el objeto.
- obj[2] Almacena el número de imagen en que aparece por última vez el objeto.
- obj[3] Almacena la persistencia del objeto.
- obj[4] Almacena el identificador en la trayectoria de referencia al que fue asociado el objeto del algoritmo.

Este método crea dos tuplas, una para los datos del algoritmo (obj[]) y otra para almacenar los datos de la trayectoria de referencia (obj1[]), los resultados se envían al archivo denominado “Evaluacion tiempo.txt”.

4. la métrica Fragmentación de IDs, también hace uso del método anterior, y a partir de la tupla generada por dicho método simplemente cuenta los identificadores diferentes que fueron asociados a un mismo objeto de referencia:

```

inicio
1   ComparaIDs()
2   para j=0 hasta fin_tupla_ref hacer:
3       dato = obj1[j]
4       para k=0 hasta fin_tupla_alg hacer:
5           si obj_asociado(k)==dato
6               numIDs = numIDs+1
7           fin si
8       fin para
9       sumatoria = sumatoria + numIDs

```

```

10     numIDs = 0
11   fin para
12   numo = num_obj_referencia
13   sumatoria = 1/sumatoria
14   persistencia = (1/numo)*sumatoria
fin

```

En las líneas 4, 5 y 6 se calcula el número de identificadores diferentes que se asocian a un solo objeto en la trayectoria de referencia (numIDs). Al final del ciclo (línea 11), se realiza el cálculo correspondiente a la persistencia. Este método arroja sus resultados en un archivo de texto llamado “Fragmentacion ID.txt”.

5. La métrica siguiente (Criterio de Confusión), parte del método anterior, ya que es el mismo procedimiento, pero ahora se deben contar los identificadores diferentes que se asignaron a un objeto diferente en los resultados del algoritmo. Sus resultados se pueden ver en el archivo de texto generado por el método, se llama “confusion.txt”.
6. Por último para la sexta métrica (Trayectorias 2D), también hace uso del método ComparaIDs() y a partir de la tupla generada, toma el inicio y fin de cada asociación entre objetos por la imagen en la que aparecen y desaparecen dichos objetos y calcula la distancia entre ellos:

```

inicio
1   para i=0 hasta fin_tupla_referencia hacer:
2     para j=0 hasta fin_tupla_algoritmo hacer:
3       si obj1[i] == obj[j]
4         fin = obj1[2]
5         inicio = obj1[1]
6         para k=inicio hasta fin hacer:
7           para l=0 hasta num_obj_algoritmo hacer:
8             si obj[j]==toma_obj_algoritmo(inicio,l)
9               x1 = toma_xc_obj(inicio,l)
10              y1 = toma_yc_obj(inicio,l)
11            fin si
12          fin para
13          para h=0 hasta num_obj_referencia hacer:
14            si obj1[i] == toma_obj_referencia(k,h)
15              x2 = toma_xc_obj(k,h)
16              y2 = toma_yc_obj(k,h)
17            fin si
18          fin para
19          d = raiz_cuad(potencia((x2-x1),2)+potencia((y2-y1),2))
20          GT = GT+1
21        fin para
22      MT1 = MT1+1

```

```
23         fin si
24     fin para
25 fin para
26 OT = GT + FT
27 RT = GT + MT1
28 si OT!=0      P=GT/OT
29 si RT!=0      S=GT/RT
30 si P!=0 y S!=0      FS=(2*P*S)/(P+S)
fin
```

Como se explicó anteriormente, las tuplas resultantes del método `ComparaIDs()` son almacenadas en los arreglos `obj` (algoritmo) y `obj1` (trayectoria de referencia), por lo tanto en la línea 4 se está tomando el número de imagen en la que aparece por primera vez el objeto `j`, y en la línea número 5 se toma la última imagen en la que aparece el mismo objeto, de esta manera el ciclo tiene un inicio y un fin. Debe hacerse énfasis en la línea número 19, ya que la distancia calculada entre trayectorias es la distancia euclidiana. Estos datos son enviados a “Trayectorias.txt”.

