

Seguimiento y estimación de trayectorias

Tesina de grado

Facultad de Informática, UNLP

Alumnos: Juan Cortabitarte y Pablo Novarini

Director: Ing. Armando De Giusti

Codirector: Dr. Marcelo Naiouf

Agenda

- Introducción
- Caso de estudio
- Análisis de la solución
- Paralelización de la solución
- Pruebas realizadas
- Conclusiones
- Líneas de trabajo futuro

Introducción

- Motivación y objetivos generales
 - Seguimiento y estimación de trayectorias en video
 - Trayectoria: Es el conjunto de todas las posiciones por las que pasa un cuerpo en movimiento.
 - Analizar algoritmos para cada etapa
 - Tiempo real → Procesamiento paralelo
 - Ejemplos
 - Análisis de movimiento de partículas en imágenes microscópicas
 - Fútbol robot
 - Seguimiento de un automovil en una carretera

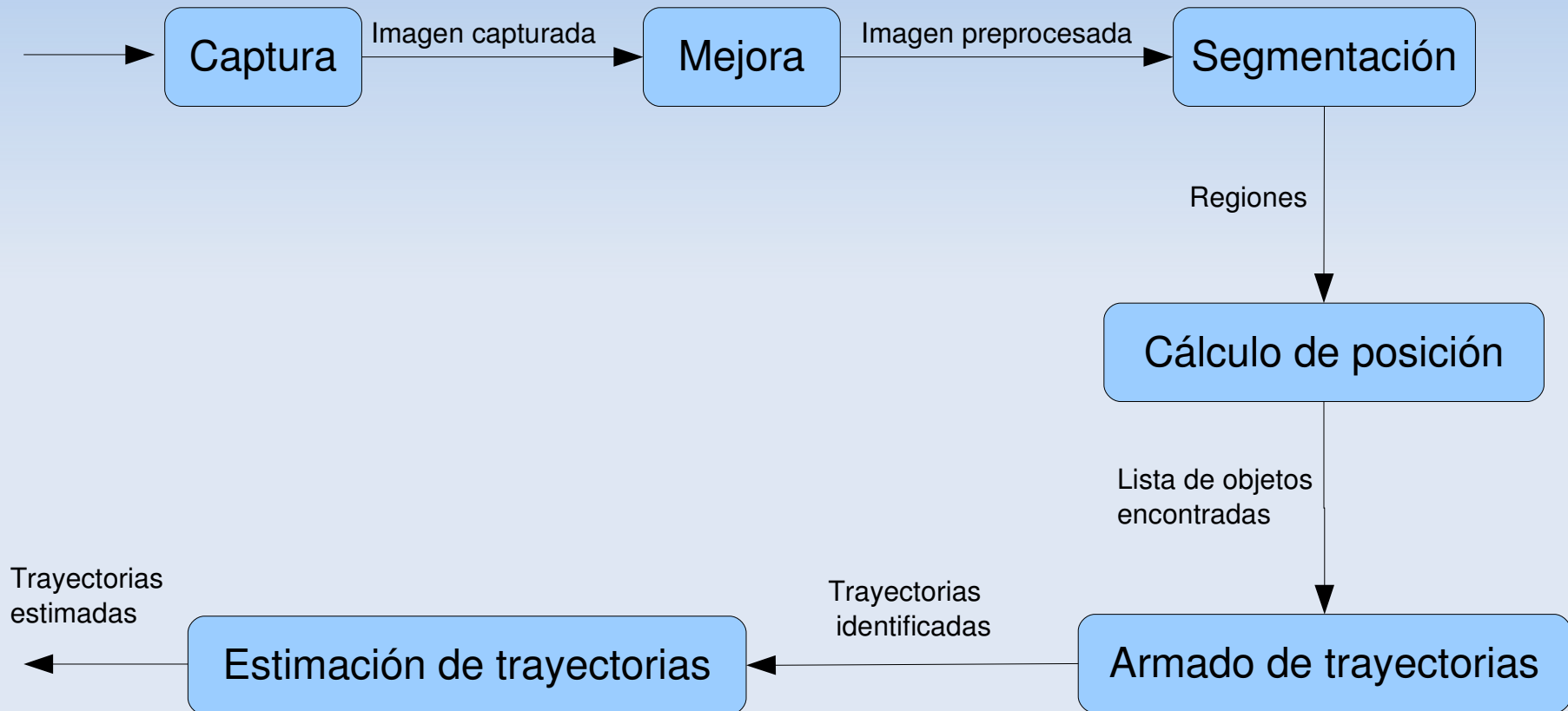
Líneas de investigación

- Procesamiento de imágenes
- Seguimiento de trayectorias
- Tiempo real
- Paralelización de algoritmos
- Arquitecturas de procesamiento paralelo

Caso de estudio

- Escena con objetos en movimiento capturada desde un plano superior
- Cámara/s en una posición fija
- Proyección 2D de una escena 3D
- Contraste entre objetos y fondo.
- Suavidad en el movimiento
- Escenas generadas artificialmente
 - Obtener escenarios controlados
 - Hacer énfasis en las etapas de interés

Fases



Captura de la imagen

- Proceso de adquirir y digitalizar señales eléctricas obtenidas a través de un sensor con un adecuado sistema óptico, y transformarlas en señales digitales.
- Distorsiones
 - Geométrica
 - Sistema de la iluminación
 - Pérdida de información
- Cantidad de adquisidores
 - Un adquisidor
 - Múltiples adquisidores

Mejora de la imagen

- Transformar una imagen de manera tal que resulte más adecuada que la original para una aplicación específica.
- Aplicación de filtros y transformaciones
 - Se corrigen distorsiones
 - Se eliminan partes indeseables de la imagen (ruido)
 - Se realzan las partes interesantes
- Caso de estudio: Se aplicó un filtro por la mediana para suavizar la imagen (se redujo el ruido causado por la transformación de avi a mpeg)

Segmentación

- Consiste en identificar los componentes significativos de la imagen
- Técnicas
 - Detección de bordes
 - Umbralización
 - Etiquetado
- Caso de estudio:
 - La umbralización nos permitió aislar los objetos del fondo
 - El etiquetado nos permitió identificar cada objeto de la escena.

