

# Seguimiento de Puntos en Secuencia de Imágenes Ruidosas

Domingo Mery y Mónica Villanueva

Universidad de Santiago de Chile

Departamento de Ingeniería Informática

Av. Ecuador 3659, Santiago de Chile

eMail: {dmery,mvilla}@diinf.usach.cl

<http://www.diinf.usach.cl/>

## Abstract

In this paper a modification of an algorithm that performs a tracking of points in a noisy image sequence is presented. Using this modification the same performance in the tracking is obtained. However, the execution time is reduced considerably. The algorithms perform the tracking according to the multiple view geometry in a calibrated image sequence, where the parameters of the function that transforms the 3D space into the 2D images are known. The key idea of these algorithms is to consider as noise those points that cannot be tracked in the sequence. The presented modification is performed using indexes that reduce the number of evaluations in the search of corresponding points. Additionally, the algorithm is assed in several cases where the number of noisy points and the noise in the measurement of the points to be tracked are varied. Using this study, it is possible to know the performance of the tracking method. An example that shows a perfect tracking of 8 points in a sequence of 10 images with 500 noisy points per image is shown.

**Keywords:** Tracking, computer vision, multiple view geometry, epipolar geometry, trifocal tensors, image processing.

## Resumen

En este artículo se presenta una modificación de un algoritmo que realiza el seguimiento de puntos en una secuencia de imágenes ruidosas. Con esta modificación se obtienen los mismos resultados en cuanto al desempeño del algoritmo, sin embargo se reduce considerablemente el tiempo de ejecución. Ambos algoritmos realizan el seguimiento sobre la base de las restricciones multi-focales existentes en la geometría multi-imagen para una secuencia calibrada de imágenes, es decir donde los parámetros de la función que realiza la transformación del espacio 3D a las imágenes 2D de la secuencia son conocidos. La idea principal de estos algoritmos es considerar como ruido aquellos puntos que no pueden ser seguidos en la secuencia. La modificación presentada se basa en la utilización de índices para reducir el número de evaluaciones de correspondencia en los puntos candidatos. Adicionalmente, se evalúa el algoritmo ante diversas situaciones en las que se hace variar el número de puntos de ruido en las imágenes y el ruido presente en la medición de la ubicación de los puntos a seguir. Con este estudio es posible medir fácilmente el desempeño del algoritmo. Se muestra un ejemplo de seguimiento perfecto de 8 puntos en una secuencia de 10 imágenes que presenta 500 puntos ruidosos por imagen.

**Palabras claves:** Seguimiento, visión artificial, geometría multi-imagen, geometría epipolar, tensores trifocales, procesamiento de imágenes.

## 1 Introducción

Un algoritmo para el seguimiento de puntos en una secuencia de imágenes ruidosas fue presentado recientemente en [8, 7]. El algoritmo fue desarrollado para realizar una detección automática de fallas en piezas de metal fundido usando secuencias de imágenes radioscópicas<sup>1</sup>. Si bien es cierto que el algoritmo es eficaz en cuanto a la eliminación de ruido, el tiempo de ejecución puede reducirse considerablemente. En este artículo se presenta una modificación que acelera el algoritmo sin afectar su desempeño. La modificación presentada se basa en la utilización de índices para reducir el número de evaluaciones de correspondencia en los puntos candidatos.

Adicionalmente, se evalúa el algoritmo ante diversas situaciones en las cuales se hace variar el número de puntos de ruido en las imágenes y el ruido presente en la medición de la ubicación de los puntos a seguir. Con este estudio es posible medir fácilmente el desempeño del algoritmo.

El artículo se ha organizado de la siguiente manera: La sección 2 presenta una introducción al análisis multi-imagen. En la sección 3 se describe el algoritmo original. La modificación a este algoritmo es explicada en la sección 4. Los resultados experimentales son expuestos en la sección 5. Finalmente, se presentan en la sección 6 las conclusiones a las que se ha llegado y sugerencias para trabajos futuros.

## 2 Análisis Multi-imagen

En esta sección se presenta el modelo geométrico que establece la relación entre las coordenadas 3D del objeto de análisis y las coordenadas 2D del píxel de las imágenes tomadas. Adicionalmente, se establecen las restricciones geométricas y algebraicas existentes en dos y tres imágenes tomadas del mismo objeto desde distintos puntos de vista. Finalmente, se describe el problema de reconstrucción 3D.

Para este análisis se supone que fueron tomadas  $N$  imágenes del objeto. El índice  $p$ ,  $p = 1, \dots, N$ , es usado para indicar la proyección  $p$  del objeto. Esta proyección se obtiene mediante un movimiento de la cámara o del objeto. Se asume que el sistema de adquisición de imágenes está calibrado. Esto quiere decir que los parámetros intrínsecos y extrínsecos de la función de proyección del mundo 3D a las imágenes 2D ya fueron estimados mediante alguna técnica de calibración (ver por ejemplo [2]). En esta sección se le brinda al/a lector/a una introducción al tema de análisis multi-imagen, un estudio extensivo se puede encontrar en los libros [6, 3].

### 2.1 Proyección de una vista

El modelo de proyección que se usará en esta explicación es el de la proyección en perspectiva [2], mostrado en la Figura 1. En este modelo hay un punto  $M$  en el espacio 3D que es proyectado en perspectiva a un plano de imagen  $R$  a través de un centro óptico  $C$ . La proyección del punto  $M$  en el plano es el punto  $m$ . A continuación se definirán sistemas de coordenadas para establecer la relación del punto 3D  $M$  y su imagen 2D  $m$ . En estas relaciones se usarán *coordenadas homogéneas* mediante las cuales un punto  $(x_1, x_2, \dots, x_n)$  en un espacio de  $n$  dimensiones es expresado como un vector de  $n + 1$  elementos  $(y_1, y_2, \dots, y_n, y_{n+1})$  donde  $x_i = y_i/y_{n+1}$  para  $i = 1, \dots, n$ .

El **sistema 3D de coordenadas del objeto** está referido al objeto 3D de estudio. Un punto  $M$  del objeto en este sistema de coordenadas se expresa como  $\mathbf{M} = [X \ Y \ Z \ 1]^T$  en coordenadas homogéneas. Se asume que el centro de rotación del objeto se encuentra en el origen  $O$  de este sistema de coordenadas.

El **sistema 3D de coordenadas de referencia** es definido en el centro óptico de la proyección que tiene lugar en la cámara. Esto quiere decir que el origen de este sistema de coordenadas  $C$  coincide con el centro óptico de la proyección. El punto del objeto  $M$  es visto en la posición  $p$  en este sistema de coordenadas como  $\bar{\mathbf{M}}_p = [\bar{X}_p \ \bar{Y}_p \ \bar{Z}_p \ 1]^T$  en coordenadas homogéneas. El sistema de coordenadas del objeto es considerado como un desplazamiento rígido del sistema de coordenadas de referencia representado por la matriz de rotación  $\mathbf{R}_p$  de  $(3 \times 3)$  elementos y el vector de traslación vector  $\mathbf{t}_p$  (de  $3 \times 1$  elementos) [2]. Utilizando la matriz de  $4 \times 4$   $\mathbf{S}_p$ :

$$\mathbf{S}_p = \begin{bmatrix} \mathbf{R}_p & \mathbf{t}_p \\ \mathbf{0} & 1 \end{bmatrix} \quad (1)$$

se obtiene una relación lineal entre el sistema de coordenadas del objeto y el sistema de coordenadas de referencia:

$$\bar{\mathbf{M}}_p = \mathbf{S}_p \mathbf{M}. \quad (2)$$

A continuación se define el **sistema 2D de coordenadas en el plano retinal**. Este nuevo sistema de coordenadas se usa para indicar las coordenadas de la proyección del punto 3D en el plano de imagen,

---

<sup>1</sup>Resultados en experimentos reales pueden encontrarse en <http://www.diinf.usach.cl/~dmery/sequences.htm>.

