

Técnicas de *tracking*

Seguimiento visual de objetos mediante el control de los parámetros de una cámara motorizada

DEPARTAMENTO DE INGENIERÍA DE SISTEMAS INDUSTRIALES
DIVISIÓN DE INGENIERÍA DE SISTEMAS Y AUTOMÁTICA

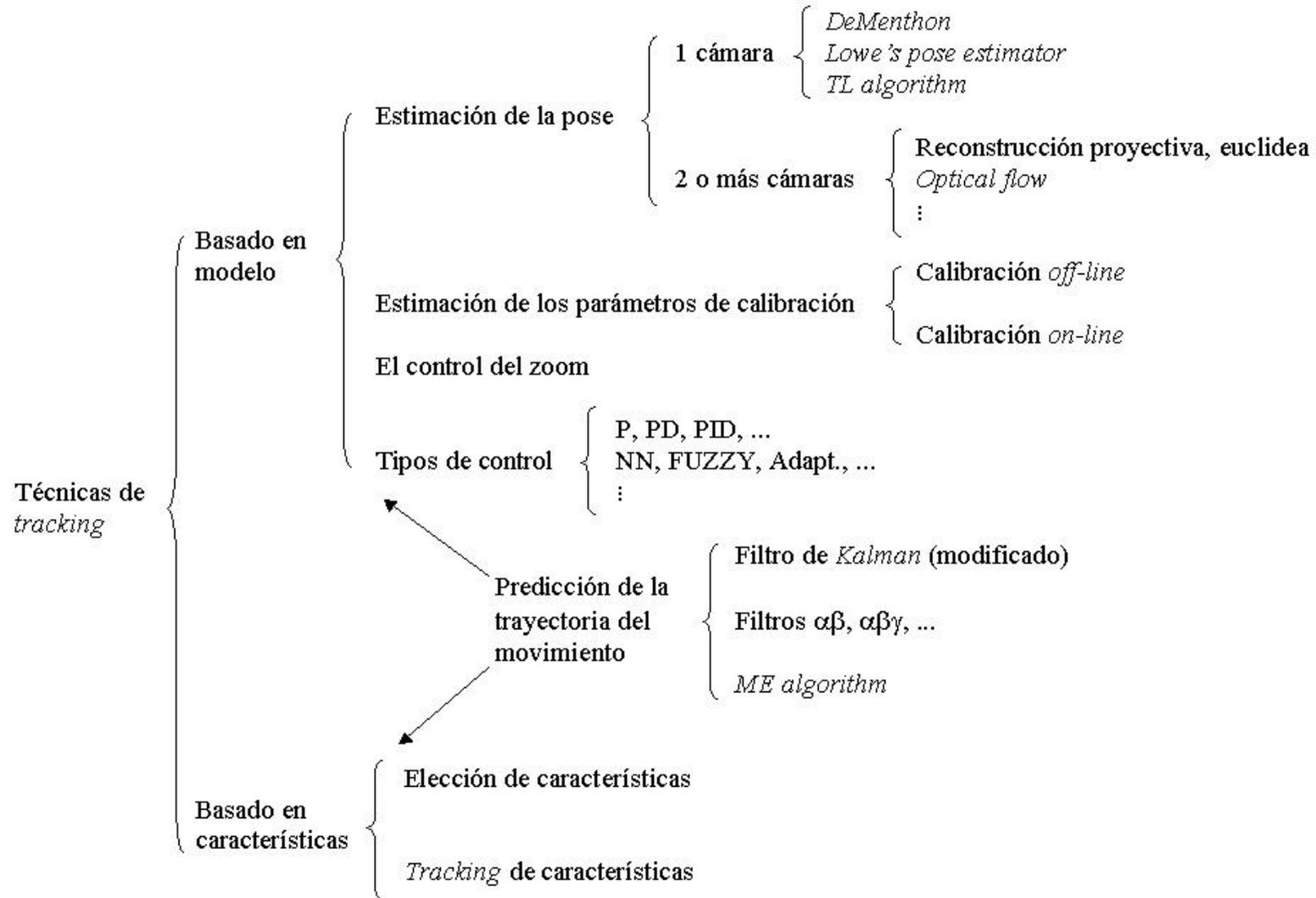
RED TEMÁTICA REDVISION

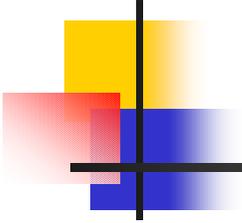
JORNADAS DE TRABAJO DE CONTROL VISUAL

22 Y 23 DE MAYO DE 2003



Esquema de la presentación



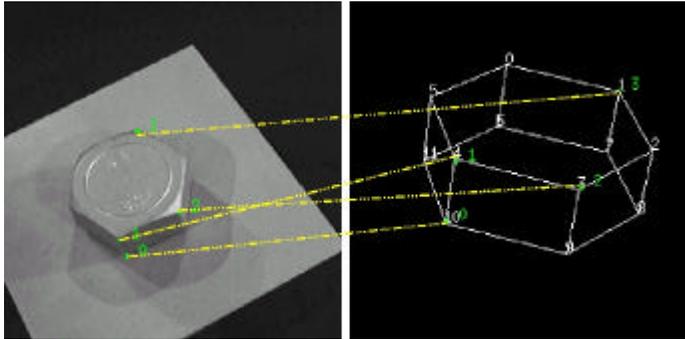


Técnicas de *tracking*

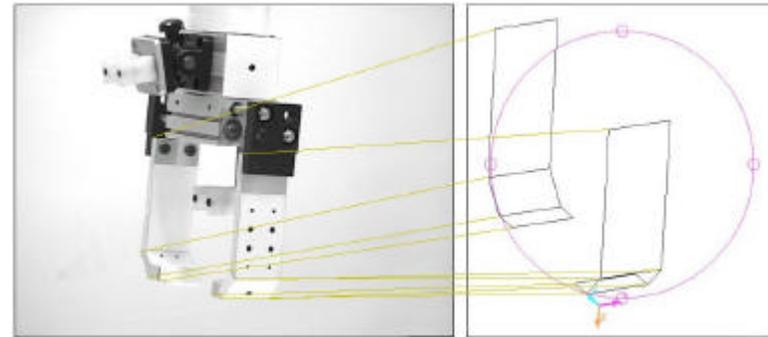
Definición de *tracking* óptico:

El tracking óptico es el control de los parámetros (posición, orientación y *zoom*) de una cámara en función de características extraídas de la imagen para mantener un determinado objeto en movimiento dentro del campo de visión.

Tracking basado en modelo