Cálculo de posición

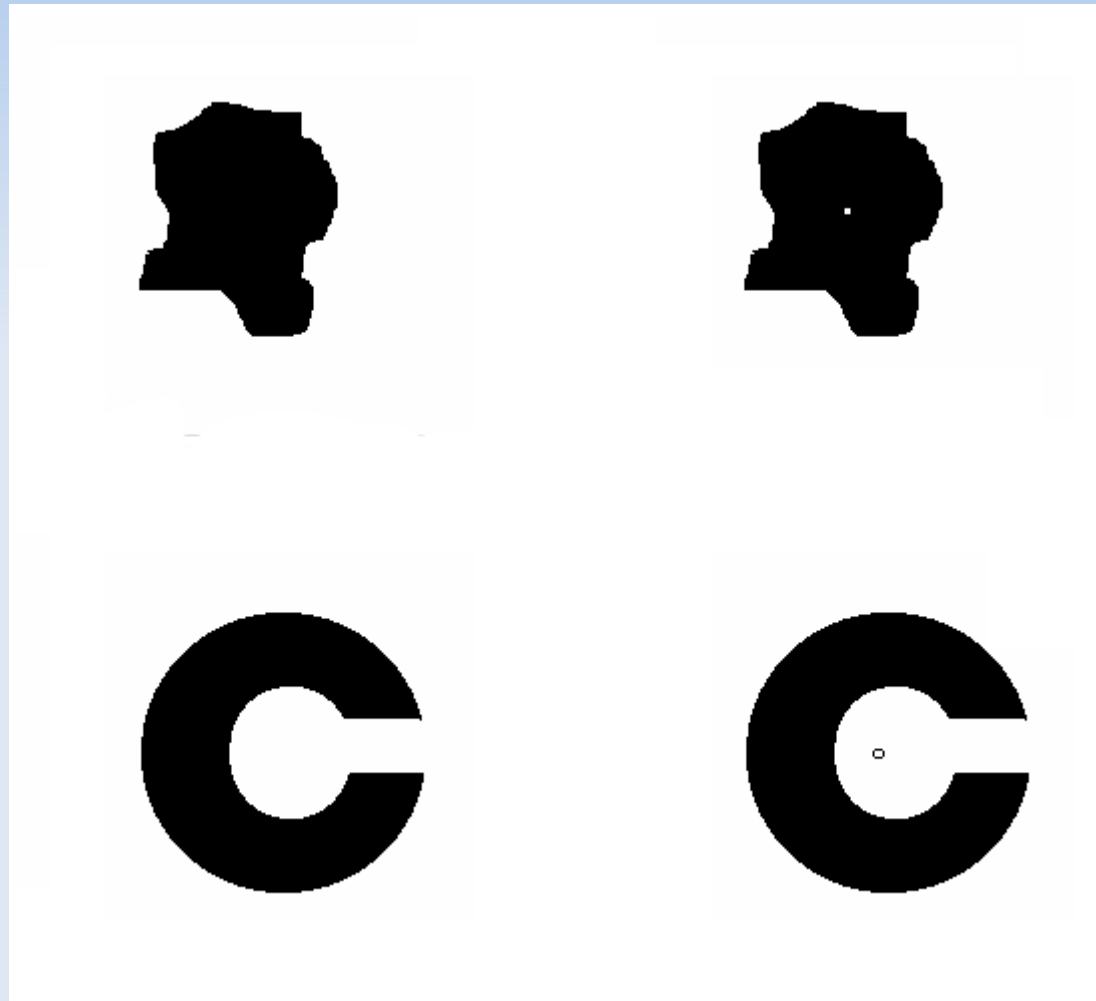
- Problema: dado un conjunto de píxeles, que representa un objeto en una imagen determinar las coordenadas del mismo.
- Requisitos
 - Sistema de coordenadas fijo a lo largo del video
 - Independencia de la rotación
 - Cálculo en función de la información segmentada

Cálculo de posición

- Centro de masa: Permite localizar un punto donde la masa del objeto se concentra.
 - Relativo al objeto
 - Independientemente de su ubicación y orientación

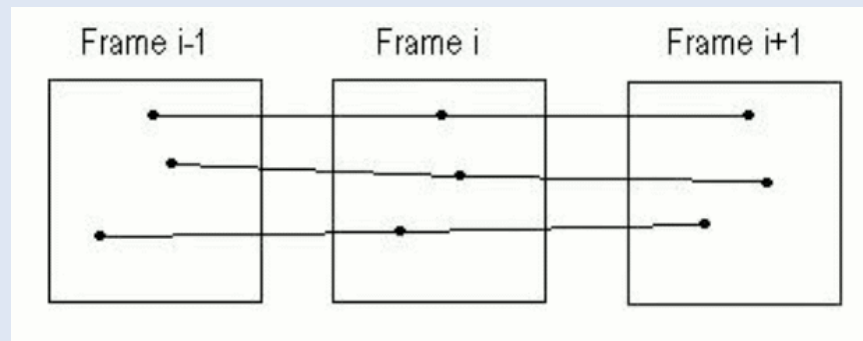
$$x_{cm} = \frac{\sum_{i=1}^n m_i x_i}{\sum_{i=1}^n m_i} = \frac{\sum_{i=1}^n m \cdot x_i}{\sum_{i=1}^n m} = \frac{m \sum_{i=1}^n x_i}{n \cdot m} = \frac{\sum_{i=1}^n x_i}{n}$$
$$y_{cm} = \frac{\sum_{i=1}^n m_i y_i}{\sum_{i=1}^n m_i} = \frac{\sum_{i=1}^n m \cdot y_i}{\sum_{i=1}^n m} = \frac{m \sum_{i=1}^n y_i}{n \cdot m} = \frac{\sum_{i=1}^n y_i}{n}$$

Cálculo de posición



Armado de trayectorias

- Identificación del recorrido realizado por cada objeto en la secuencia de imágenes analizadas
- Correspondencia : relacionar un objeto en un frame con el siguiente.
 - Por características del objeto
 - En función de la ubicación
- Caso de estudio: Objetos de similares características.