7. Referencias

- [1] Domingo Mery, Visión por Computador. Departamento de Ciencia de la Computación. Universidad Católica de Chile. Santiago de Chile 17 de agosto de 2004.
- [2] Black, J. et al., A Novel Method for Video Tracking Performance Evaluation, Joint IEEE Int'l Grupo de trabajo sobre Vigilancia Visual y Evaluación del desempeño de Seguimiento de Objetos y Vigilancia (VS-PETS), Francia, Octubre 11-12, 2003, p125-132.
- [3] Tim Ellis James Black and Paul Rosin, A novel method for video tracking performance evaluation, Reunión Internacional de grupos de trabajo de la IEEE sobre video vigilancia y evaluación del desempeño del Seguimiento de objetos y Vigilancia. (VS-PETS) (2003).
- [4] Nils T Siebel, Steve Maybank. Fusion of Multiple Tracking Algorithms for Robust People Tracking. In Proceedings of the 7th European Conference on Computer Vision (ECCV 2002), IV, p. 373–387, Mayo, 2002.
- [5] D. Young y J. Ferryman. PETS Metrics: On-line Performance Evaluation Service. The Second Joint IEEE International Workshop on Visual Surveillance and Performance Evaluation of Tracking and Surveillance (VS-PETS 2005) en Beijing, China del 15 al 16 de Octubre 2005.
- [6] Nils T Siebel y Steve Maybank. Real-Time Tracking of Pedestrians and Vehicles. In 2nd IEEE Workshop on Performance Evaluation of Tracking and Surveillance (PETS'2001), Diciembre 2001.
- [7] Janeth Cruz Enriquez, Fusion de sensores adaptable y Distribuida para Detección y Seguimiento de Objetos. Tesis Doctoral. Instituto Nacional de Astrofísica, Óptica y Electrónica, 2005.
- [8] Jonathan Owens, Andrew Hunter y Eric Fletcher, A Fast Model-Free Morphology-Based Object tracking Algorithm. Universidad de Sunderland, UK, Escuela de Computación y Tecnología.
- [9] M. Xu, and T.J. Ellis, Partial Observation vs Blind Tracking through Occlusion, British Machine Vision Conference (BMVC 2002), Cardiff, Septiembre 2002, pp 777-786
- [10] K. Nummiaro, E. Koller-Meier, L. Van Gool. Color Features for Tracking Non-Rigid Objects. Special Issue on visual Surveillance, Chinese Journal of Automation, May 2003, Vol. 29, No. 3, pp 345-355

-
- [11] Héctor Barrón González, Seguimiento Robusto de Múltiples Objetos con Precisión Subpixel. Tesis de Maestría. Instituto Nacional de Astrofísica, Óptica y Electrónica, 2004.
- [12] Luis David López Gutiérrez. Fusión de sensores multiespectrales para la detección y seguimiento de múltiples objetivos. Tesis de Maestría. Instituto Nacional de Astrofísica, Óptica y Electrónica, 2004.
- [13] Iván Olivera Romero, Detección y seguimiento de objetos en imágenes infrarrojas usando información temporal. Tesis de Maestría en Ciencias Computacionales, Instituto Nacional de Astrofísica Óptica y Electrónica.
- [14] Forsyth, Ponce, Visión por Computadora, un Acercamiento moderno. 2003, pp 393.
- [15] ETISEO, Internal Technical Note Metrics Definition. Versión 2.0
- [16] Stan Z. Li. Quan Pan Tao Yang and Jing Li, Real-time multiple objects tracking with occlusion handling in dynamic scenes, IEEE Computer Vision and Pattern Recognition Conference (CVPR'05) (2005).
- [17] Chris J. Needham y Roger D. Boyle, Performance Evaluation Metrics and Statistics for positional Tracker Evaluation. School of computing, the university of Leeds, Leeds, LS2 9JT, UK.
- [18] T. Schlögl, C. Beleznai, M. Winter, H. Brischof, Performance Evaluation Metrics for Motion Detection and Tracking, Institute for Computer Graphics and Vision, University of technology, Graz, Austria.
- [19] ETISEO, Internal Technical Note, Video Annotation Rules. 2 - Approved, IN_ETI.1.009. (6 de Febrero de 2006).
- [20] ETISEO, Internal Technical Note, Data structure and output format. 2 - Approved, IN_ETI.1.011. (6 de Febrero de 2006).
- [21] Silogic - Inria. Internal Technical Note Metrics Definition,
<http://www.silogic.fr/etiseo/bibliothequeDocuments0001007b.html>