XXV Jornadas de Automática
Ciudad Real, del 8 al 10 de septiembre de 2004

COMPENSACIÓN DEL MOVIMIENTO EN SECUENCIAS DE IMÁGENES AÉREAS

Fernando Caballero Benítez, Joaquín Ferruz Melero, Aníbal Ollero Baturone
Departamento de Ingeniería de Sistemas y Automática
Universidad de Sevilla
caba@cartuja.us.es ferruz@cartuja.us.es aollero@cartuja.us.es

Resumen

El tratamiento digital de secuencias de imágenes aéreas resulta de gran complejidad debido a las vibraciones y movimientos inducidos por el vehículo. En este artículo se describe un procedimiento para anular el movimiento de las imágenes de forma digital y poder aplicar sobre las secuencias de imágenes algoritmos convencionales de detección, seguimiento o de cualquier otra índole. Se describirá el fundamento matemático del procedimiento y diferentes optimizaciones que permitan mejorar el resultado del mismo.

Palabras Clave: Compensación del movimiento, homografía, imagen, outlier.

1 INTRODUCCIÓN

En términos generales, el tratamiento digital de imágenes para su posterior estudio y extracción de datos se facilita notablemente si la escena que se observa se mantiene fija, ya que las referencias de posición permanecen inalterables a lo largo del proceso de análisis. Cuando dicho proceso se realiza sobre secuencias de imágenes aéreas se encuentra el inconveniente de introducir en ellas efectos debidos a las turbulencias y a las propias vibraciones del vehículo aéreo.

El análisis de imágenes tomadas desde helicóptero puede facilitar datos que las tomas fijas en tierra no pueden captar, ya que en vuelo podemos obtener mayor variedad de ángulos de visión y movimiento. Las imágenes aéreas se utilizan en [4] y [5] donde se proponen sistemas para la monitorización de incendios basada en visión por computador y UAVs (vehículos autónomos no tripulados)

La visión por computador aplicada a vehículos aéreos puede llegar a ser especialmente útil para el control; así en [9] se utiliza para conseguir aterrizajes seguros, y en [12] se para el aterrizaje de un helicóptero sobre un objetivo que se mueve lentamente. La estimación de movimiento relativa al punto de aterrizaje y el aterrizaje de vehículos basado en visión se investigan también en [7] y [11]. El procesamiento digital de imágenes para vehículos autónomos se estudia también en el proyecto WITAS [3].

Para aprovechar las múltiples posibilidades que ofrece el proceso digital de imágenes tomadas desde vehículos aéreos es necesario solucionar el problema intrínseco de las vibraciones existentes las mismas. Una posible mejora podría ser la corrección de la posición de las imágenes de forma digital, en un ordenador. Así podríamos usar las grabaciones tomadas desde el aire, sin el inconveniente que ello conlleva.

En este artículo se describen los pasos a seguir para poder conseguir la eliminación del movimiento entre imágenes de una misma secuencia de forma íntegramente digital. En el mercado existen diferentes sistemas electromecánicos que permiten reducir notablemente el movimiento de la cámara actuando sobre ésta. Este tipo de sistemas resultan caros y pesados, por tanto, el hecho de necesitar tan solo un ordenador donde ejecutar el algoritmo, permite importantes reducciones de costo, y la posibilidad de utilizar vehículos con poca capacidad de carga.

2 DESCRIPCIÓN DEL PROBLEMA

Así pues, se pretende el análisis de la compensación del movimiento en secuencias de imágenes tomadas desde helicóptero. Analizando el problema, los pasos a seguir para conseguir nuestro objetivo deberán ser los siguientes:

- Cálculo del movimiento que se ha producido de la imagen anterior a la actual.
- Formalización matemática de dicho movimiento y filtrado de posibles errores mediante un modelo.
- Aplicación del modelo matemático sobre los píxeles de la imagen actual para compensar el movimiento respecto a la imagen anterior.

El método que se utilizará para poder deducir el movimiento que se ha producido de una imagen a la siguiente se describe en [1] y [2]. Tiene como objetivo determinar las correspondencias de regiones automáticamente seleccionadas, como se muestra en la figura 1. Combina técnicas de correlación con la formación de agrupaciones, y utiliza un algoritmo predictivo para mejorar la eficiencia del proceso de correspondencia.