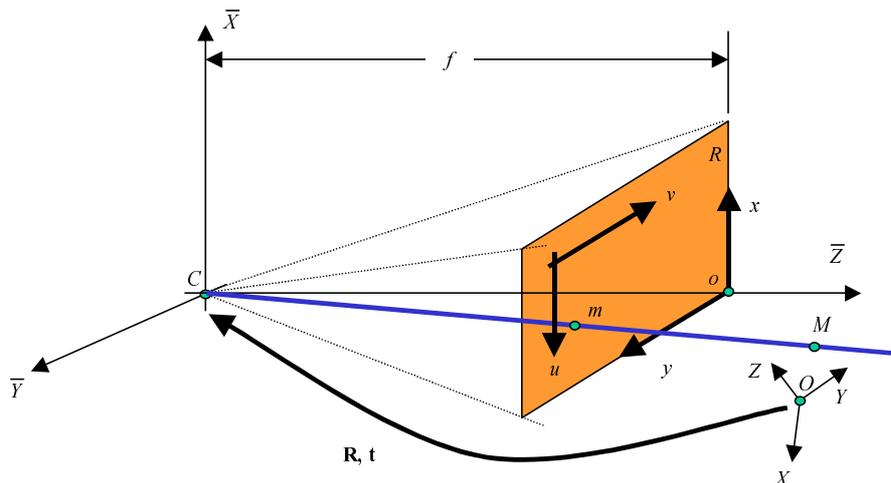


Figura 1: Modelo de proyección.

llamado también plano retinal. Tal como se ilustra en la Figura 1, el plano retinal  $R$  se encuentra en  $\bar{Z} = f$  y el eje óptico  $\bar{Z}$  intersecta este plano en el origen  $o$  de este nuevo sistema de coordenadas 2D. La variable  $f$  es conocida como la distancia focal de la proyección.

En la proyección  $p$  el punto  $M$  es visto como el punto  $m_p$  en el plano de proyección. Aplicando el Teorema de Thales, la representación homogénea de  $m_p$  en este nuevo sistema de coordenadas es  $\mathbf{m}_p = [x_p \ y_p \ 1]^T$ , donde  $x_p = f\bar{X}_p/\bar{Z}_p$  y  $y_p = f\bar{Y}_p/\bar{Z}_p$ . Usando las representaciones homogéneas de  $M$  y  $m_p$  ( $\bar{\mathbf{M}}_p$  y  $\mathbf{m}_p$  respectivamente) se obtiene la siguiente ecuación lineal:

$$\lambda_p \mathbf{m}_p = \underbrace{\begin{bmatrix} f & 0 & 0 & 0 \\ 0 & f & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}}_{\mathbf{B}} \bar{\mathbf{M}}_p, \quad (3)$$

donde  $\lambda_p$  es un factor de escala.

Finalmente, el **sistema 2D de coordenadas de la imagen** se utiliza para representar las coordenadas del píxel en la imagen obtenida por la cámara. El punto  $\mathbf{m}_p$  es visto en la cámara como un punto  $\mathbf{w}_p = [u_p \ v_p \ 1]^T$  en coordenadas homogéneas. El nuevo sistema de coordenadas dado por los ejes  $u$  y  $v$ , como se ilustra en la Figura 1, ha sufrido una rotación y una traslación de los ejes  $x$  e  $y$ . Es posible que también exista un cambio de escala entre ambos ejes como ocurre en las cámaras CCD, donde las unidades de los ejes están dadas en píxeles y no en milímetros por ejemplo. Adicionalmente, muchas veces los ejes  $u$  y  $v$  no son ortogonales debido a que los píxeles en los arreglos CCD no son rectangulares. En este caso es necesario introducir en el modelo de la cámara un factor de torcimiento (*skew factor*). En términos generales se puede decir que entre ambos sistemas de coordenadas existe una transformación general proyectiva 2D [6] que puede ser expresada como:

$$\mathbf{w}_p = \mathbf{H}\mathbf{m}_p \quad (4)$$

donde  $\mathbf{H}$  es una matriz homogénea de  $3 \times 3$  elementos. Resumiendo, se puede obtener una expresión que relaciona las coordenadas de  $M$  en el sistema de coordenadas del objeto con las coordenadas de  $w_p$  en la imagen a partir de las ecuaciones (2), (3) y (4):

$$\lambda_p \mathbf{w}_p = \mathbf{P}_p \mathbf{M} \quad (5)$$

donde  $\mathbf{P}_p = \mathbf{H}\mathbf{B}\mathbf{S}_p$  es una matriz de  $3 \times 4$  elementos y corresponde a la matriz de proyección para la vista  $p$ . Independientemente del tipo de proyección que se utilice en el modelo (perspectiva, paraperspectiva, ortográfica, etc. [3]) la transformación de coordenadas  $3D \rightarrow 2D$  se lleva a cabo por medio de una matriz de proyección de  $3 \times 4$  elementos. La ecuación (5) es entonces válida para cualquier proyección siempre y cuando la matriz de proyección  $\mathbf{P}_p$  se defina claramente para el tipo de proyección.

En las siguientes secciones se estudiarán las restricciones geométricas y algebraicas usadas para resolver el problema de correspondencia en imágenes de un mismo objeto tomadas desde distintos puntos de vista.

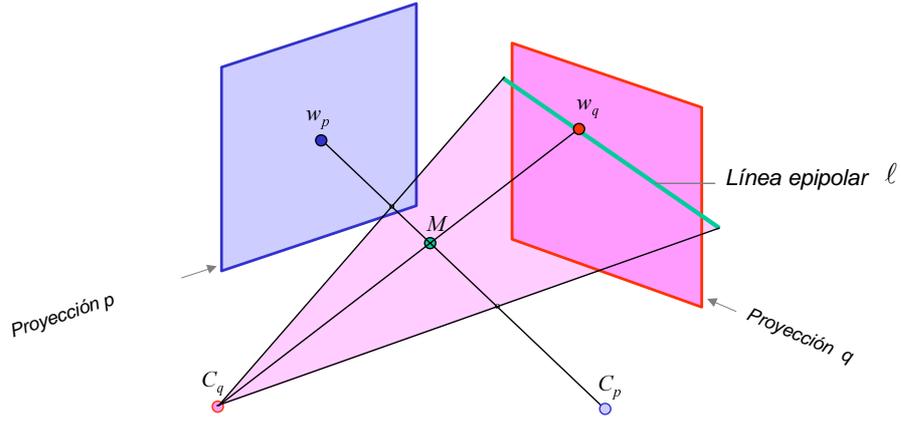


Figura 2: Geometría epipolar para dos vistas.

## 2.2 Análisis de dos vistas

Dos puntos (en dos vistas distintas) son correspondientes si son proyecciones del mismo punto tridimensional. Si un punto  $w_p$  y un punto  $w_q$  son correspondientes, quiere decir entonces que son proyecciones de un punto 3D  $M$  en las proyecciones  $p$  y  $q$  respectivamente.