Armado de trayectorias

- Errores que pueden cometerse en la correspondencia:
 - Reconocer a dos objetos diferentes como un mismo objeto
 - No reconocer un objeto como el mismo en dos frames consecutivos
- Suposiciones acerca del movimiento de los objetos:
 - La ubicación en un frame es cercana a la ubicación en el frame siguiente
 - La velocidad y la dirección permanecen relativamente sin cambios de un frame al siguiente.

Armado de trayectorias

- Las trayectorias se van armando a medida que se leen los frames.
- Algoritmo de correspondencia

Armar una trayectoria por cada objeto encontrado en el frame 1

Loop

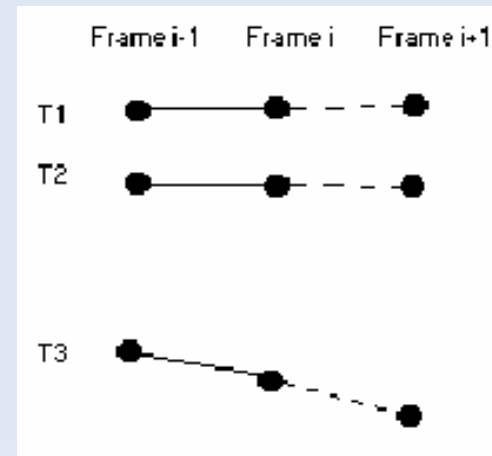
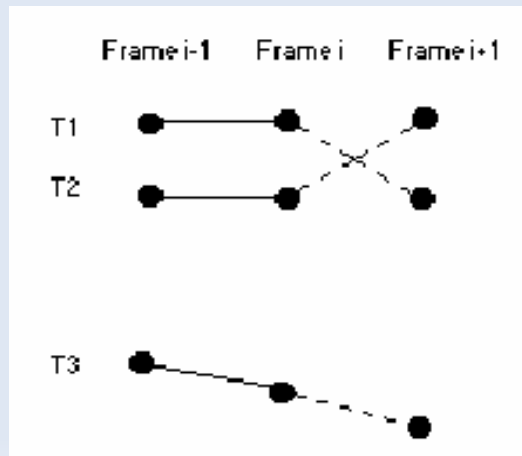
Adquirir nuevo frame

Encontrar los objetos

Actualizar las trayectorias con los nuevos objetos

Armado de trayectorias

- Actualizar trayectoria con los nuevos objetos
 - Inicialmente a cada trayectoria se le asigna un punto que este a distancia menor que d_{max} .
 - Para cada par de trayectorias se evalúa si conviene intercambiar los últimos puntos agregados usando la función de ganancia



Armado de trayectorias

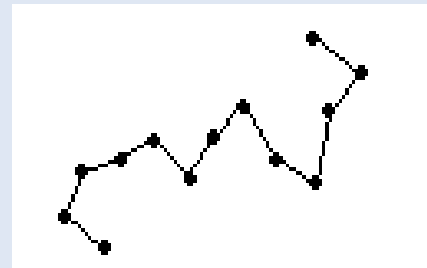
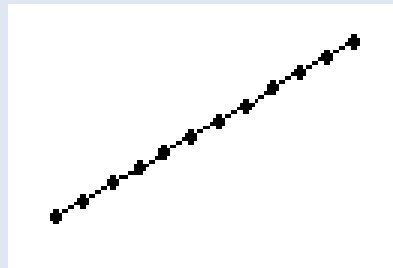
- Función de ganancia de intercambio
 - Sean P_{ki-1} , P_{ki} , P_{ki+1} los puntos de la trayectoria T_k correspondientes a los frames $i-1$, i e $i+1$ respectivamente; y P_{hi-1} , P_{hi} , P_{hi+1} los puntos de los mismos frames correspondientes a la trayectoria T_h . La función de ganancia determina la conveniencia de intercambiar los últimos puntos de tal manera que la trayectoria T_k quede formada por los puntos P_{ki-1} , P_{ki} , P_{hi+1} y la trayectoria T_h por los puntos P_{hi-1} , P_{hi} , P_{ki+1} .
 - Tiene en cuenta variación de velocidad y la variación de dirección.

Armado de trayectorias

- El algoritmo desarrollado tiene en cuenta las siguientes situaciones:
 - Un objeto puede no ser encontrado en un frame determinado
 - Detección de un objeto que no existe en la escena real (ruido)
 - Una trayectoria finaliza.
 - Una nueva trayectoria comienza.