Si el numero de correspondencias es el adecuado al tamaño de la imagen y están lo suficientemente distribuidas a lo largo de la misma, podrá deducirse el movimiento aparente que se ha producido en toda a la imagen a partir del análisis del movimiento conocido de las regiones seleccionadas.

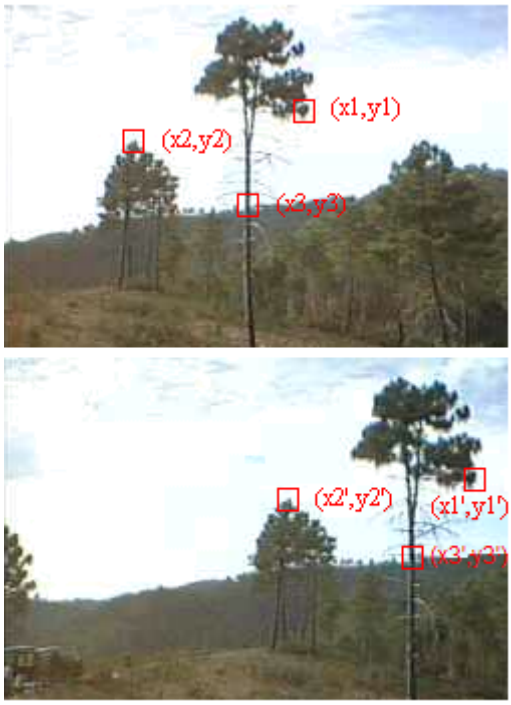

Figura 1: Ejemplo de búsqueda de correspondencias

El modelo que usaremos para representar el movimiento entre una imagen y la siguiente será la homografía, que permite describir la transformación de la imagen de un plano cuando se modifica la posición u orientación de la cámara.

En esta situación resulta vital que la escena que se esta analizando no posea demasiados objetos en movimiento, ya que puede enmascarar la componente fundamental debida a desplazamiento o vibración del vehículo. Será necesario, por tanto, algún tipo de algoritmo que permita establecer cuales de las correspondencias obtenidas pertenecen a objetos que se mueven de forma independiente a la escena. Para rechazar el movimiento independiente se utilizará el algoritmo de "mínima mediana de los residuos al cuadrados" descrito por Zhang en [13].

Finalmente, conocido el movimiento de la perturbación, bastará con aplicar la nueva posición a cada uno de los píxeles actuales, compensando las perturbaciones. Posteriormente se describirán métodos que permiten minimizar en lo posible los tiempos de cómputo de este proceso, el cual, a pesar de ser sencillo, puede llegar a suponer una carga computacional no despreciable.

3 CÁLCULO DE HOMOGRAFÍAS

A partir de las correspondencias obtenidas del algoritmo de seguimiento de ventanas, es necesario ajustar un modelo al movimiento a las mismas, minimizando el error. El modelo utilizado será la homografía. Si la escena 3D que recogen las cámaras es un plano, al tomar dos imágenes I e I', los puntos (x,y) y (x',y') correspondientes a I e I' vendrán relacionados mediante la siguiente ecuación:

$$k\tilde{m}' = H\tilde{m} \qquad (1)$$

$$\tilde{m} = \begin{bmatrix} x \\ y \\ 1 \end{bmatrix} \qquad \tilde{m}' = \begin{bmatrix} x' \\ y' \\ 1 \end{bmatrix}$$

Donde H es un matriz no singular de orden 3x3 definida salvo un factor de escala, y por tanto depende solo de 8 parámetros. Debido a que cada correspondencia aporta dos ecuaciones, serán necesarias 4 correspondencias para poder resolver el sistema y deducir el valor de H.

En [6], Faugueras cita las características de la matriz de homografía, la cual permitir describir rotaciones, traslaciones, escalados e incluso pequeñas deformaciones del plano que se está tratando, quedando demostrado su correcto funcionamiento en numerosos artículos como [8] y [10]. Se supone por tanto que la escena 3D se puede asemejar por un plan; atendiendo a los resultados experimentales y teniendo en cuenta que el vehículo aéreo está localizado a una altura considerable, dicha suposición resulta correcta. También será válido el planteamiento cuando el movimiento consiste esencialmente en rotación pura, como es el caso de un helicóptero en vuelo estacionario.