Para resolver el problema de correspondencia en dos imágenes se usa comúnmente la *geometría epipolar* [2]. La restricción epipolar es muy conocida en visión estéreo: para cada punto  $w_p$  en la proyección  $p$ , su punto correspondiente  $w_q$  en la proyección  $q$  se encuentra sobre la línea epipolar  $\ell$  de  $w_p$ . Esta situación es mostrada en la Figura 2, donde  $C_p$  y  $C_q$  son los centros ópticos de las proyecciones  $p$  y  $q$  respectivamente. En esta representación se ha asumido una rotación y traslación relativa al sistema de coordenadas del objeto. Una representación análoga puede ser obtenida si los planos retinales se encuentran entre los centros ópticos y el objeto, o bien los centros ópticos entre los planos y el objeto. Cualquiera que sea el caso, la línea epipolar  $\ell$  puede ser calculada como la línea  $\langle w_p, C_p \rangle$  proyectada por el centro de proyección  $C_q$  en el plano  $q$ . Después de algunos cálculos (ver [2, 6, 3]) se obtiene la relación lineal:

$$\mathbf{w}_q^T \mathbf{F}_{pq} \mathbf{w}_p = 0 \quad (6)$$

donde  $\mathbf{F}_{pq}$  es la matriz de  $3 \times 3$  elementos conocida como la *Matriz Fundamental* de las proyecciones  $p$  y  $q$ , que depende sólo de las matrices de proyección  $\mathbf{P}_p$  y  $\mathbf{P}_q$  (y no de los puntos proyectados).

## 2.3 Análisis de tres vistas

Dados tres puntos  $w_p$ ,  $w_q$  y  $w_r$  (en las proyecciones  $p$ ,  $q$  y  $r$  respectivamente), la proyección del punto correspondiente 3D  $M$  en las tres vistas puede ser expresado como una ecuación matricial usando la ecuación (5) tres veces [6]:

$$\underbrace{\begin{bmatrix} \mathbf{P}_p & \mathbf{w}_p & \mathbf{0} & \mathbf{0} \\ \mathbf{P}_q & \mathbf{0} & \mathbf{w}_q & \mathbf{0} \\ \mathbf{P}_r & \mathbf{0} & \mathbf{0} & \mathbf{w}_r \end{bmatrix}}_{\mathbf{G}} \underbrace{\begin{bmatrix} \mathbf{M} \\ -\lambda_p \\ -\lambda_q \\ -\lambda_r \end{bmatrix}}_{\mathbf{v}} = \mathbf{0}, \quad (7)$$

Si  $w_p$ ,  $w_q$  y  $w_r$  son puntos correspondientes, entonces existe una solución no trivial para  $\mathbf{v}$  en (7). Esto implica que cualquier subdeterminante  $7 \times 7$  de la matriz  $\mathbf{G}$  de  $9 \times 7$  elementos es igual a cero. Esta condición conduce a cuatro relaciones *trilineales* independientes, conocidas como las *Trilinearidades de Shashua* entre los vectores  $\mathbf{w}_p$ ,  $\mathbf{w}_q$  y  $\mathbf{w}_r$  [9, 10, 5]. Usando una estructura llamada *tensor trifocal*, que depende sólo de las matrices de proyección  $\mathbf{P}_p$ ,  $\mathbf{P}_q$  y  $\mathbf{P}_r$  (y no de los puntos proyectados), se puede calcular directamente las coordenadas del tercer punto  $w_r$ , a partir de los puntos  $w_p$  y  $w_q$ .

## 2.4 Reconstrucción 3D a partir de dos o más vistas

Para  $n$  proyecciones distintas en las que se han encontrado los puntos correspondientes  $w_i$ ,  $i = 1, \dots, n$ , con coordenadas  $(u_i, v_i)$  se puede reconstruir el punto correspondiente 3D  $M$  que ha producido los puntos proyectados. Esto se hace utilizando la ecuación (5) para cada una de las proyecciones:  $\lambda_i \mathbf{w}_i = \mathbf{P}_i \mathbf{M}$ . Cada proyección representa tres ecuaciones lineales para las variables desconocidas  $X$ ,  $Y$ ,  $Z$  y  $\lambda_i$ . Luego de algunos

arreglos se obtiene un sistema sobredeterminado que puede ser resuelto para encontrar  $[X Y Z]^T$  utilizando una técnica de mínimos cuadrados [2]. Cabe señalar que también existe una técnica lineal y directa para estimar  $M$  a partir de dos puntos correspondientes (en dos vistas) [4].

### 3 Seguimiento de Puntos

En esta sección se explicará el algoritmo original presentado en [7, 8] para realizar el seguimiento de puntos en una secuencia de imágenes. Se supone que los puntos ya han sido detectados en cada una de las imágenes utilizando alguna técnica de segmentación [1]. El seguimiento consta de las siguientes etapas: *i)* *matching* en dos vistas; *ii)* seguimiento en tres vistas; *iii)* seguimiento en cuatro vistas; y *iv)* seguimiento en más vistas.

#### 3.1 Correspondencia en dos vistas

Un punto detectado puede ser considerado como la proyección en la imagen de un punto 3D del objeto de análisis. Como el objeto puede ser proyectado en varias imágenes de la secuencia, es posible que un punto en una imagen tenga un punto correspondiente en otra imagen. Para establecer si dos puntos  $w_a$  y  $w_b$ , en dos imágenes distintas  $p$  y  $q$  respectivamente, son correspondientes se utilizan los siguientes criterios:

**Condición epipolar:** *Los puntos deben satisfacer la condición epipolar.* Para este fin se utiliza comúnmente la restricción epipolar práctica, que señala que  $w_a$  y  $w_b$  satisfacen la condición epipolar si  $w_b$  se encuentra a una distancia Euclídea suficientemente pequeña de la línea epipolar de  $w_a$  en la segunda imagen [5]:

$$d_2(\mathbf{w}_a, \mathbf{F}_{pq}, \mathbf{w}_b) = \frac{|\mathbf{m}_b^T \mathbf{F}_{pq} \mathbf{w}_a|}{\sqrt{l_1^2 + l_2^2}} < \varepsilon_2 \quad (8)$$

donde  $\mathbf{F}_{pq}$  es la matriz fundamental existente entre la imagen  $p$  y  $q$ , y  $[l_1 \ l_2 \ l_3]^T = \mathbf{F}_{pq} \mathbf{w}_a$ .

**Localización correcta en el espacio 3D:** *El punto 3D reconstruido, obtenido por triangulación a partir de los puntos  $w_a$  y  $w_b$ , debe encontrarse dentro del espacio 3D ocupado por el objeto de análisis.* Muchas veces se sabe a-priori dónde está ubicado en el espacio el objeto de análisis. Este espacio puede corresponder a un cubo, una esfera, un cilindro o bien un volumen más complejo almacenado como un modelo CAD. Si se cuenta con esta información, es posible entonces realizar una reconstrucción del punto 3D  $\mathbf{M}$  utilizando el método lineal citado en la sección 2.4 y verificar si las coordenadas de este punto pertenecen al volumen que ocupa el objeto de análisis.

Es posible que se cuente con información adicional para establecer de una manera más robusta el *matching* entre puntos. En la gran mayoría de casos no sólo hay información de coloración del punto y sus alrededores, sino también hay información de la forma de la región que contiene el punto. De esta manera es posible agregar un criterio de semejanza.