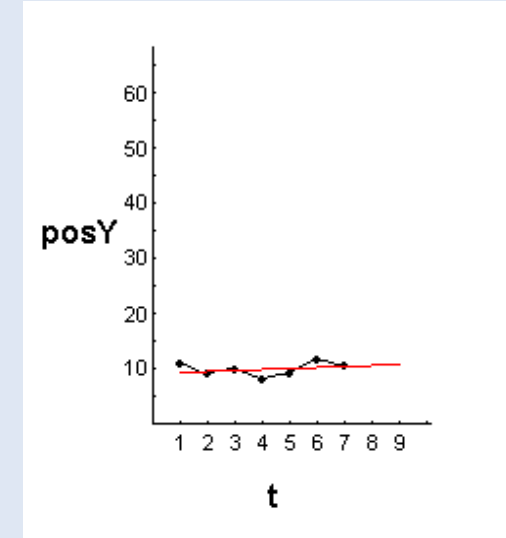
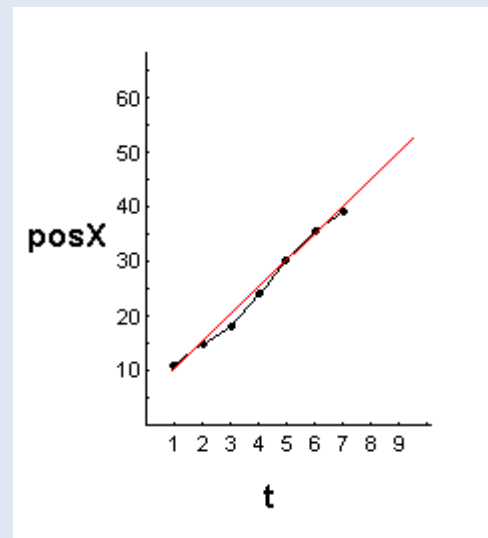
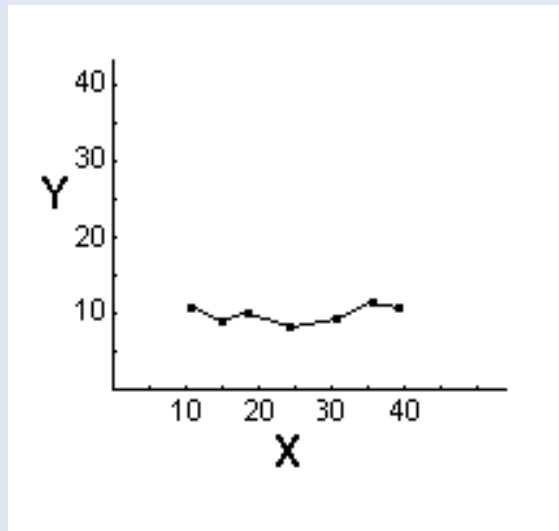
Estimación de trayectorias

- Se basa en la información del armado de la trayectoria para estimar el recorrido posterior.
- Depede de la cantidad de información previa que se tiene.
- La dirección y velocidad del movimiento se mantienen relativamente sin cambios si se analizan desplazamientos consecutivos de un objeto
- A mayor conocimiento de las características del movimiento de un objeto, mejor será la estimación.



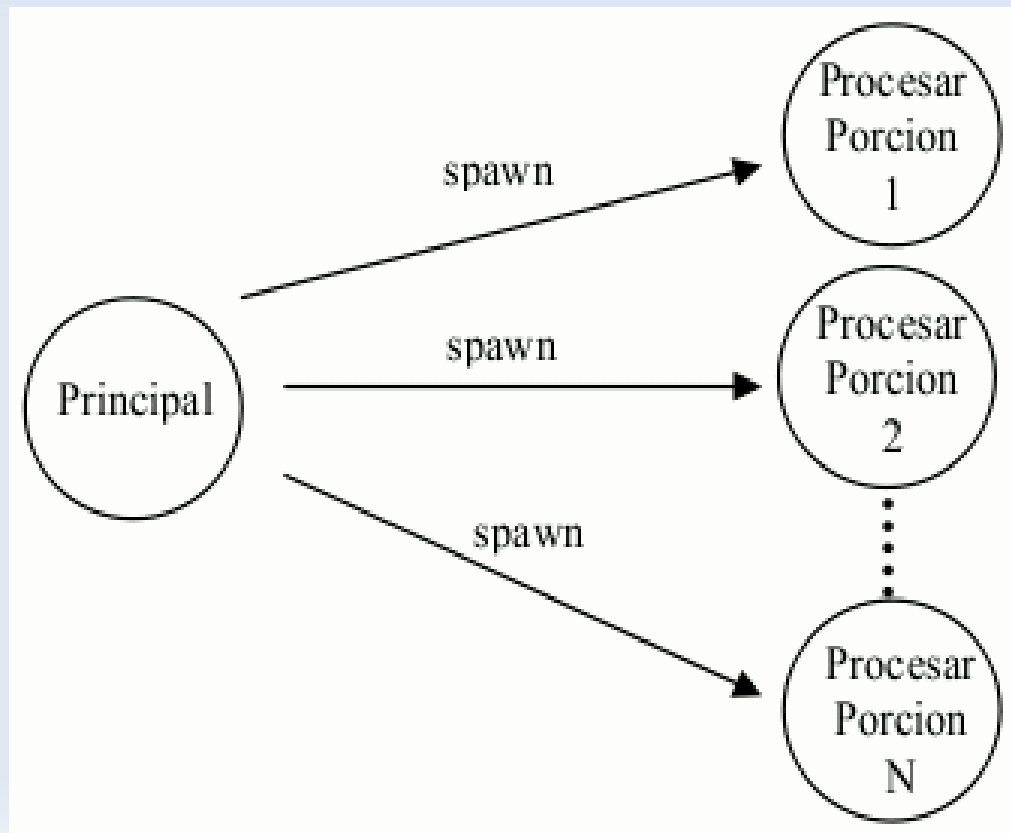
Estimación de trayectorias

- Estimación por regresión lineal por cada coordenada.
- Por cada componente se obtiene una función lineal que indica la predicción.