Como se indicó anteriormente, es necesario establecer algún método que nos permita discernir entre las correspondencias asociadas a objetos en movimiento y correspondencias asociadas a la escena fija, la que se quiere estabilizar. El método de la mínima mediana de los residuos al cuadrados estima

qué correspondencias son correctas y cuales de ellas son "outliers" resolviendo el siguiente problema de minimización:

$$\min\left[\underset{i}{med}\left(r_i^2\right)\right] \qquad (2)$$

Es decir, es necesario determinar el valor mínimo de las medianas de los residuos al cuadrado, aplicado a todo el conjunto de datos. Éste es un método muy robusto para la detección de correspondencias erróneas y, por tanto, para la detección de "outliers" en general.

El valor de la mínima mediana lo podemos calcular mediante la búsqueda de la mejor estimación de la matriz de homografía en el espacio de los datos. Debido a que analizar todas las posibilidades existentes sería demasiado costoso, es necesario aplicar algún método aleatorio de selección de datos.

El algoritmo de LMedS se puede describir de la siguiente manera. Partimos de un conjunto de n correspondencias $m_i = (x_i \quad y_i)$:

- En primer lugar se debe utilizar algún algoritmo del tipo Montecarlo para la selección de m conjuntos de 4 elementos del espacio de datos. Así, hemos supuesto que la probabilidad de que una correspondencia esté en una posición determinada de la imagen se rige por una distribución uniforme, seleccionando estas de forma aleatoria. El número de elementos seleccionados es 4 debido a que es el número mínimo de correspondencias necesarias para determinar la matriz de homografía.
- Para cada uno de los conjuntos de 4 elementos, indexados mediante j, calcularemos la matriz de homografía H_j que mejor se ajusta a éstos.
- Para cada H_j debe calcularse la mediana de los residuos al cuadrado, M_j, respecto a los valores que indica la correspondencia.

$$M_j = med\left[\sum_{i=1}^{n} r_i^2(H_j, m_i)\right] \qquad (3)$$

- Para calcular el valor de r_i pueden usarse diferentes técnicas. En la solución adoptada se utiliza la distancia euclídea entre la posición calculada con la matriz H_j y el valor de la correspondencia.
- Se selecciona el valor mínimo del conjunto de valores de M_j. A este mínimo se le denominará M_J.

Por último tan solo nos queda calcular la varianza de los "outliers" a partir de la mínima mediana M_J obtenida en el proceso anterior. Asumiendo que entre todo el conjunto de correspondencias existe un conjunto E de outliers, la probabilidad de que al menos uno de los m conjuntos sea correcto será:

$$Pb = 1 - \left[1 - (1 - E)^4\right]^m \qquad (4)$$

Imponiendo que el valor de *Pb* debe ser cercano a uno se obtiene:

$$m = \frac{\log(1 - Pb)}{\log(1 - (1 - E)^p)} \qquad (5)$$

De forma experimental se llega a que la varianza de los datos se puede expresar como sigue:

$$\sigma = 1.4826\left(1 + \frac{5}{n - p}\right)\sqrt{M_J} \qquad (6)$$

Conocida la varianza del espacio de datos, podemos utilizarla para detectar cuáles son outliers, de modo que si la distancia r_i^2 es mayor que $2.5\sigma^2$ el dato será un outlier, siendo un dato correcto en caso contrario.

Es necesario tener en cuenta que al aplicar el algoritmo de LMedS sobre conjuntos de datos que se encuentran próximos en la imagen puede provocar problemas de convergencia. Para evitar esta circunstancia, Zhang aconseja el uso de lo que denomina rejilla.

El uso de la rejilla no es más que la división del espacio de datos en un conjunto de cuadrados o cubos. Con esto se pretende que cuando se tome alguno de los datos de un cuadrado, no se vuelva a tomar ningún otro más del mismo. Así se asegura que los datos que se toman en cada iteración del LMedS están lo suficientemente alejados en el espacio como para no generar problemas de convergencia al obtener el modelo.

El número de cuadrados de la rejilla será de vital importancia. Debe tener en cuenta lo siguiente para elegir las características de la rejilla:

- Cada vez que se toma una correspondencia de un cuadrado de la rejilla no se puede volver a tomar ninguna correspondencia del

mismo hasta la siguiente iteración del algoritmo de LMedS.

- El número de correspondencias mínimo para poder obtener la matriz de homografía es cuatro.