Definiendo: *i)*  $N$  como el número de imágenes en la secuencia; *ii)*  $N_1$  como el total de puntos de la secuencia; *iii)*  $t_i$  como el número de la imagen en la que se encuentra el punto  $i$  para  $i = 1, \dots, N_1$ ; y *iv)*  $e_j, f_j$  el número del primer y último punto detectado en la imagen  $j$  de la secuencia, para  $j = 1, \dots, N$ , donde  $e_1 = 1$ ,  $e_k = f_{k-1} + 1$  para  $k = 2, \dots, N$  y  $f_N = N_1$ ; la correspondencia en dos imágenes se determina evaluando los criterios mencionados en todos los puntos  $a$  y  $b$  de dos imágenes distintas  $p = t_a$  y  $q = t_b$  ( $p \neq q$ ) en  $m$  imágenes consecutivas de la secuencia, para  $p = 1, \dots, N - m$ ;  $q = p + 1, \dots, p + m$ ;  $a = e_p, \dots, f_p$  y  $b = e_q, \dots, f_q$ , como se muestra en la Figura 3. Si un punto no encuentra un *matching* con ningún otro punto en la secuencia será considerado como un punto que no tiene correspondencia y, por lo tanto, no podrá ser seguido en la secuencia. Estos puntos serán eliminados y no serán tomados en cuenta en los pasos posteriores.

Como se ilustra en la Figura 3, se define una matriz  $\mathbf{B}$  de  $N_2 \times 2$  elementos, cuya  $k$ -ésima fila  $[\mathbf{b}_{k1} \ \mathbf{b}_{k2}]$  significa un *matching* entre el punto  $\mathbf{b}_{k1}$  y el punto  $\mathbf{b}_{k2}$  (para  $k = 1, \dots, N_2$ ), donde  $N_2$  representa el número de pares de puntos que encontraron *matching* en la secuencia.

La consideración de  $m$  imágenes consecutivas en este análisis puede superar el problema de no segmentación de los puntos en cada una de las imágenes, ya que un punto que es detectado en la imagen  $p$  y no es detectado en la imagen  $p + 1$  puede igualmente ser seguido en la secuencia si es detectado en las imágenes  $p + 2, \dots, p + m$ . Después de este procedimiento muchos puntos que corresponden a ruido son eliminados, sin discriminar los puntos pertenecientes al objeto.

#### 3.2 Correspondencia en tres imágenes

En el seguimiento se determinan *trayectorias* de puntos correspondientes en la secuencia de imágenes. A partir de los *matchings* encontrados en dos imágenes se puede ahora buscar si existen trayectorias con más de dos puntos que sean proyecciones de un mismo punto 3D. Como se hizo en el procedimiento anterior,

**Algoritmo track-2:**

1.  $k = 0$
2. **for**  $p = 1, \dots, N - 1$
3.     **for**  $a = e_p, \dots, f_p$
4.         **for**  $q = p + 1, \dots, \min(p + m, N)$
5.             **for**  $b = e_q, \dots, f_q$
6.                 **if**  $\text{criterios}(w_a, w_b)$  **then**
7.                      $k = k + 1$
8.                      $B_k = [a \ b]$
9.  $N_2 = k$

Figura 3: Algoritmo para el *matching* en dos imágenes. La función  $\text{criterios}(w_a, w_b)$  es verdadera si se satisfacen los criterios de correspondencia para los puntos  $w_a$  y  $w_b$ .

**Algoritmo track-3:**

1.  $k = 0$
2. **for**  $i = 1, \dots, N_2 - 1$
3.     **for**  $j = i + 1, \dots, N_2$
4.         **if**  $b_{i2} = b_{j1}$  **then**
5.              $a = b_{j1}, \quad b = b_{j2}, \quad c = b_{i2}$
6.             **if**  $\|\hat{\mathbf{w}}_c - \mathbf{w}_c\| < \varepsilon_3$  **then**
7.                  $k = k + 1$
8.                  $C_k = [a \ b \ c]$
9.  $N_3 = k$ .

Figura 4: Algoritmo para el seguimiento en tres imágenes.

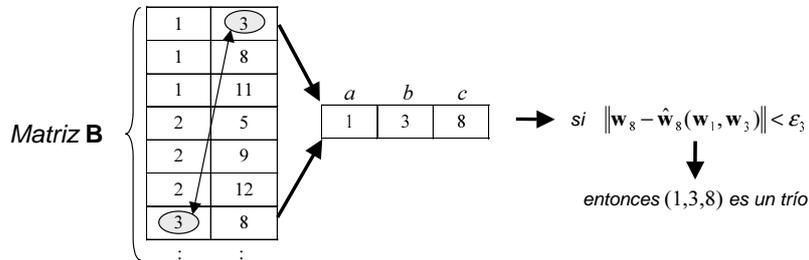


Figura 5: Ejemplo de seguimiento en tres imágenes.

aquellos puntos que no puedan ser seguidos en la secuencia serán considerados como ruido y por lo tanto serán eliminados.

A continuación se buscan todos los posibles enlaces de tres puntos en la matriz  $B$  (definida en la Sección 3.1) que cumplan la condición de correspondencia en tres imágenes. Se busca entonces las filas  $i$  y  $j$  de la matriz  $B$  (para  $i, j = 1, \dots, N_2$  y  $i \neq j$ ) que

$$b_{i2} = b_{j1}. \quad (9)$$

En el caso de que las filas  $i$  y  $j$  cumplan la condición (9), es decir si  $B_i = [a \ b]$  y  $B_j = [b \ c]$ , se establecerá entonces la correspondencia del *trío*  $(a, b, c)$  con coordenadas  $\mathbf{w}_a$ ,  $\mathbf{w}_b$  y  $\mathbf{w}_c$  en las imágenes  $p = t_a$ ,  $q = t_b$  y  $r = t_c$ , si la condición trifocal es verdadera, es decir si:

$$\|\hat{\mathbf{w}}_c - \mathbf{w}_c\| < \varepsilon_3, \quad (10)$$

donde  $\hat{\mathbf{w}}_c$  representa la *reproyección* de las coordenadas del tercer punto en la imagen  $r$ . La estimación de  $\hat{\mathbf{w}}_c$  se realiza a partir de  $\mathbf{w}_a$  y  $\mathbf{w}_b$  usando las trilinearidades de Shashua citadas en la sección 2.3. Para visualizar este problema se puede pensar que  $\hat{\mathbf{w}}_c$  es la intersección de las líneas epipolares de  $\mathbf{w}_a$  y  $\mathbf{w}_b$  en la tercera imagen. Esta forma de resolver el problema presenta sin embargo una singularidad cuando las

líneas epipolares son iguales. Shashua resolvió esta singularidad haciendo un análisis algebraico mediante las trilinearidades y los tensores trifocales [10].

Los tríos encontrados se agrupan en una nueva matriz de  $N_3 \times 3$  elementos:  $C = [c_{k1} \ c_{k2} \ c_{k3}]$ , para  $k = 1, \dots, N_3$ . El algoritmo para el seguimiento en tres imágenes es mostrado en la Figura 4. El lazo para  $j$  puede empezar a partir de  $i + 1$  ya que la matriz  $B$  ya está ordenada de menor a mayor. Un ejemplo del seguimiento en tres imágenes se ilustra en la Figura 5.

Nuevamente, los puntos que no pudieron ser agrupados en tríos son considerados ruido y son eliminados.