Ejemplo 1



Ejemplo 2

Este tipo de técnicas se basan en el conocimiento del modelo del objeto. Este modelo puede ser un modelo CAD o una proyección del objeto.

Esta técnica es más robusta que la basada en características.

Estimación de la pose

- Usando 1 cámara.

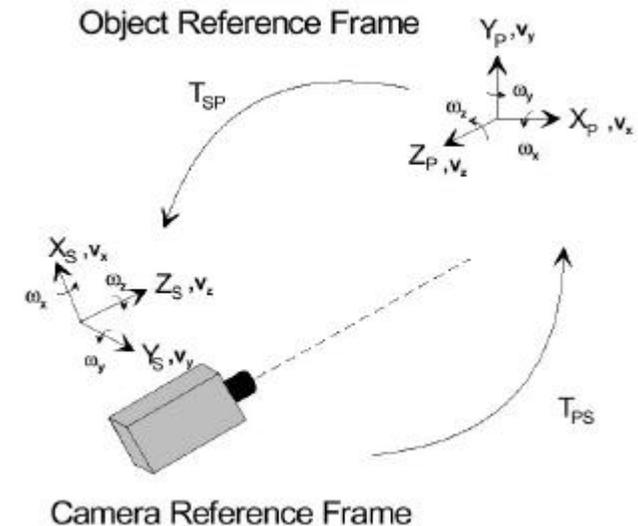
Con la información aportada por una sola imagen, podemos obtener la pose de un objeto si conocemos su modelo.

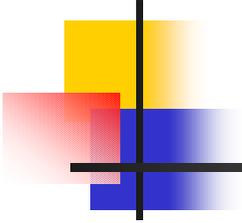
Para ello podemos emplear alguno de los algoritmos propuestos: *DeMenthon*, *Lowe*, *TL*, ...

- Usando 2 o más cámaras.

Teniendo varias imágenes, podemos obtener la pose de un objeto estableciendo la correspondencia entre puntos.

Para ello podemos emplear alguna de las técnicas propuestas: Reconstrucción proyectiva-euclídea, *Optical flow*, ...