Así, la primera aproximación a la solución será la selección de una rejilla con cuatro elementos, dividiendo la imagen en cuatro zonas. Aparentemente, esta solución resulta la mejor, pero no cuenta con el hecho de que en iteraciones siguientes se vuelvan a coger los mismos puntos o puntos cercanos a los que se tomaron en la iteración anterior. Es necesario evitar esta circunstancia, ya que la desviación típica conseguida para discriminar los "outliers" debe calcularse a partir de la mayor cantidad de correspondencias posible.

Con la intención de evitar el problema anterior se tomará una rejilla de 8 elementos, de modo que el proceso de catalogación de las correspondencias no sea demasiado costoso y al mismo tiempo mejore el funcionamiento del algoritmo de LMedS. Así, el algoritmo de selección de correspondencias será el siguiente:

- Subdivisión de la imagen en 8 zonas del mismo tamaño.
- En cada iteración se toman 4 correspondencias, cada una de las cuales pertenece a un cuadrado diferente, anulando este para la iteración siguiente. Si cuando se toma una correspondencia de la rejilla no quedan cuadrados libres, se liberan todos y se toma una correspondencia de cualquiera de ellos.

4 COMPENSACIÓN DEL MOVIMIENTO

Una vez obtenida la matriz de homografía puede considerarse definido el movimiento aparente de una a otra imagen. Conocido dicho movimiento, el objetivo es transformar la imagen actual de forma que lo contrarreste.

La matriz de homografía nos indica a qué posición debe ir cada uno de los píxeles de la imagen anterior para que se pueda compensar el movimiento que los ha hecho ocupar posiciones diferentes a las actuales.

Suponiendo que la matriz de homografía no es afín, la forma general de obtener la nueva posición de los píxeles sería:

$$\begin{aligned} x' &= (x * H[0] + y * H[1] + H[2])/k \\ y' &= (x * H[3] + y * H[4] + H[5])/k \\ k &= x * H[6] + y * H[7] + H[8] \end{aligned} \quad (7)$$

$$H = \begin{bmatrix} H[0] & H[1] & H[2] \\ H[3] & H[4] & H[5] \\ H[6] & H[7] & 1 \end{bmatrix}$$

Cada una de estas transformaciones generará una posición (x',y') a la que se desplazará el píxel (x,y). Pero dichas coordenadas, al haberse obtenido tras operaciones algebraicas, no tienen por qué ser números enteros, de modo que se plantea la cuestión de qué píxel es mejor asignar a dicha posición. Será necesario usar algún método para tomar esta decisión. En el sistema que se describe se han utilizado la aproximación bilineal y la del "píxel más cercano".

Haciendo recuento del número de operaciones necesarias para hacer la transformación sobre los píxeles se obtiene un total de 6 multiplicaciones, 6 sumas y 2 divisiones. Si se toma como referencia una imagen de tamaño 384x287 píxeles, el número de operaciones asciende a 661248 sumas, 661248 multiplicaciones y 220416 divisiones. Resulta evidente que la carga computacional es bastante elevada, sobre todo por el número de divisiones, ya que los microprocesadores comerciales no están preparados para hacer esta operación en un solo ciclo de hardware, y un número de ciclos considerable para poder implementar el algoritmo que resuelva la operación.

Tiene sentido, por tanto, estudiar el proceso de compensación del movimiento con la intención de reducir el procesamiento en lo posible. Así, a la hora de buscar posibles métodos matemáticos pueden diferenciar dos casos:

- Matriz de homografía afín. En este caso, la matriz de homografía posee la siguiente forma:

$$H = \begin{bmatrix} H[0] & H[1] & H[2] \\ H[3] & H[4] & H[5] \\ 0 & 0 & 1 \end{bmatrix}$$

  Pueden simplificarse las ecuaciones, llegando a lo siguiente:

$$\begin{aligned} x' &= x * H[0] + y * H[1] + H[2] \\ y' &= x * H[3] + y * H[4] + H[5] \end{aligned} \quad (8)$$

Puede observarse cómo en este caso particular disminuye notablemente el número de operaciones, pues sólo son son necesarias 4 multiplicaciones y 4 sumas. No son necesarias las divisiones, que resultan especialmente costosas, y además se reduce en 2 el número de multiplicaciones y sumas por cada píxel.

- Matriz de homografía no afín. En este caso no es posible hacer ningún tipo de simplificación sobre las ecuaciones, de modo que el procesamiento es el mismo.

