

MÉTODOS PARA EL ALINEAMIENTO DE MAPAS DE CARACTERÍSTICAS VISUALES

Mónica Ballesta, Óscar Reinoso, Arturo Gil, Miguel Juliá, Luis Payá
Departamento de Ingeniería de Sistemas Industriales
Universidad Miguel Hernández
03202, Elche (Alicante), SPAIN
{m.ballesta, arturo.gil,o.reinoso,mjulia,lpaya}@umh.es

Resumen

En un sistema multirobot, en el que cada robot construye su propio mapa de forma independiente, puede ser necesario realizar la fusión de estos mapas locales en un único mapa global. Esta tarea se conoce como Fusión de Mapas y se realiza fundamentalmente en dos pasos: el alineamiento entre los mapas y la posterior fusión de los datos en un único mapa. Este artículo se centra en el primer paso. La fase de alineamiento consiste en calcular la transformación, si existe, entre los mapas locales construidos por los robots. En este sentido, se ha hecho una evaluación de varios métodos que calculan el alineamiento entre dos mapas. Dichos mapas están formados por landmarks visuales 3D y se construyen con el algoritmo FastSLAM a partir de observaciones reales realizadas por los robots.

Palabras clave: SLAM visual, Fusión de Mapas, Landmarks visuales, Sistema Multirobot.

1. INTRODUCCIÓN

Una de las habilidades fundamentales que debe tener un robot autónomo es que sea capaz de explorar el entorno y construir un mapa de él. Por este motivo, el problema de SLAM, de las siglas *Simultaneous Localization and Mapping*, ha recibido gran atención. En cuanto al tipo de sensores utilizados en esta tarea, muchos autores utilizan sensores SONAR [22] o LASER en 2D [13, 19] y 3D [21]. Sin embargo, existe un interés creciente en el uso de cámaras como sensores. Esta propuesta se conoce como SLAM visual. Las cámaras ofrecen mayor información acerca del entorno y son menos costosas que los sensores mencionados previamente. Además, podemos obtener información 3D del entorno si utilizamos cámaras estéreo.

La mayoría de las propuestas de SLAM visual trabajan con mapas basados en características (*landmark-based maps*). Estos mapas representan la localización de un conjunto de puntos distintivos del entorno respecto a un sistema de referencia global [14, 7].

El proceso de SLAM puede realizarse con un solo

robot, pero será más eficiente si un conjunto de robots coopera en la solución de esta tarea. Este enfoque se denomina SLAM multi-robot. En un sistema multi-robot, los robots exploran simultáneamente el entorno y realizan observaciones de él. De este modo, se va construyendo progresivamente el mapa de dicho entorno de forma cooperativa. Hasta ahora, existen varias soluciones al problema de SLAM multi-robot [10, 12]. Estas soluciones se pueden clasificar en dos grupos diferentes. Por un lado, hay soluciones en las que la estimación de las trayectorias y la construcción del mapa son realizadas de forma conjunta [5, 8, 19]. En estos casos, se construye un único mapa del entorno. Una ventaja de este tipo de solución es que los robots tienen una noción global del entorno y la exploración se puede realizar de un modo más eficiente. Sin embargo, el coste computacional es mayor y la posición inicial de los robots tiene que ser conocida, lo cual en la práctica no siempre es posible. Por otro lado, existe otro tipo de soluciones en las que cada robot construye su propio mapa local de forma independiente [25]. Los robots mantienen sus propios mapas locales hasta que la fusión de los mapas sea necesaria. Según [23], una de las ventajas principales de trabajar con mapas locales es que la asociación de datos se puede llevar a cabo de un modo más sencillo. En este caso, las nuevas observaciones sólo han de ser asociadas con un número limitado de marcas en los mapas locales. Además, la fusión de los mapas locales en un mapa común aumenta la robustez de la asociación de datos. Finalmente, se puede considerar el caso en el que las posiciones iniciales de los robots sean desconocidas.