Implementación paralela

- Motivación: Tiempo real
- Forma en que se divide la imagen
- Proceso principal que hace de coordinador



Implementación paralela

- Captura de la imagen (Local a cada proceso)
- Mejora de la imagen
 - Aplicación de filtro (Local a cada proceso)
- Reconocimiento de objetos
 - Umbralización (Local a cada proceso)
 - Etiquetado (Procesamiento de bordes)
 - Calcular posición (Procesamiento de bordes)
- Armado y estimación de trayectorias (Centralizado)

Implementación paralela

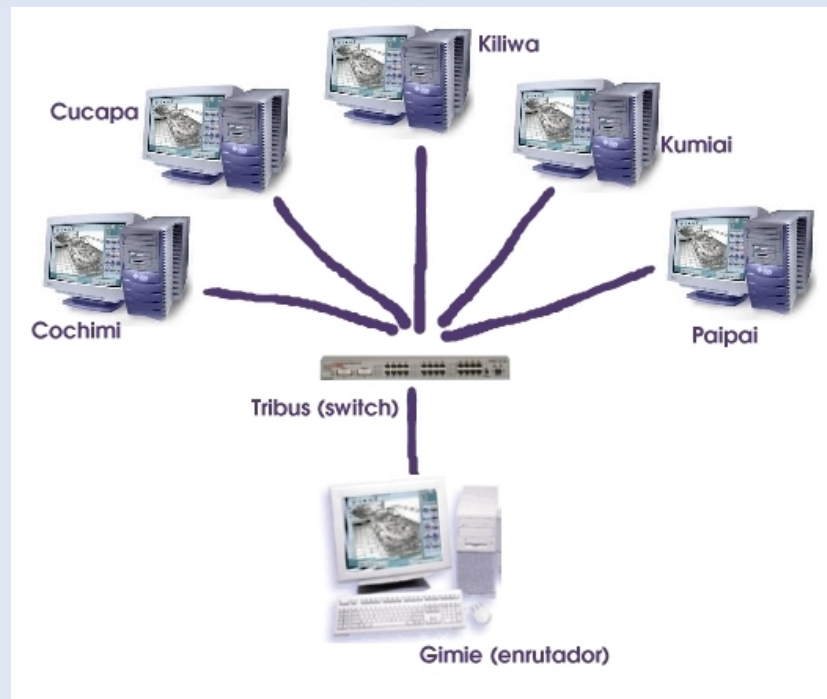
- Resolución de bordes en el etiquetado
 - Las etiquetas repetidas se solucionan cuantificando la etiqueta con el nro de proceso.
 - El que un mismo objeto esté etiquetado por mas de una marca se soluciona generando relaciones de equivalencia entre las etiquetas.
 - La relación de equivalencia que se genera es “pertenece al mismo objeto”
 - Estructura de datos : conjunto ajeno

Implementación paralela

- Resolución de bordes para calcular la posición
 - Uso de coordenadas unificadas
 - Cada proceso calcula la masa y las sumatorias de las coordenadas por cada etiqueta.
 - Para cada etiqueta que pertenece a la misma imagen se consolida la masa y las coordenadas.

Clusters de PCs

- Un cluster es un grupo de equipos independientes que ejecutan una serie de aplicaciones de forma conjunta y aparecen ante clientes y aplicaciones como un solo sistema.



Clusters de PCs

- Ventajas
 - Disponibilidad
 - Bajo costo
 - Alta escalabilidad
 - Heterogeneidad
 - Librerías libres de pasaje de mensajes (PVM, MPI)
- Desventajas
 - Alto costo de comunicación

PVM (Parallel Virtual Machine)



- <http://www.csm.ornl.gov/pvm/>
- Crea una abstracción empleando los recursos computacionales de todas las máquinas de la red que se pongan a disposición.
- API para diferentes lenguajes (C, Fortran, Java, etc.)
- Multiplataforma (Linux, Windows)

PVM (Parallel Virtual Machine)

- Comandos de consola
 - add, delete, conf, ps, halt, kill
- API para lenguaje C
 - pvm_spawn
 - pvm_send, pvm_recv, pvm_nrecv, pvm_probe
 - pmv_mytid
 - pvm_joingroup, pvm_lvgroup

Pruebas: infraestructura

- Hardware
 - Cluster homogéneo de 16 nodos (Sala IBM)
 - Intel Pentium 4, 2.40GHz.
 - 1GB de memoria RAM.
 - Red Ethernet
- Software
 - Lenguaje C
 - PVM
 - MPEGLib

Pruebas de rendimiento

- Objetivos
 - Determinar como se comportan los algoritmos analizando el tiempo de respuesta bajo distintos escenarios.
 - Determinar el máximo tamaño del video que puede ser procesado en tiempo real en función de la capacidad de procesamiento.
- Métricas
 - speedup.
 - eficiencia

Pruebas de rendimiento

- Unico modelo de video con diferentes tamaños
 - 100x400 (40.000 píxeles)
 - 500x2000 (1.000.000 píxeles)
 - 1000x4000 (4.000.000 píxeles)

Pruebas de rendimiento

- Speedup: medida que indica cuanto más rápido es un algoritmo paralelo respecto al equivalente secuencial