A pesar de todo, en función de las circunstancias, se podría hacer una aproximación. Si los valores de H[6] y H[7] son cercanos a cero puede suponerse que la matriz de homografía es afín y por tanto aplicar las simplificaciones anteriores. El problema es saber cual es la cota a partir de la cual se puede considerar que estos dos elementos de la matriz son nulos. Para poder saberlo se han hecho una serie de experimentos analizando el distanciamiento entre la H aproximada y la H original. De los experimentos se ha establecido que, si la matriz de homografía esta normalizada respecto de H[8], es decir, si H[8]=1, y H[6] y H[7] son menores que 10-4, la matriz de homografía se puede aproximar por una matriz afín, pudiéndose aplicar las reducciones anteriores..

De este modo se consiguen extrapolar las simplificaciones de las matrices afines para determinadas matrices de homografía. La simplificación se puede extender a matrices de homografía arbitrarias si se aproxima la transformación de forma lineal. La aproximación lineal consiste en calcular la transformación del primer y último píxel de cada fila, obteniéndose la recta que va desde la posición inicial a la final de x' e y'. Calculada dicha recta, la posición x' e y' del resto de píxeles intermedios de la fila se calculará sustituyendo en las ecuaciones halladas para cada fila. Así las operaciones necesarias para cada píxel serán:

$$\begin{aligned} x' &= ax + b \\ y' &= cy + d \end{aligned} \qquad (9)$$

Suponiendo el ancho de la imagen lo suficientemente elevado podrá despreciarse el coste de realizar los cálculos para la obtención de las rectas por cada fila, de modo que las operaciones matemáticas por píxel serían 2 multiplicaciones y 2 sumas. Así, en una imagen patrón de 384x287 píxeles, el número total de operaciones sería de 220416 multiplicaciones y 220416 sumas. Puede observarse cómo el número de operaciones por píxel es inferior a los necesarios usando una transformación afín. Debido a que la transformación no tiene porque ser lineal, debe hacerse una generalización del método haciendo una linealización por tramos de la transformación, utilizando el número suficiente de tramos para no generar errores demasiado apreciables. Así, en función de la no linealidad de la transformación se crearan mayor o menor cantidad de tramos linealizados. Debe comenzarse por caracterizar de alguna manera la no linealidad de la transformación de los píxeles; así, si la matriz de homografía está escalada respecto al coeficiente H[8], podemos representar la no linealidad de la transformación desde el píxel (x,y) al (x',y') como una función NL(x,y) a partir de las ecuaciones siguientes:

$$\begin{aligned} x' &= (x * H[0] + y * H[1] + H[2])/(1 + NL(x,y)) \\ y' &= (x * H[3] + y * H[4] + H[5])/(1 + NL(x,y)) \\ NL(x,y) &= x * H[6] + y * H[7] \end{aligned} \qquad (10)$$

Por tanto, la no linealidad depende de la posición de (x,y) del píxel sobre el cual se realiza la transformación. En términos generales, la menor no-linealidad de la transformación se dará cuando los coeficientes multiplicativos H[6] y H[7] son de un valor muy pequeño.

El siguiente problema es obtener una función que permita saber cual es el número de tramos necesario. Para conseguir esto se utiliza una ecuación obtenida de forma heurística que en función de la no linealidad de la transformación permite calcular el número de tramos necesarios para lienalizarla. Como parámetro, esta ecuación recibe el valor máximo que toma el modulo de NL(x,y), al cual se denominará $|NL|_{\max}$, a lo largo de la transformación. Debido a que NL(x,y) es un plano, los valores máximos deben estar en las esquinas de este, de modo que para obtener el máximo de NL(x,y) a lo largo de la transformación basta con calcular el valor de NL(x,y) en (0,0), (ancho,0), (alto,0) y (ancho, alto) y seleccionar con el que tiene mayor módulo. Así, la ecuación vendrá dada por lo siguiente:

$$Segments = ceil(5.68 * |NL|_{\max}^{0.5024}) \qquad (11)$$

Su resultado, se muestra en la figura 2 como función de $|NL|_{\max}$.