La fusión de los mapas locales supone la realización de dos tareas diferentes: el alineamiento de los mapas y la posterior fusión de los datos para obtener un único mapa común. Este artículo se centra en la tarea de alineamiento de los mapas que consiste en encontrar la transformación, si existe, entre dichos mapas de modo que todas las medidas estén dadas respecto de un sistema de referencia común. En lo que concierne a la solución de este problema, algunas propuestas tratan de obtener las posiciones relativas de los robots. Cuando estas posiciones son conocidas, entonces

el alineamiento de los mapas es inmediato. En este sentido, el caso más sencillo se muestra en [19], donde la pose relativa de los robots se considera conocida. Un enfoque con más dificultad es el presentado en [12, 25]. En [6], el punto de encuentro se estima mediante un filtro de partículas. No obstante, serían más interesantes las soluciones en las que los robots sean capaces de determinar si se pueden alinear sus mapas sin la necesidad de establecer un punto de encuentro. En este caso la transformación entre los mapas se hallaría a partir de los datos compartidos entre los robots.

Algunos autores proponen técnicas basadas en características para alinear los mapas [11, 18, 20]. La base de estas técnicas es encontrar correspondencias entre las marcas de los mapas y obtener la transformación entre ellos. Este artículo se centra en este tipo de soluciones. Concretamente, nuestro propósito es analizar el funcionamiento de varios métodos que calculan el alineamiento entre mapas de características 3D. Los experimentos se llevan a cabo con datos reales capturados por los robots y procesados con el algoritmo FastSLAM.

2. CONSTRUCCIÓN DE MAPAS

Los robots móviles utilizados en este trabajo son Pioneer-3AT, dispuestos con un LASER y un par estéreo STH-MDCS2 de *Videre Design*, como se aprecia en la Figura 1. El algoritmo de navegación y construcción de mapas es el algoritmo *FastSLAM* presentado en [16]. En este caso, dos robots van construyendo progresivamente su mapa de características visuales a partir de la información del par estéreo y la odometría. Las marcas visuales son puntos de Harris obtenidos con el detector de esquinas de Harris [9] y descritos con U-SURF [3]. Esta combinación detector–descriptor fue seleccionada como la más adecuada teniendo en cuenta una serie de requerimientos como se detalla en [15, 2]. Los experimentos se han desarrollado en un edificio típico de oficinas. En dicho entorno los elementos más comunes son puertas, ventanas, pósters en las paredes, etc. Inicialmente, los robots parten de posiciones diferentes y a continuación van navegando y construyendo progresivamente sus mapas de forma independiente. Por tanto, los robots no tienen conocimiento acerca de la posición relativa del resto de robots ni de sus observaciones. Los mapas locales construidos por los robots están formados por las coordenadas 3D de los puntos de Harris y por sus correspondientes descriptores U-SURF. Además, cada mapa está referenciado al sistema de referencia local de cada robot, que está situado en su posición de partida.



Figura 1: Pioneer-3AT con el LASER y el par estéreo.

3. MÉTODOS DE ALINEAMIENTO DE MAPAS

3.1. RANSAC

Este método ha sido anteriormente aplicado a la alineación de mapas en [18]. A continuación, se detallan los pasos de este algoritmo.

1. Primeramente se obtiene una lista de posibles correspondencias. El criterio seguido para asociar correspondencias en base a su descriptor es la distancia Euclídea, además esta distancia ha de ser menor que un umbral th_0 . Las coordenadas $m = (x_i, y_i, z_i)$ son las landmarks de uno de los dos mapas, y $m' = (x'_i, y'_i, z'_i)$ sus correspondencias en el otro mapa.
2. En el siguiente paso, se selecciona un par de correspondencias al azar de la lista creada en el paso anterior. Este par de correspondencias debe satisfacer la siguiente restricción geométrica:

$$A^2 + C^2 \approx B^2 + D^2 \quad (1)$$

donde $A = (x'_i - x'_j)$, $B = (y'_i - y'_j)$, $C = (x_i - x_j)$ y $D = (y_i - y_j)$. En este caso se debe cumplir que $|(A^2 + C^2) - (B^2 + D^2)| < th_1$. Estos dos pares de correspondencias se utilizan para hallar los parámetros de alineación (t_x, t_y, θ) a partir de las siguientes ecuaciones:

$$t_x = x_i - x'_i \cos \theta - y'_i \sin \theta \quad (2)$$

$$t_y = y_i - y'_i \cos \theta + x'_i \sin \theta \quad (3)$$

$$\theta = \arctan \frac{BC - AD}{AC + BD} \quad (4)$$

3. El tercer paso consiste en buscar posibles correspondencias que apoyan la transformación calculada en el paso anterior (t_x, t_y, θ) .