### 3.3 Correspondencia en cuatro imágenes

Para encontrar las trayectorias de cuatro puntos se buscan *cuartetos*, es decir cuatro puntos correspondientes (en cuatro imágenes distintas). Como la matriz  $C$  contiene en cada fila los tríos encontrados en la secuencia, se puede buscar en ella aquellas filas que contengan dos puntos en común (ver Figura 6). Si dos tríos contienen dos puntos en común, se puede afirmar que los cuatro puntos involucrados en este enlace también son correspondientes ya que todos ellos son proyecciones del mismo punto tridimensional. Como los elementos de las filas de  $C$  están ordenadas en forma ascendente ( $c_{i1} < c_{i2} < c_{i3}$ ), los cuartetos se encuentran simplemente buscando las filas  $i$  y  $j$  de  $C$  (para  $i, j = 1, \dots, N_3$  y  $i \neq j$ ) que cumplen

$$c_{i2} = c_{j1} \quad \text{y} \quad c_{i3} = c_{j2}. \quad (11)$$

Si se cumple (11) quiere decir que las dos filas tienen la forma  $C_i = [a \ b \ c]$  y  $C_j = [b \ c \ d]$ , encontrándose así el cuarteto  $(a, b, c, d)$ . Los cuartetos encontrados se agrupan en una nueva matriz de  $N_4 \times 4$  elementos:  $D = [d_{k1} \ d_{k2} \ d_{k3} \ d_{k4}]$ ,  $k = 1, \dots, N_4$ . El algoritmo es mostrado en la Figura 7.

Al igual que en los dos pasos anteriores, los puntos que no pudieron ser agrupados en cuartetos son considerados ruido y son eliminados.

### 3.4 Correspondencia en más imágenes

Si se desea establecer una correspondencia en más imágenes, se puede utilizar un razonamiento similar al de la correspondencia en cuatro imágenes. Es necesario tomar en cuenta que para encontrar una correspondencia en  $n$  imágenes se debe asegurar que los puntos a seguir deben haber sido segmentados inicialmente en por lo menos  $n$  imágenes de la secuencia, de lo contrario van a ser considerados ruido y serán eliminados.

Muchas veces la detección de un punto falla en alguna(s) imagen(es) de la secuencia. Sin embargo, teniendo sus puntos correspondientes en una trayectoria, es posible realizar –por medio del método de

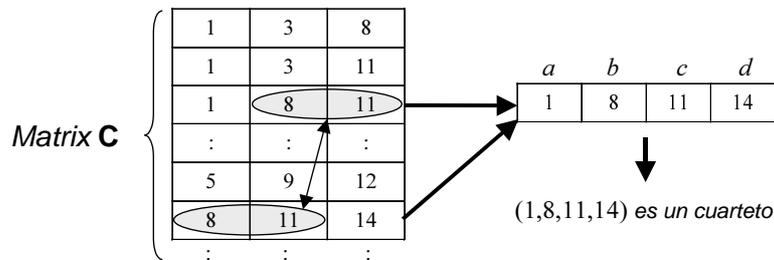


Figura 6: Ejemplo de seguimiento en cuatro imágenes.

```

Algoritmo track-4:
1.  $k = 0$ 
2. for  $i = 1, \dots, N_3 - 1$ 
3.   for  $j = i + 1, \dots, N_3$ 
4.     if  $(c_{i2} = c_{j1})$  and  $(c_{i3} = c_{j2})$  then
5.        $k = k + 1$ 
6.        $D_k = [a \ b \ c \ d]$ 
7.  $N_4 = k.$ 

```

Figura 7: Algoritmo para el seguimiento en cuatro imágenes.

mínimos cuadrados citado en la sección 2.4— una reconstrucción 3D del punto que produjo cada uno de estos puntos de la trayectoria. De esta manera, el punto 3D reconstruido puede ser reproyectado en aquella(s) imagen(es) donde la detección falló para obtener así una trayectoria completa en la secuencia.

## 4 El Algoritmo

A continuación se presenta una modificación al algoritmo original explicado en la sección anterior que acelera el algoritmo sin afectar su desempeño. El tiempo de ejecución puede reducirse si se disminuye el número de comparaciones que se debe hacer para establecer las correspondencias. La idea principal del nuevo algoritmo presentado en esta sección es reducir los lazos en **track-3** y **track-4** (ver Figuras 4 y 7 respectivamente) de tal forma que no se hagan preguntas innecesarias. Esto puede lograrse si se indexan los enlaces como se explicará a continuación.

Estudiando la eficiencia del algoritmo **track-3** en el ejemplo de la Figura 5, se puede observar que para encontrar el posible par correspondiente a (1, 3), es decir el par (3, 8), ha sido necesario hacer varias preguntas innecesarias, ya que el algoritmo pregunta por la correspondencia de (1, 3) con los pares (1, 8), (1, 11), (2, 5), (2, 9) y (2, 12) antes de evaluar el par (3, 8). Todas estas preguntas inútiles pueden evitarse si en un vector se tiene almacenado dónde están los pares que tienen como primer elemento el 3. Ésta es la idea principal de la modificación, incorporar índices que indiquen dónde se debe realizar la búsqueda.

La modificación del algoritmo **track-2** se muestra en la Figura 8. Se puede observar que la única modificación es la que tiene que ver con la incorporación de los vectores  $I$ ,  $J$  y  $X$ . En la búsqueda de pares correspondientes, no es posible disminuir el número de preguntas, ya que es necesario comparar todos los puntos de una imagen con todos los puntos de las siguientes  $m$  imágenes para encontrar pares correspondientes. La modificación mencionada se hace para acelerar la búsqueda de los puntos correspondientes en tres imágenes. En los vectores  $I$ ,  $J$  y  $X$  se almacena la información de dónde empieza en la tabla  $B$  un nuevo elemento  $b_{i1}$  y cuántas veces este elemento aparece en la tabla como primero. La forma como se almacena esta información es la siguiente. Si se desea averiguar por el punto número  $a$ , se lee primero el elemento  $a$  del vector  $X$ , i.e.  $g = X_a$ . Si  $g$  es cero quiere decir que no hay ningún elemento  $b_{i1} = a$ ,  $i = 1, \dots, N_2$ . De lo contrario  $I_g$  indica el número de fila de  $B$  en que  $a$  aparece por primera vez (en la primera columna). A su vez,  $J_g$  indica cuántas veces aparece  $a$  en la primera columna de  $B$ . Con esta información se sabe que

```

Algoritmo track-2N:
1.  $k = 0$ 
2.  $g = 0$ 
3.  $a_0 = 0$ 
4. for  $i = 1, \dots, N_1$ 
5.      $I_i = 0$ 
6.      $J_i = 0$ 
7.      $X_i = 0$ 
8. for  $p = 1, \dots, N - 1$ 
9.     for  $a = e_p, \dots, f_p$ 
10.        for  $q = p + 1, \dots, \min(p + m, N)$ 
11.           for  $b = e_q, \dots, f_q$ 
12.              if  criterios( $w_a, w_b$ )  then
13.                  $k = k + 1$ 
14.                  $B_k = [a \ b]$ 
15.                 if  $a \neq a_0$   then
16.                     $a_0 = a$ 
17.                     $g = g + 1$ 
18.                     $I_g = k$ 
19.                     $X_a = g$ 
20.                     $J_g = J_g + 1$ 
21.  $N_2 = k$ 

```

Figura 8: Algoritmo nuevo para el *matching* en dos imágenes. La función  **criterios**( $w_a, w_b$ ) es verdadera si se satisfacen los criterios de correspondencia para los puntos  $w_a$  y  $w_b$ .

**Algoritmo track-3N:**

```

1.  $k = 0$ 
2.  $g = 0$ 
3.  $a_0 = 0$ 
4. for  $i = 1, \dots, N_1$ 
5.      $I'_i = 0$ 
6.      $J'_i = 0$ 
7.      $X'_i = 0$ 
8. for  $i = 1, \dots, N_2 - 1$ 
9.      $b = b_{i2}$ 
10.    if  $X_b > 0$  then
11.        for  $j = I_{X_b}, \dots, I_{X_b} + J_{X_b} - 1$ 
12.             $a = b_{i1}, \quad c = b_{j2}$ 
13.            if  $\|\hat{\mathbf{w}}_c - \mathbf{w}_c\| < \varepsilon_3$  then
14.                 $k = k + 1$ 
15.                 $C_k = [a \ b \ c]$ 
16.                if  $a \neq a_0$  then
17.                     $a_0 = a$ 
18.                     $g = g + 1$ 
19.                     $I'_g = k$ 
20.                     $X'_a = g$ 
21.                     $J'_g = J'_g + 1$ 
22.  $N_3 = k$ .