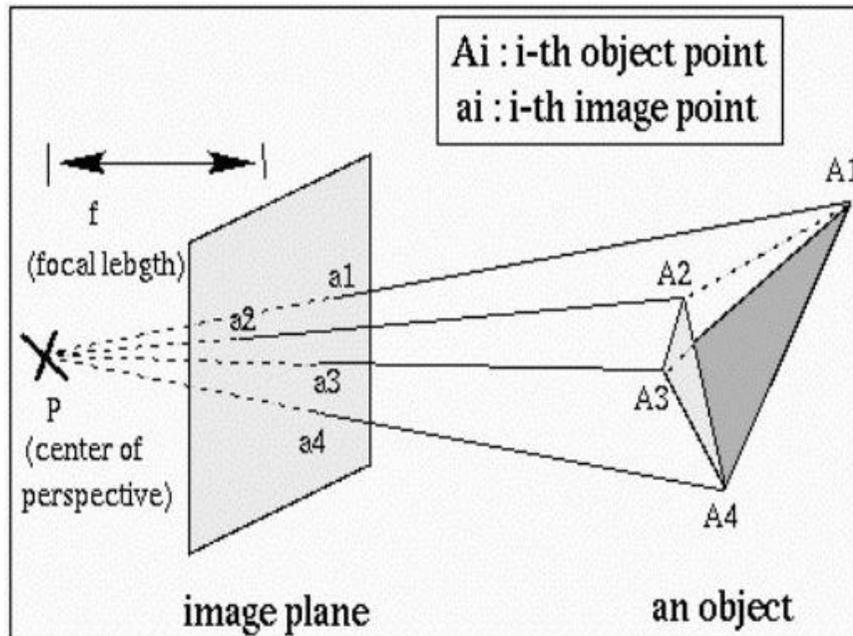
Lowe's pose estimator

(usando 1 cámara)

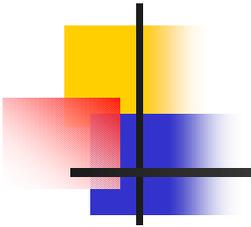
- *Lowe* propone una reparametrización de las ecuaciones de proyección para simplificar el cálculo y expresar la traslación en términos de coordenadas de la cámara.
- La estimación de la pose basada en modelo usando cuatro puntos se llama “problema de la perspectiva de cuatro-puntos” (P4P).

TL algorithm

(usando 1 cámara)



- Algoritmo del tipo P4P con menor gasto computacional que *Lowe* y basado también en estimación de la pose basada en modelo
- Cálculo iterativo basado en el método de *Gauss-Newton* para resolver el sistema de ecuaciones propuesto.

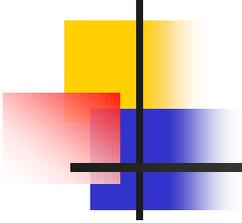


Estimación de los parámetros de calibración

Los parámetros de calibración son necesarios para la obtención de la pose del objeto.

Estos parámetros cambian con variaciones del *pan*, *tilt* o *zoom*, por lo que es necesario calibrar la cámara en todo el rango de estas variables.

La calibración se puede realizar *off-line* (antes del proceso de *tracking*) o *on-line* (durante el proceso de *tracking*).