Por cada punto del conjunto m' se le aplica la transformación calculada y se busca su correspondencia en el conjunto m estableciendo que la distancia Euclídea entre ambos sea menor que el umbral th_2 . A estas correspondencias se les denomina *apoyos*. Finalmente, se repiten el segundo y tercer paso M veces. La solución final será aquella que obtenga mayor número de apoyos.

En nuestros experimentos, hemos seleccionado los siguientes valores para los umbrales mencionados anteriormente: $th_0 = 2$ para establecer la lista inicial de correspondencias, $th_1 = 2$ para la restricción geométrica de la Ecuación (1) y $th_2 = 2$ para la búsqueda de apoyos. Además, el parámetro $min = 20$ establece el número mínimo de apoyos exigidos para que una solución se considere válida. Finalmente, $M = 70$ es el número de veces que los pasos 2 y 3 se repiten. Estos se consideran parámetros internos del algoritmo y se han seleccionado de forma experimental.

3.2. SVD

La Descomposición en Valores Singulares (*Singular Value Decomposition*, SVD) ha sido anteriormente aplicada en [1, 17] para establecer correspondencias entre conjuntos de puntos 3D. Concretamente, se obtiene un sistema de referencia común mediante el cálculo de la transformación existente entre los conjuntos de datos. En este trabajo, se aplica el método SVD para el cálculo del alineamiento entre dos mapas de características.

Dada una lista de posibles correspondencias, el objetivo es minimizar la siguiente expresión:

$$\|m'B - m\| \quad (5)$$

donde m' y m son los conjuntos de correspondencias entre los dos mapas. Por otro lado, B es la matriz de transformación entre los dos sistemas de coordenadas, que viene dada por la siguiente expresión:

$$B = \begin{pmatrix} \cos \theta & -\text{sen } \theta & 0 & 0 \\ \text{sen } \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ t_x & t_y & 0 & 1 \end{pmatrix} \quad (6)$$

B se calcula como indica el Algoritmo 1 de este apartado. La lista de correspondencias (m y m') se obtiene realizando el primer paso del algoritmo RANSAC indicado en la Sección 3.1. A continuación, la restricción geométrica de la ecuación 1 también se comprueba. Los parámetros internos son idénticos a los especificados en la Sección 3.1.

Algoritmo 1 Cálculo de la matriz de transformación con SVD.

Entrada: m y m'

Salida: Cálculo de la matriz B

$[u, d, v] = \text{svd}(m')$;

$z = u' \cdot m$;

$sv = \text{diag}(d)$;

$z_1 = z(1 : n)$; // n es el número de valores propios (distinto de 0) en sv .

$w = z_1 ./ sv$;

$B = v * w$;

3.3. ICP

El algoritmo ICP (*Iterated Closest Point*) se presentó por primera vez en [4, 24]. Este algoritmo itera en dos pasos:

1. Cálculo de correspondencias(m, m'). Dada una estimación inicial de los parámetros de alineamiento B_0 , se calcula el conjunto de correspondencias que apoya dicha estimación inicial.
2. Actualización de la transformación B . El conjunto anterior de correspondencias se utiliza para actualizar la transformación B . El nuevo B_{x+1} minimizará la expresión $\|m - m'B_{x+1}\|$, que es análoga a la expresión 5. Por este motivo, se ha realizado este paso con el algoritmo SVD (Algoritmo 1 en la Sección 3.2).

El algoritmo termina cuando el conjunto de correspondencias no cambia en el primer paso, y, por tanto, B_{x+1} es igual a B en el segundo paso.

Un aspecto a tener en cuenta con esta técnica es que requiere una estimación inicial de los parámetros de transformación bastante buena para asegurar la convergencia. Por este motivo, hemos realizado los dos primeros pasos del algoritmo de RANSAC para hallar esta estimación inicial, como se indica en la Sección 3.1.