```

Figura 9: Algoritmo nuevo para el seguimiento en tres imágenes.

**Algoritmo track-4N:**

```

1.  $k = 0$ 
2. for  $i = 1, \dots, N_3 - 1$ 
3.      $b = c_{i2}, \quad c = c_{i3}$ 
4.     if  $X'_b > 0$  then
5.          $a = c_{i1}$ 
6.         for  $j = I'_{X'_b}, \dots, I'_{X'_b} + J'_{X'_b} - 1$ 
7.             if  $c_{j2} = c$  then
8.                  $d = c_{j3}$ 
9.                  $k = k + 1$ 
10.                 $D_k = [a \ b \ c \ d]$ 
11.  $N_4 = k$ .

```

Figura 10: Algoritmo nuevo para el seguimiento en cuatro imágenes.

$b_{j,1}, b_{j+1,1}, \dots, b_{j+J_g-1,1}$  son iguales a  $a$ , siendo  $j = I_g$  y  $g = X_a$ , o bien  $j = I_{X_a}$ .

Con la información almacenada en  $I$ ,  $J$  y  $X$  se acelera considerablemente la búsqueda de tres puntos correspondientes. La nueva versión del algoritmo **track-3** se muestra en la Figura 9. Se aprecia que el lazo **for** para la variable  $j$  (línea 11) ha sido mejorado ya que no es desde  $i + 1$  hasta  $N_2$  como era originalmente (ver línea 3 en Figura 4) sino que sólo desde  $I_{X_b}$  hasta  $I_{X_b} + J_{X_b} - 1$ .

En este mismo algoritmo se observa la incorporación de las variables  $I'$ ,  $J'$  y  $X'$ . Estos vectores se construyen de la misma manera que los vectores  $I$ ,  $J$  y  $X$  mencionados para el algoritmo **track-2N**. En los vectores se almacena la información de dónde empieza en la tabla  $C$  un nuevo elemento  $c_{i1}$  y cuántas veces este elemento aparece en la tabla como primero. Se podría tener de manera indexada dónde en la tabla  $C$  aparece un nuevo par de elementos (en vez de tener dónde empieza un elemento único), sin embargo esta alternativa consume demasiados recursos computacionales ya que sería necesario tener índices en variables

matriciales. Para el caso de secuencias con muchos puntos, el consumo de memoria puede llegar a ser crítico. Por esta razón se recomienda el uso de la primera alternativa.

De la misma manera se modifica el algoritmo para buscar puntos correspondientes en cuatro imágenes. El nuevo algoritmo, llamado `track-4N`, se muestra en la Figura 10. Se aprecia la modificación en el lazo `for` para la variable  $j$  (línea 6) (comparar con línea 3 del algoritmo original `track-4` en Figura 7).

## 5 Resultados Experimentales

En esta sección se presentan algunos resultados experimentales obtenidos recientemente. Para la evaluación del algoritmo desarrollado se implementó en MATLAB un programa que simula puntos de manera aleatoria en una secuencia de  $N$  imágenes binarias de  $400 \times 400$  píxeles. A su vez, se cuenta con un modelo 3D de un cubo que es movido de imagen a imagen. Los vértices del cubo son proyectados en cada una de las imágenes. A manera de ejemplo la Figura 11 muestra una de estas imágenes de la secuencia, en la que los 8 puntos del cubo se mezclan con  $N_p = 500$  puntos colocados aleatoriamente. La secuencia completa de 10 imágenes se presenta en la Figura 12a (la Figura 11 corresponde a la quinta imagen de la secuencia), donde los puntos del cubo se han marcado con círculos con el fin de que el/la lector/a pueda distinguirlos. Cabe señalar que los puntos  $(u_i, v_i)$  proyectados de los vértices del cubo fueron alterados con ruido de distribución normal, de tal forma que la ubicación de los puntos generados presenta una desviación de la proyección ideal:

$$u_i := u_i + e_u \quad v_i := v_i + e_v \quad (12)$$

donde  $e_u$  y  $e_v$  son variables aleatorias de media cero y desviación estándar  $\sigma$ , esto significa que la desviación estándar de la variable que mide la distancia entre los puntos reales e ideales es  $\sqrt{2}\sigma$ . En el caso mostrado en la Figura 12 se escogió  $\sigma = 0.5$  [píxel].

Luego de evaluar la correspondencia en dos imágenes se obtienen los resultados de la Figura 12b. En esta representación los puntos que no encontraron correspondencia son considerados ruido y por esta razón se eliminan. En esta evaluación no se utilizó ningún criterio de semejanza, es decir sólo se consideró la condición epipolar y la localización correcta en el espacio 3D. Para esta última condición se consideró que el objeto 3D de análisis estaba en un espacio cúbico de dimensiones  $8^3$  siendo el objeto de análisis un cubo de dimensiones  $2^3$  en el centro del cubo mayor. Esto quiere decir que el espacio 3D correcto es 64 veces más grande que el objeto de análisis. Por esta razón, los puntos que han encontrado una correspondencia se ubican en un área aproximada de  $4h \times 4h$ , siendo  $h$  la longitud de un lado del cubo proyectado.

El análisis se repite para 3 y para 4 imágenes (ver Figuras 12c y 12d respectivamente). Al final se aprecia que todos los puntos correspondientes a ruido fueron eliminados sin discriminar los puntos correspondientes

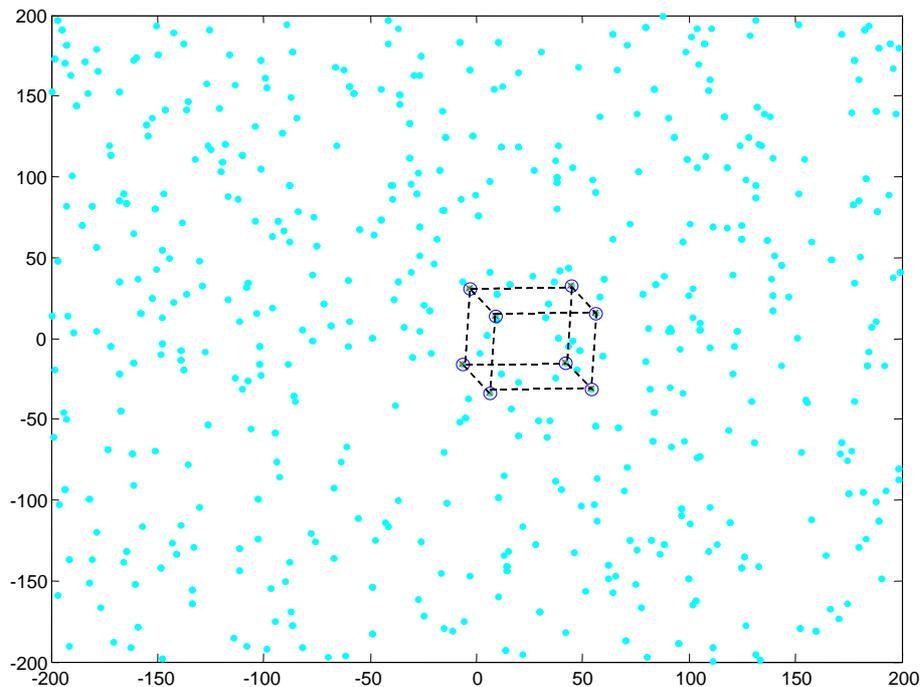


Figura 11: Simulación de la proyección de los vértices de un cubo en un conjunto de 500 puntos aleatorios.