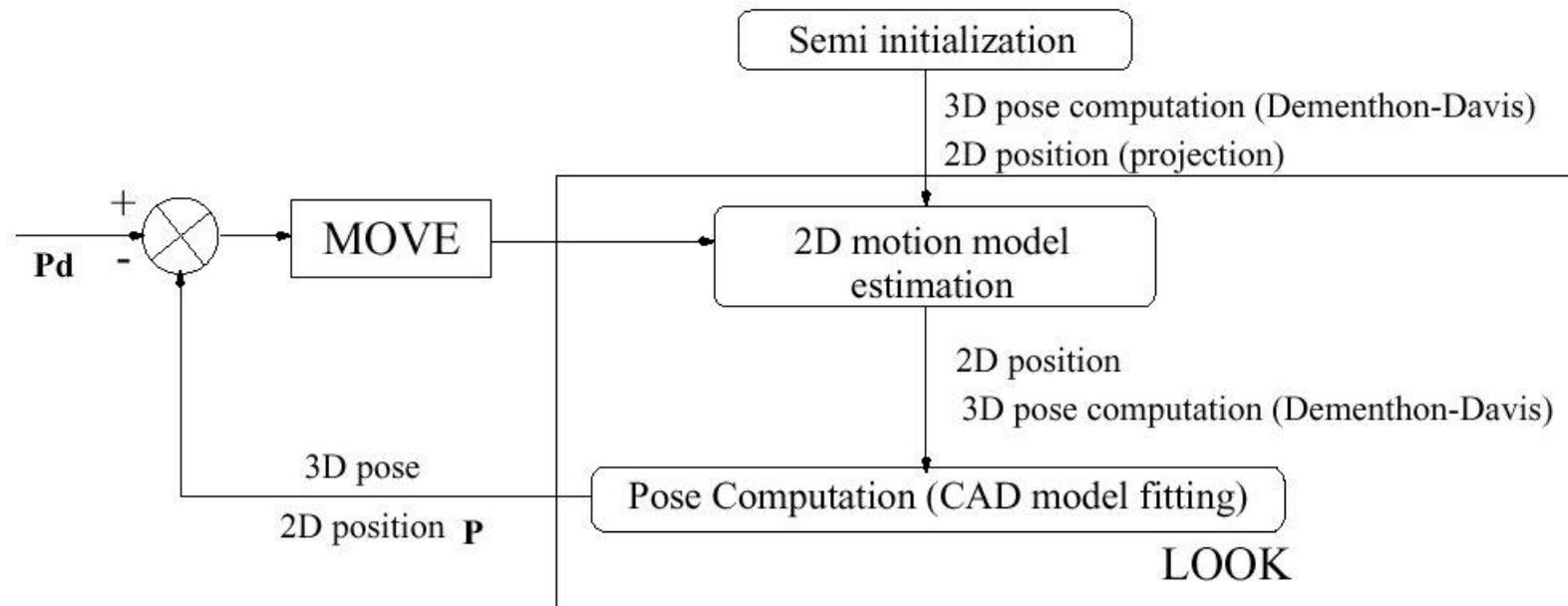


El control del *zoom*

Una de las características del *tracking* basado en modelo es que una vez obtenida la pose, se puede extender con facilidad el control para el manejo del *zoom* en función de la profundidad o alejamiento del objeto.

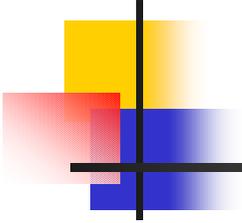
Con ello mantenemos el objeto con el mismo tamaño en la imagen aunque se acerque o aleje.

Tipos de control



Ejemplo para *tracking* basado en modelo y estimación de la pose con algoritmo de *DeMenthon*

El control a aplicar responde al esquema presentado en la figura. Donde el bloque *MOVE* implementa un regulador de los ya conocidos: P, PD, PID, NN, FUZZY, Adapt., ...

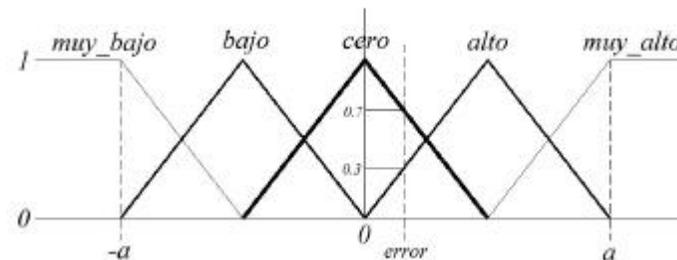
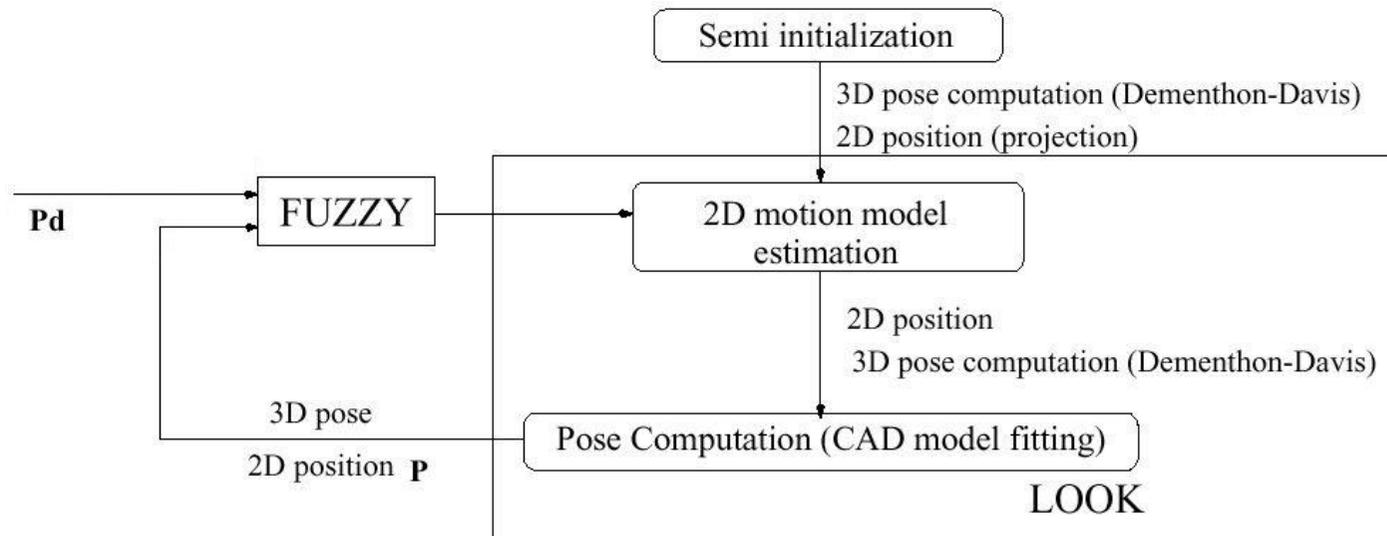