$$Sp = \frac{T1}{Tp}$$

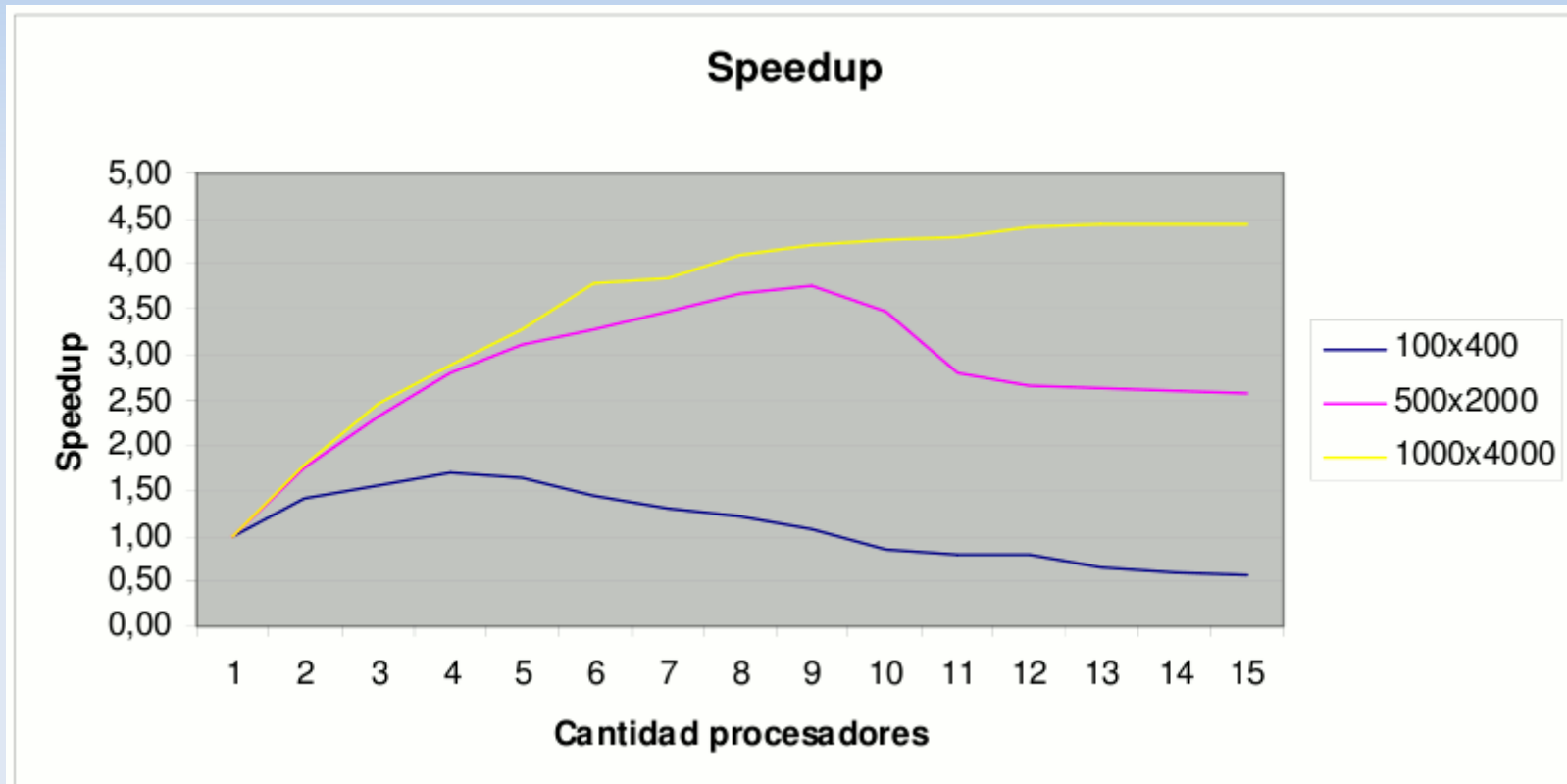
p = número de procesadores

T1 = tiempo de ejecución secuencial

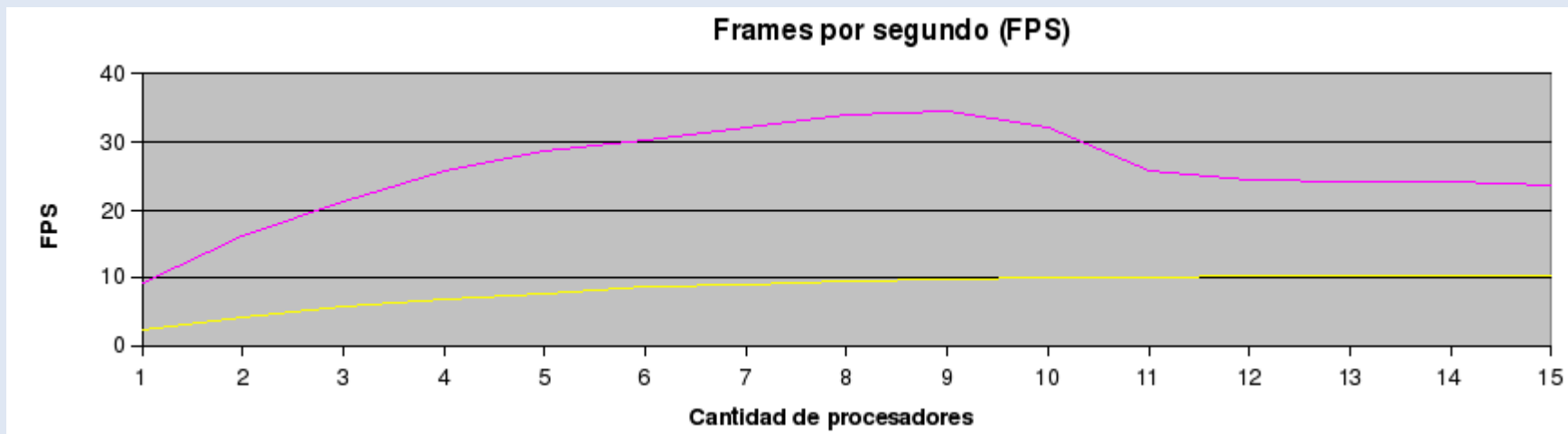
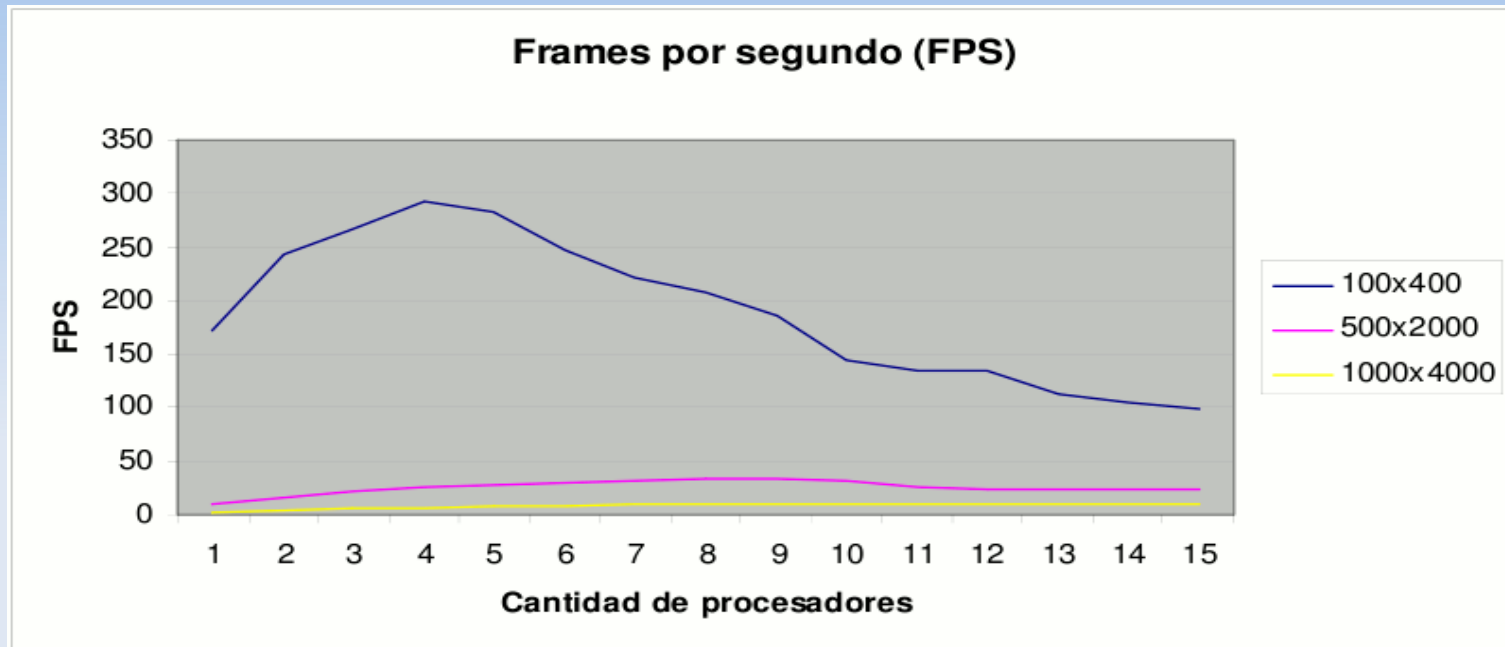
Tp = tiempo de ejecución utilizando p procesadores