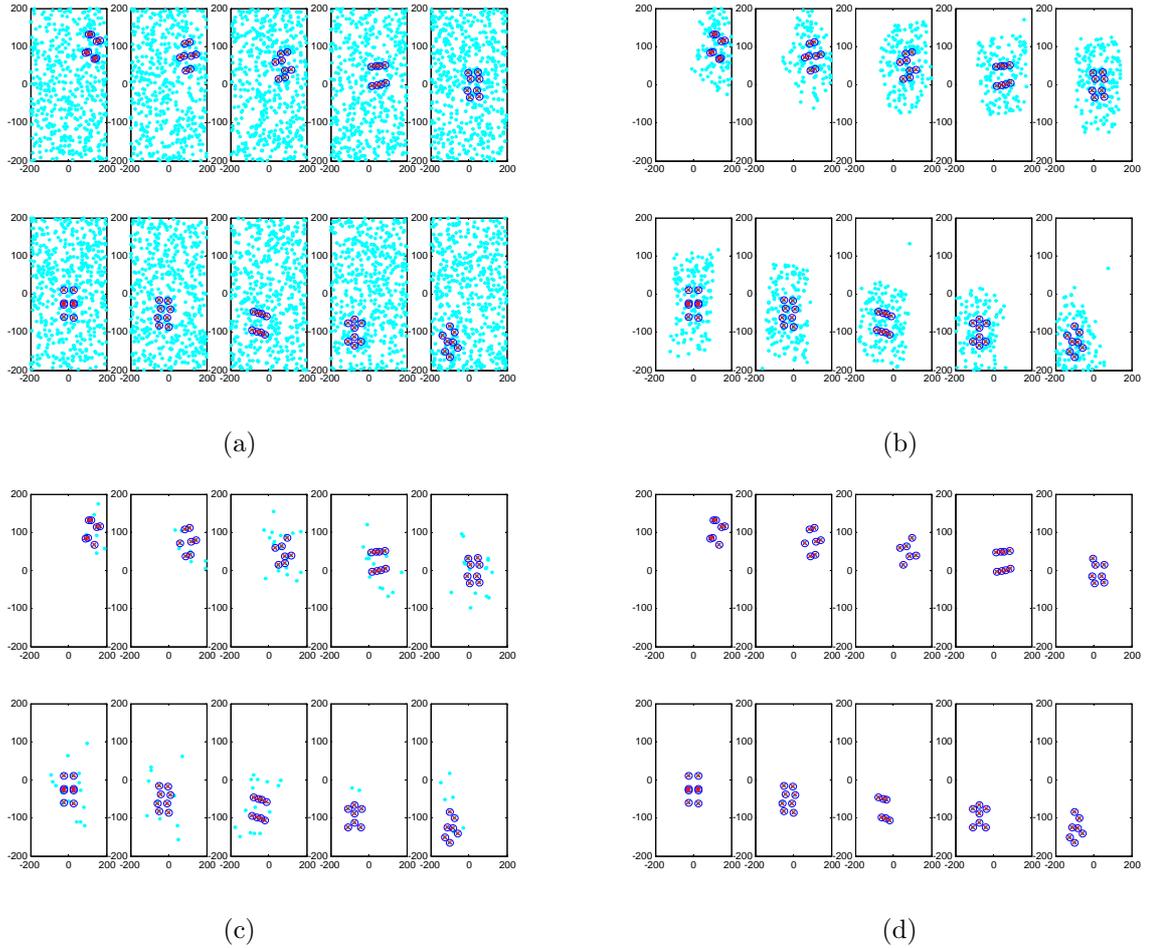


Figura 12: Seguimiento de vértices de un cubo con 500 puntos de ruido por imagen en una secuencia de 10 imágenes: a) Secuencia original y puntos que han pasado la prueba del seguimiento en b) dos, c) tres y d) cuatro imágenes.

a los vértices del cubo. En ciertos experimentos fueron eliminados puntos del cubo en algunas imágenes, sin embargo debido a que esos puntos no fueron eliminados en las otras imágenes de la secuencia y se llevó a cabo su seguimiento en cuatro imágenes, su detección fue posible. Como se mencionó en la sección 3.4, no hay que olvidar que es posible reconstruir los puntos no detectados (o eliminados) a través de la reproyección del punto 3D reconstruido.

El desempeño de este método fue evaluado en secuencias de 10 imágenes en las que fueron proyectados los 8 puntos del cubo 3D. Se hizo variar por una parte el número  $N_p$  de puntos aleatorios presentes en cada imagen y por otra parte la magnitud de la desviación estándar  $\sigma$ . En todos los experimentos se mantuvo  $\varepsilon_2 = \varepsilon_3 = 1$  [píxel] necesarios para evaluar las restricciones en dos y tres vistas (ver ecuaciones (8) y (10) respectivamente). Los resultados se muestran en la Figura 13. Se observa que mientras más pequeño sea  $\sigma$  mejor es el seguimiento de los puntos. Por otra parte, si el ruido aumenta ( $N_p$  grande) también aumenta el número de falsas alarmas. La restricción  $\varepsilon_2 = \varepsilon_3 = 1$  indica que los puntos correspondientes que estén fuera del radio de 1 píxel no podrán ser considerados como tales. Por otra parte, si el ruido es grande aumenta la probabilidad de que se generen, de manera aleatoria, pares, tríos y cuartetos correspondientes.

Adicionalmente se midió el tiempo de ejecución de los algoritmos originales y los modificados para imágenes de 10 secuencias haciendo variar el número  $N_p$  de puntos aleatorios por imagen. En estos experimentos el universo de puntos detectados es  $N_1 = 10(N_p + 8)$  (10 imágenes en las que hay  $N_p$  puntos de ruido y 8 puntos del cubo). Los resultados se muestran en la Figura 14. Se aprecia que los algoritmos **track-2** y **track-2N** no presentan mayores diferencias en el tiempo de ejecución, lo que no ocurre en **track-3** y **track-3N**, y **track-4** y **track-4N**, ya que para  $N_p = 75, 500, 1000, 2000$  si **track-3** demora 100 unidades de tiempo, **track-3N** necesita 44.5, 5.9, 3.0 y 1.5 unidades respectivamente. Análogamente, si **track-4** demora 100, **track-4N** necesita sólo 9.3, 3.0, 0.9 y 0.2 respectivamente. Sin embargo, debido a que el mayor tiempo se consume en el análisis de dos vistas (donde no es posible mejorar el algoritmo), la reducción

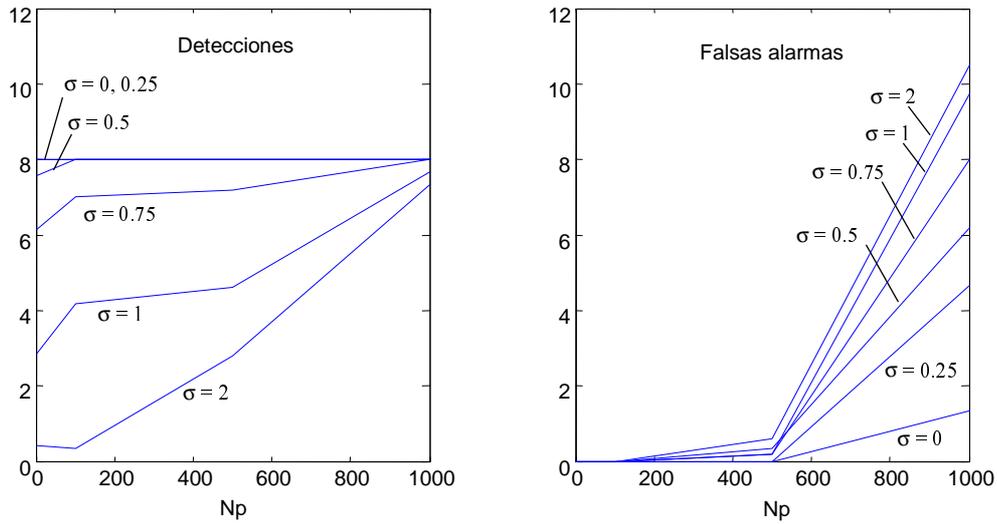


Figura 13: Evaluación del desempeño del método después de un seguimiento en cuatro imágenes: a) Número de puntos detectados del cubo (ideal = 8); (b) número de falsas alarmas (ideal = 0).