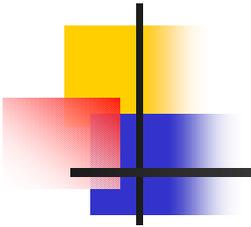
Tipos de control (II)

- Controles implementados en posición y velocidad.
- Algoritmos implementados:
 - Reguladores P y PD
 - Reguladores P con cambio de pendiente
 - Reguladores FUZZY
 - Reguladores FUZZY con parámetros adaptativos

Tipos de control (III)

■ Reguladores FUZZY



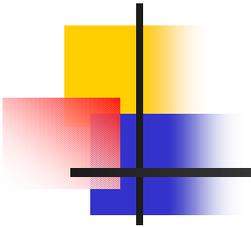


Tracking basado en características

Este tipo de *tracking* se basa en las características extraídas de la imagen y no en la búsqueda de un modelo conocido.

La idea fundamental en la que se basa el *tracking* de características es '*¿por qué hacer tracking del objeto entero cuando se puede obtener el mismo resultado haciendo tracking solo de las características?*'.

Este planteamiento suele ser computacionalmente más eficiente que el basado en modelo, pero es menos robusto.

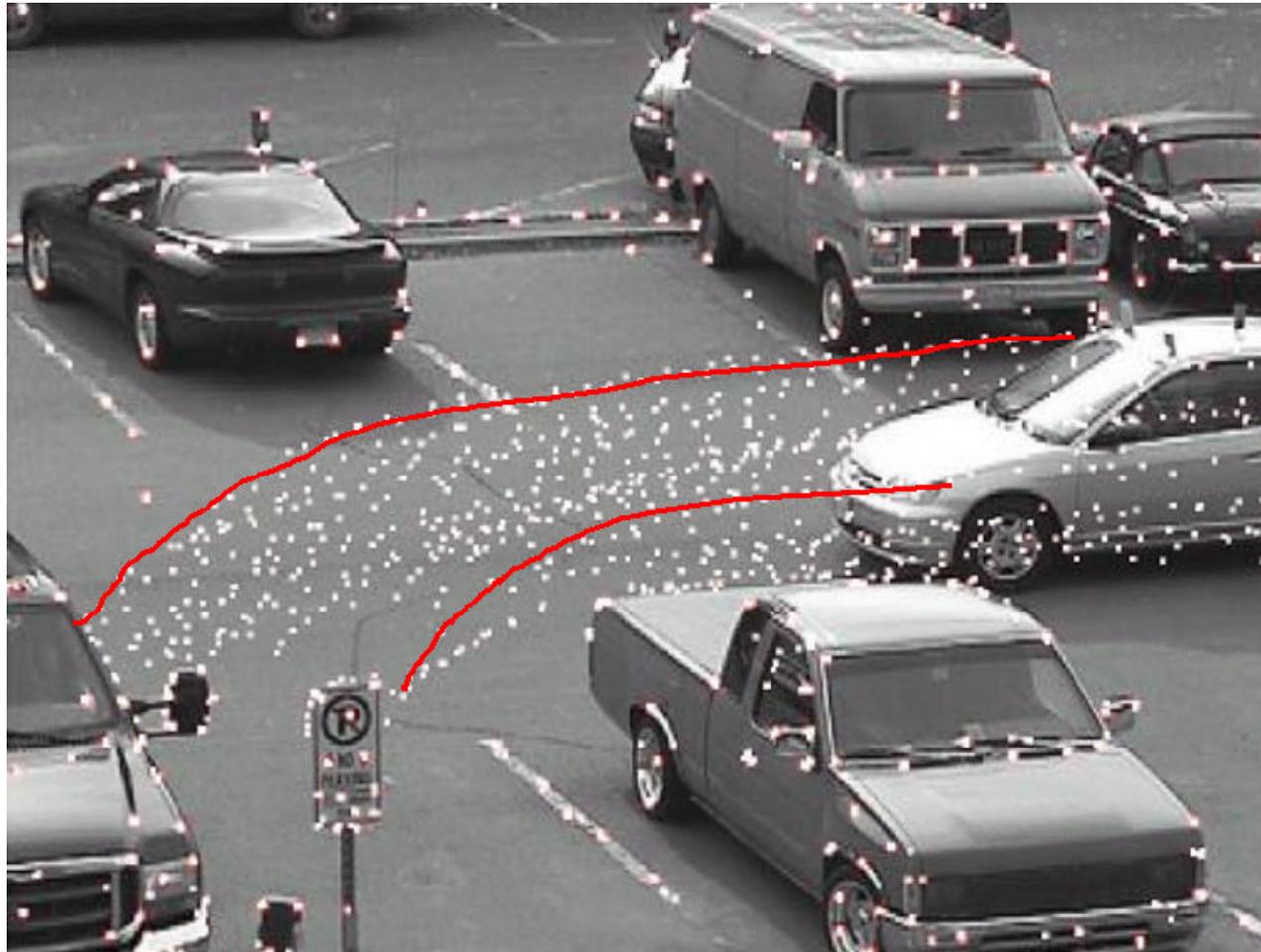


Elección y *tracking* de características

- El paso más importante antes de hacer el *tracking* es determinar las características a seguir. (p.e. esquinas).
- Discriminar entre buenas y malas características.
Calculamos para ello el vector gradiente de la imagen en ese punto y con los valores propios, podremos saber la calidad de la característica. Sólo aquellas que cumplan un cierto umbral serán finalmente elegidas.
- Una vez seleccionadas las características buenas a seguir, es sencillo llevar un registro del movimiento de cada característica, con lo que tenemos determinada la trayectoria de movimiento del objeto.

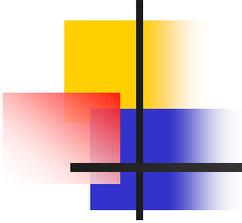