- Speedup ideal $\rightarrow Sp = p$

Pruebas de rendimiento



Pruebas de rendimiento



Pruebas de rendimiento

- Eficiencia: medida que indica cuan eficazmente se utiliza el hardware

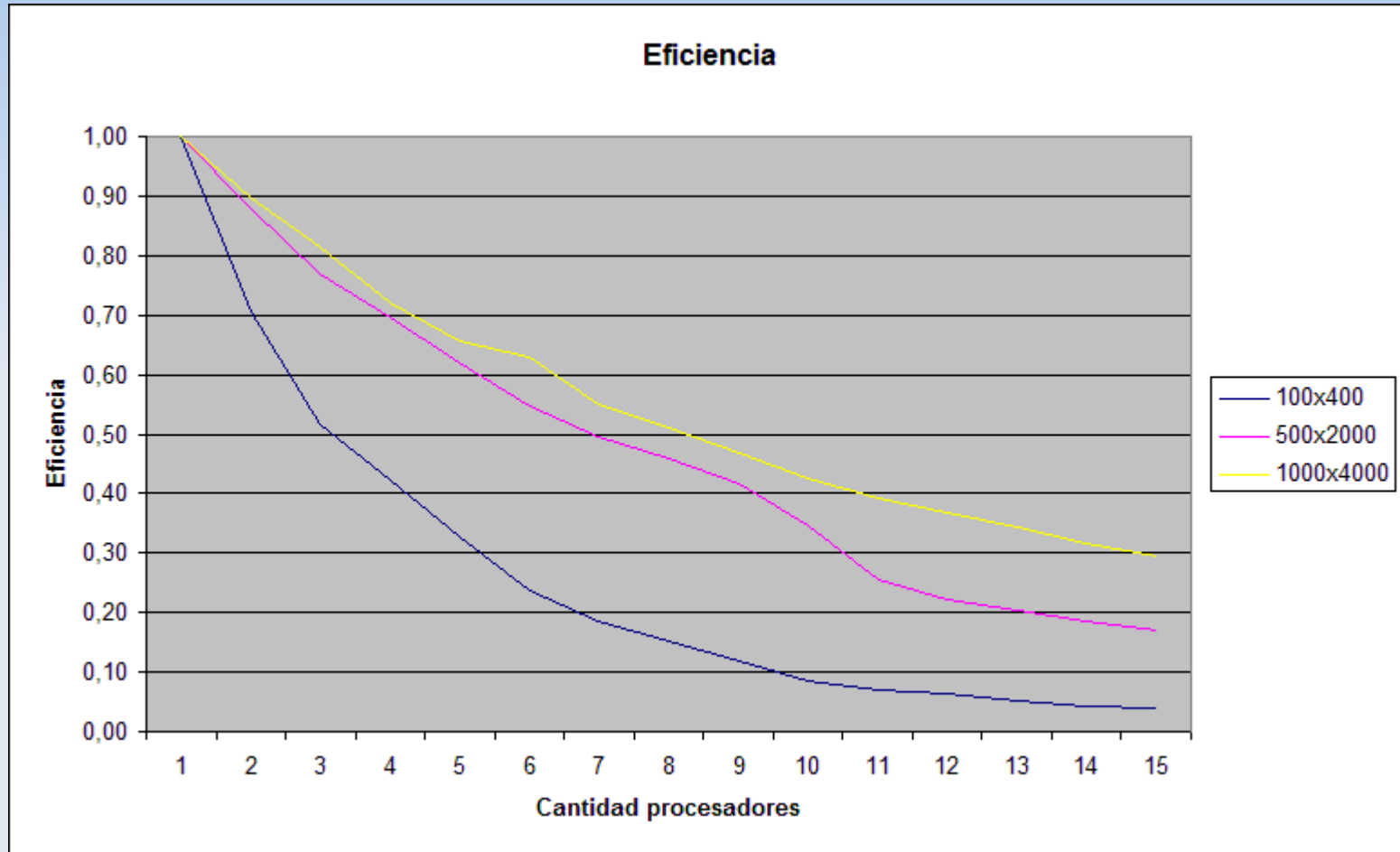
$$E_p = \frac{S_p}{p}$$

p = número de procesadores

S_p = speedup con p procesadores

- Valores entre 0 (poca eficiencia) y 1 (ideal)
- Típicamente E_p se aproxima a $\frac{1}{\log p}$