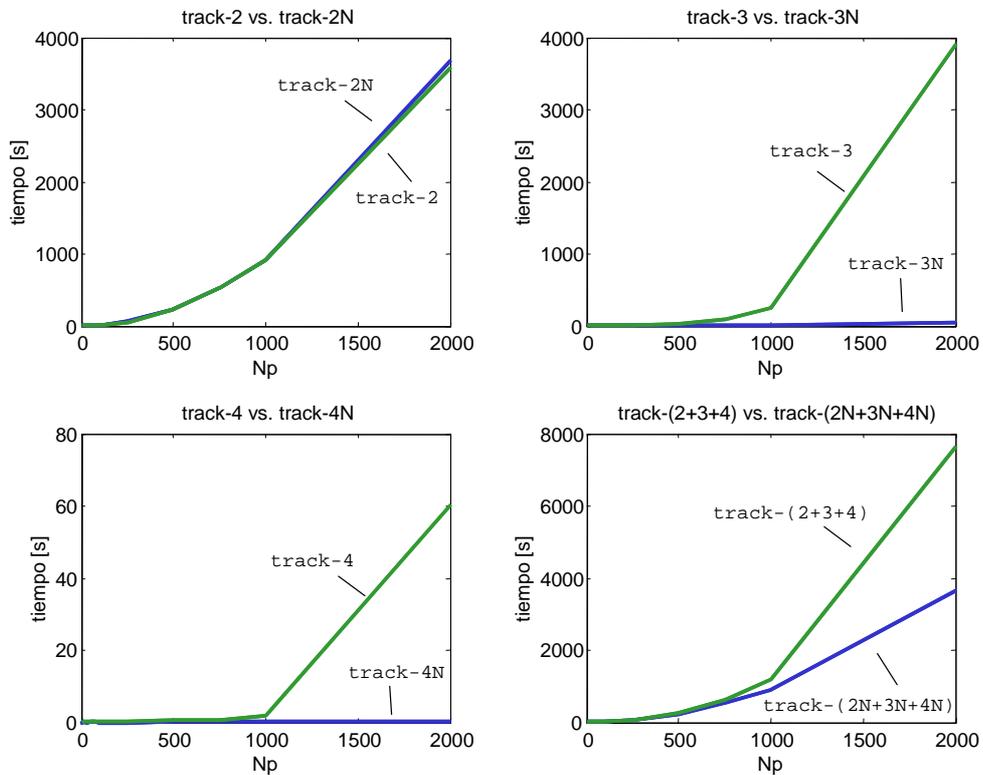


Figura 14: Tiempo de ejecución de algoritmos track-2, track-2N, track-3, track-3N, track-4, track-4N, track-(2+3+4) y track-(2N+3N+4N) versus  $N_p$ .

no es tan significativa: si el algoritmo original demora 100 el nuevo necesita 96.3, 89.9, 78.5 y 47.5 para  $N_p = 75, 500, 1000, 2000$  respectivamente. Los experimentos se realizaron en MATLAB en un computador Intel Pentium 4, con una CPU de 1.500 MHz y 256 kB RAM.

## 6 Conclusiones

En este artículo se han presentado dos resultados interesantes: por una parte se ha logrado mejorar un algoritmo capaz de llevar a cabo un seguimiento de puntos en una secuencia de imágenes ruidosa; y por otra parte se ha hecho una evaluación preliminar del desempeño de este método en la discriminación de ruido.

El algoritmo se ha mejorado indexando las tablas de correspondencia en dos y tres imágenes. De esta manera la búsqueda de tríos y cuartetos correspondientes en estas tablas es más eficiente, ya que la búsqueda se realiza previamente en el momento de construir los índices. Si bien es cierto que el tiempo de ejecución para la búsqueda de tríos y cuartetos se redujo significativamente, es necesario recalcar que la reducción total del tiempo de ejecución sólo es considerable para imágenes muy ruidosas, pues la búsqueda de pares, que es la que más tiempo consume, no es posible de reducir.

La evaluación del desempeño del método se hizo utilizando un programa para simular la proyección de puntos de un objeto 3D superpuestos con puntos colocados aleatoriamente. Los resultados obtenidos indican que este método puede hacer un perfecto seguimiento de puntos en una secuencia de imágenes altamente ruidosas siempre y cuando la calibración del sistema de visión sea muy precisa ( $\sigma$  bajo).

Es necesario evaluar el comportamiento de este método ante otras situaciones tales como disminución de número de imágenes de la secuencia, desplazamiento del objeto 3D de proyección a proyección y sensibilidad del método ante la no segmentación inicial de los puntos en una o varias imágenes de la secuencia.

## Agradecimientos

Los autores agradecen el apoyo brindado por el Departamento de Ingeniería Informática y el Departamento de Investigaciones Científicas y Tecnológicas de la Universidad de Santiago de Chile.

## Referencias

- [1] K.R. Castleman. *Digital Image Processing*. Prentice-Hall, Englewood Cliffs, New Jersey, 1996.
- [2] O. Faugeras. *Three-Dimensional Computer Vision: A Geometric Viewpoint*. The MIT Press, Cambridge MA, London, 1993.
- [3] O. Faugeras, Q.-T. Luong, and T. Papadopoulos. *The Geometry of Multiple Images: The Laws That Govern the Formation of Multiple Images of a Scene and Some of Their Applications*. The MIT Press, Cambridge MA, London, 2001.
- [4] R. Hartley. A linear method for reconstruction from lines and points. In *5th International Conference on Computer Vision (ICCV-95)*, pages 882–887, Cambridge, MA, 1995.
- [5] R. Hartley. Lines and points in three views and the trifocal tensor. *International Journal of Computer Vision*, 22(2):125–150, 1997.
- [6] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, 2000.
- [7] D. Mery. *Automated Flaw Detection in Castings from Digital Radioscopic Image Sequences*. Verlag Dr. Köster, Berlin, 2001. (Ph.D. Thesis in German).
- [8] D. Mery and D. Filbert. Automated flaw detection in aluminum castings based on the tracking of potential defects in a radioscopic image sequence. *IEEE Trans. Robotics and Automation*, December 2002. (In Press).
- [9] A. Shashua. Algebraic functions for recognition. *IEEE Trans. on Pattern Analysis and Machine Intelligence PAMI*, 17(8):779–789, 1995.
- [10] A. Shashua. Trilinear tensor: The fundamental construct of multiple-view geometry and its applications. In *International Workshop on Algebraic Frames For The Perception Action Cycle (AFPAC)*, Kiel, Sep. 8-9 1997.