Tracking de características (II)

Trayectoria de movimiento del objeto



23/05/2003

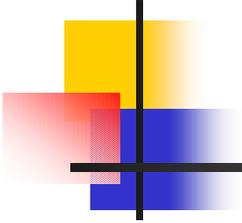
Autores: Carlos Pérez, Oscar Reinoso



Predicción de la trayectoria

Se realiza cuando los requerimientos dinámicos o la velocidad del objeto a seguir es lo suficientemente alta como para que su localización entre una imagen y la siguiente de la secuencia sea muy diferente.

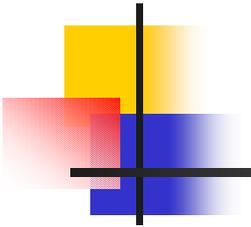
La predicción nos proporciona información sobre la posible pose cuando se esté analizando una ventana de la imagen, existan oclusiones parciales o totales del objeto o incluso éste salga del marco de imagen.



Filtro de *Kalman* (modificado)

El filtro de *Kalman* con sus extensiones EKF y EIKF son los algoritmos de predicción de la posición más utilizados (junto con el IMM – *Interacting Multiple Model*).

Se usa una combinación de los datos de obtención de la pose y predicción para hacer más robusto el algoritmo.



Filtros $\alpha\beta$, $\alpha\beta\gamma$, ...

- El filtro $\alpha\beta$, es un sencillo estimador de la posición con un tiempo de cómputo muy bajo y que proporciona unos resultados satisfactorios.
- Como modificación del primero, tenemos el filtro $\alpha\beta\gamma$, que realiza la estimación de la aceleración, efecto que no se contemplaba y que provocaba el mayor error de estimación.

Experimentos

Se han realizado experimentos con diferentes algoritmos de *tracking* centrados en el *tracking* basado en modelo con estimación de la pose de *DeMenthon*.