3.4. ImpICP

El algoritmo ICP mejorado (*Improved ICP*) es una modificación del algoritmo de la sección 3.3, que ha sido desarrollado *ad hoc*. En la sección anterior, se explicó la importancia de tener una buena estimación inicial para que el algoritmo converja. Además en esa misma sección se describió el modo de obtener esta estimación inicial. No obstante, se creó esta nueva versión del algoritmo ICP dada la elevada influencia de la estimación inicial en el resultado final. Concretamente, con esta nueva versión se aumenta la probabilidad de obtener un

buen resultado. Para ello, en este caso, se obtienen tres pares diferentes de correspondencias en el segundo paso del algoritmo RANSAC (sección 3.1). Por cada una de estas estimaciones iniciales, el algoritmo itera según lo indicado en la sección 3.3. Finalmente, la solución seleccionada es la transformación que está avalada por mayor número de apoyos.

4. EXPERIMENTOS

El objetivo de estos experimentos es comparar el comportamiento de los métodos de alineamiento descritos en el apartado 3.1 en el contexto de mapas de características visuales. La situación de partida de estos experimentos es que los robots han partido desde posiciones iniciales diferentes y la transformación entre los sistemas de referencia locales es desconocida. Nuestro propósito es evaluar el comportamiento de estos métodos de alineamiento en diferentes etapas del proceso de construcción de los mapas. Al principio, los mapas tendrán muy pocas marcas y, por tanto, el número de correspondencias encontradas entre ambos mapas será muy bajo. En consecuencia, encontrar un posible alineamiento entre estos mapas será muy poco probable. Sin embargo, esta situación mejorará a medida que el tamaño de los mapas aumenta, de modo que hay más marcas coincidentes entre ambos mapas. En esta última situación, se espera que el alineamiento se realice con éxito.

Para calcular el alineamiento entre dos mapas, se utiliza el mapa más probable de cada robot, que se corresponde con el mapa de la partícula más probable del algoritmo de *FastSLAM*. Este proceso se repite a lo largo de diferentes iteraciones del algoritmo *FastSLAM*. Los métodos que se evalúan en este artículo calculan los parámetros de alineamiento t_x , t_y y θ . Estos parámetros permiten transformar las coordenadas de uno de los mapas en el sistema de referencia del otro mapa, de este modo se realiza el alineamiento. La exactitud de los métodos de alineamiento se evalúa por medio del error cometido en la estimación de los parámetros de alineamiento. En este caso, el error se calcula como la distancia Euclídea entre los parámetros de alineamiento t_x , t_y y θ y la posición relativa real entre los dos robots. Esta medida relativa, denominada *Ground Truth* se ha obtenido calibrando la posición de cada robot estando en el punto de partida. Esta posición inicial es el origen del sistema de referencia para cada robot. El error cometido en la estimación de la solución respecto del *Ground Truth* se mide en metros.

El algoritmo *FastSLAM* consta de varias iteraciones correspondientes al número total de movimientos realizados por el robot. En los experi-

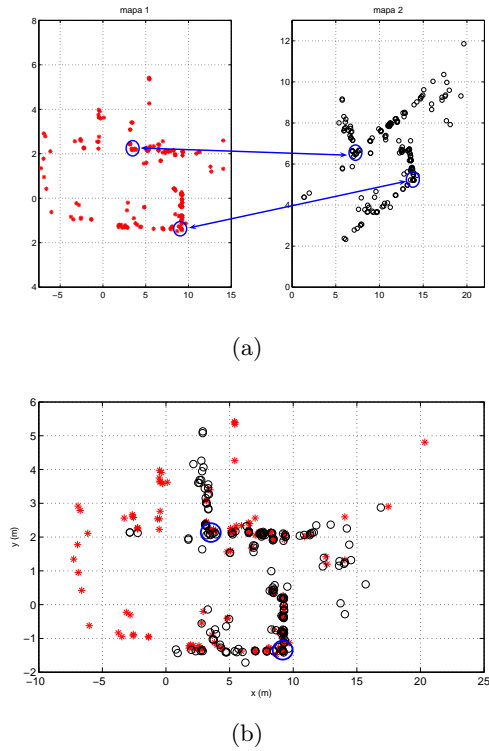


Figura 2: Alineamiento de mapas (vista 2D). Fig. 2(a) muestra los mapas locales antes del alineamiento. Fig. 2(b) muestra los mismos mapas después del alineamiento.