Pruebas de rendimiento



Pruebas de estimación

- Orientadas a comprobar el funcionamiento del algoritmo de estimación
 - Medir el error cometido en función de la cantidad de información utilizada.
 - Determinar la relación entre el error y la cantidad de frames en el futuro que se estiman.
- Error cometido: diferencia entre posición estimada y observada.

Pruebas de estimación

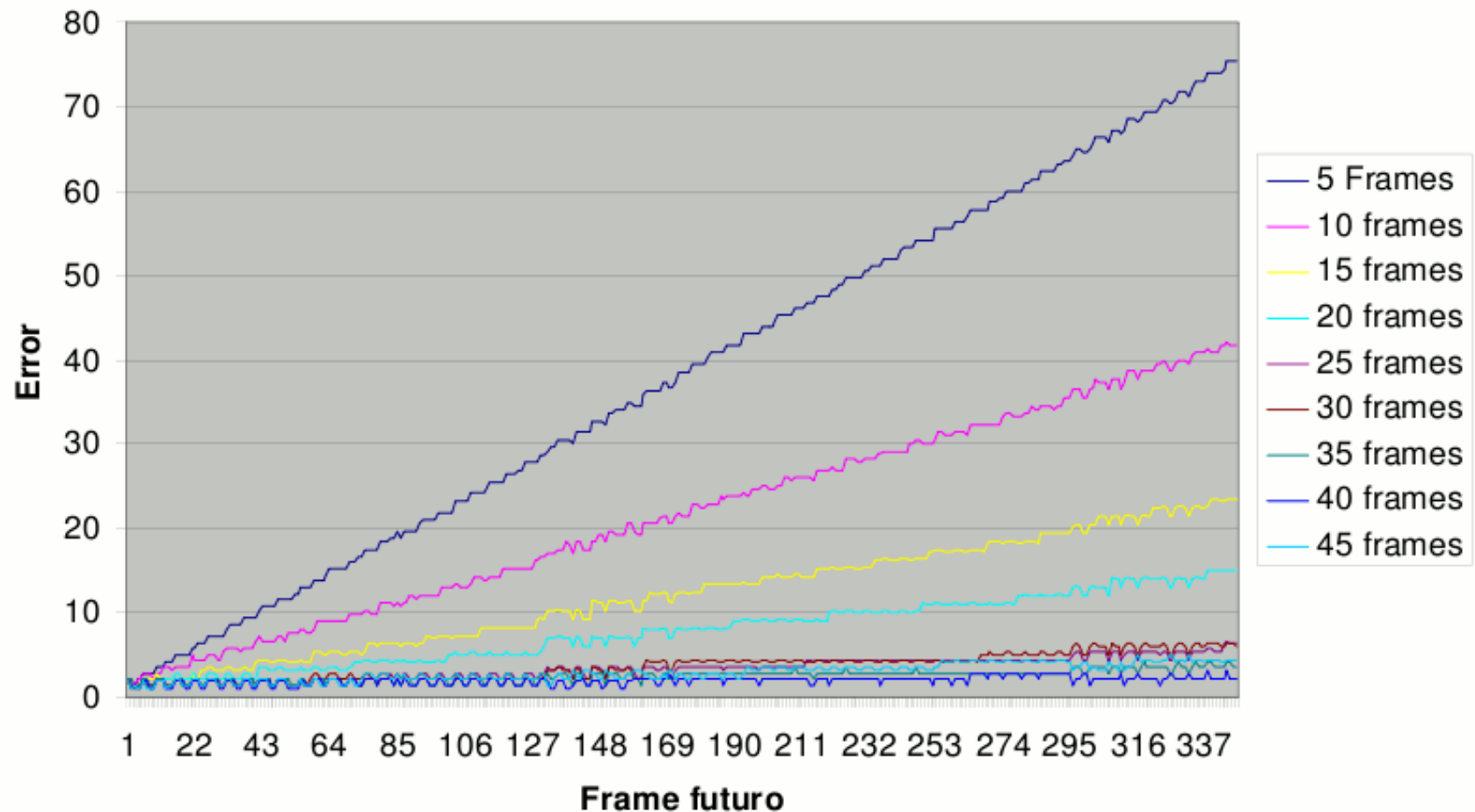
- Video de prueba
 - Única trayectoria
 - 40.000 píxeles (100x400).
 - Objeto de 70 píxeles de masa aproximadamente.

Pruebas de estimación

- Procedimiento
 - Por cada frame estimar puntos futuros variando cantidad de información previa utilizada
 - Comparar estimaciones con posiciones del objeto
 - Promediar error cometido en función del instante de tiempo estimado

Pruebas de estimación

Error cometido en función de la cantidad de frames usados para la estimación



Conclusiones

- Cuanto mayor información se utilice para estimar más precisos son los resultados
- Para la obtención de respuesta en tiempo real es necesario el uso de procesamiento paralelo
- Existe un punto a partir del cual agregar capacidad de procesamiento no mejora los tiempos de respuesta
- Diferencia importante entre speedup ideal y el obtenido

Líneas de trabajo futuro

- Otros modelos de movimiento
- Uso de procesadores multicore
- Imágenes de mayor tamaño
 - Concatenación de imágenes
 - Formatos de video que soporten mayor tamaño