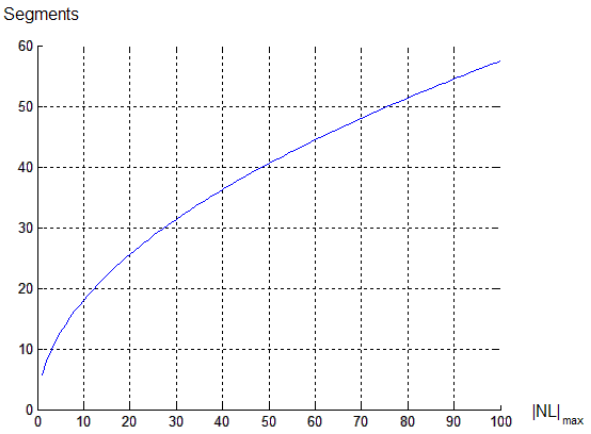

Figura 2: Evolución del número de segmentos de linealización en función del valor máximo de la no linealidad de la transfomación

Cuando de la ecuación se obtenga un valor inferior a 1, el número de tramos en los que linealizar será 1; este será el caso para matrices de homografía afines o aproximadamente afines. En caso contrario, el número de tramos en los que linealizar la transformación será el que indique la ecuación.

Podría aplicarse un sistema que permitiera calcular la no linealidad máxima de cada fila, de modo que se obtuviese un número de tramos más ajustados a las necesidades de cada zona de la imagen. El problema es que aparecería una nueva carga computacional; la solución más adecuada pasa por un compromiso entre tiempo de cómputo y calidad de la linealización.

Agradecimientos

Agradecer la inestimable ayuda del "Grupo de Robótica, Visión y Control" de la universidad de Sevilla, en cuyo seno se ha desarrollado este trabajo.

Referencias

[1] A. Ollero, J. Ferruz, F. Caballero, S. Hurtado and L. Merino. "Motion compensation and object detection for autonomous helicopter visual navigation in the COMETS system", Proceedings of the 2004 IEEE International Conference on Robotics & Automation. 2004 IEEE International Conference on Robotics & Automation - Icra 2004. New Orleans, la. IEEE Robotics and Automation Society. 2004. Pag. 19-24. ISBN: 0-7803-8233-1.

[2] J. Ferruz, "Sistema para Establecimiento de Correspondencia en secuencias de Imágenes. Aplicaciones en Robótica Móvil", Febrero 1997, Universidad de Sevilla

[3] K. Nordberg, P. Doherty, G. Farneback, P-E. Forssen, G. Granlund, A. Moe and J. Wiklund, "Vision for a UAV helicopter," 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems– IROS 2002. Proc. Workshop WS6 Aerial Robotics, pp 29-34. Lausanne, 2002.

[4] L. Merino and A. Ollero, "Forest fire perception using aerial images in the COMETS Project," 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems – IROS 2002. Proc. Workshop WS6 Aerial Robotics, pp 11-22. Lausanne, Switzerland, 2002.

[5] L. Merino, and A. Ollero, "Computer vision techniques for fire monitoring using aerial images," Proc. of the IEEE Conference on Industrial Electronics, Control and Instrumentation IECON 02 Seville (Spain), 2002.

[6] O. Faugueras and Q. Luong, "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, 2001.

[7] O. Shakernia, O., R. Vidal, C. Sharp, Y. Ma and S. Sastry, "Multiple view motion estimation and control for landing and unmanned aerial vehicle," Proceedings of the IEEE International Conference on Robotics and Automation, 2002.

[8] P. Forssén, "Updating camera location and heading using a sparse displacement field", Report LiTH-ISY-R-2318, November 2000.

[9] P.J. Garcia-Pardo, G.S. Sukhatme and J.F. Montgomery, "Towards Vision-Based Safe Landing for an Autonomous Helicopter," Robotics and Autonomous Systems, Vol. 38, No. 1, pp. 19-29, 2001.

[10] R. I. Hartley, "In defense of the height-point algorithm", IEEE transactions on pattern anlysis and machine intelligence, Vol. 19 No. 6 June 1997.

[11] R. Vidal, S. Sastry, J. Kim, O. Shakernia and D. Shim, "The Berkeley Aerial Robot Project (BEAR)," 2002 IEEE/RSJ Int. Conf. on Intelligent Robots and Systems – IROS 2002. Proc. Workshop WS6 Aerial Robotics, pp 1-10. Lausanne, Switzerland, 2002.

[12] S. Saripally, and G.S. Sukhatme, “Landing on a mobile target using an autonomous helicopter,” IEEE Conference on Robotics and Automation, 2003.

[13] Z. Zhang, “Parameters estimation techniques. A tutorial with application to conic fitting”, INRIA, No. 2676 October 1995.