mentos, k es un índice que denota el orden de la iteración. En nuestro caso, el número total de iteraciones es $k = 1410$ y el tamaño final cuando los mapas se han completado es de $mapa_1 = 263$ marcas y $mapa_2 = 346$ marcas. A su vez, estos mapas tienen una dimensión de 35×15 metros.

En la Figura 2(a) se pueden observar los mapas construidos por cada robot y referidos a su correspondiente sistema local. En esta figura, map_1 se representa mediante asteriscos y tiene 181 marcas visuales. A su vez, map_2 está representado con círculos y tiene un total de 187 marcas. Por otro lado, en la figura 2(b) se muestra el resultado después de alinear ambos mapas. Este ejemplo corresponde con la iteración $k = 810$ del algoritmo *FastSLAM*.

La Figura 3 muestra una comparación entre los métodos de alineamiento que se quieren evaluar en este artículo. Para cada método, se representan los valores del error (eje y) frente a las k -iteraciones del algoritmo de *FastSLAM* (eje x). Logicamente, conforme aumenta el número de iteraciones, el tamaño de los mapas será mayor y será más probable encontrar una solución próxima al *Ground Truth*. Por esta razón, es de esperar que el error sea menor al aumentar el número de iteraciones del algoritmo. En la Figura 3 podemos observar

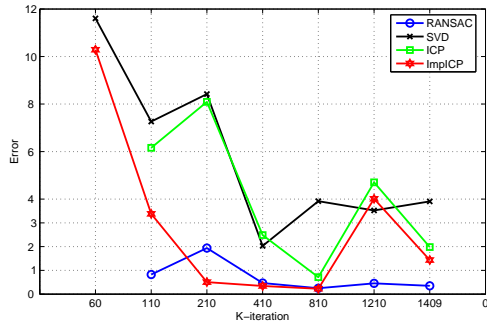


Figura 3: Evaluación de los métodos de alineamiento.

que los peores resultados se obtienen con el método SVD. A modo de ejemplo, SVD tiene un error de 4m en la k -iteración=1490, lo cual es un error bastante considerable si se tiene en cuenta que el mapa está prácticamente completo. Por otra parte, ICP obtiene resultados similares, aunque en algunos casos supera a SVD. Concretamente, con k - iteración = 810 el error es menor que 1m. Por otro lado, el algoritmo ImplICP supera a estos métodos ya que obtiene valores de error bastante pequeños. Sin embargo, el algoritmo RANSAC resulta ser el que mejor funciona para este tipo de mapas. A pesar de que no encuentra ninguna solución de alineamiento cuando los mapas tienen pocas marcas (k - iteración = 60), este algoritmo obtiene los valores de error más pequeños. De hecho, a partir de k - iteración = 410 el error no supera los 0,5m.

Finalmente, la Figura 4 se centra en los resultados obtenidos con RANSAC. En la Figura 4(a) se muestra el número de apoyos que tienen las soluciones de este algoritmo a medida que aumenta k - iteración. Como se observa el número de apoyos es mayor conforme lo es el número de marcas en los mapas. Por otro lado, la Figura 4(b) muestra los valores de error (en metros) cometido en cada una de los parámetros de alineamiento por separado. Como se aprecia en la figura, la estimación del parámetro t_y es la más crítica.

5. CONCLUSIONES

El principal objetivo de este artículo ha sido evaluar y seleccionar un método de alineamiento que funcione adecuadamente con mapas de características visuales tridimensionales. Para llevar a cabo estos experimentos, se han utilizado datos reales capturados por los robots navegando por una zona de nuestro edificio. El proceso de mapeado se ha realizado con el algoritmo FastSLAM. Como resultado, el método RANSAC es el que ha demostrado ser más adecuado para el alineamiento

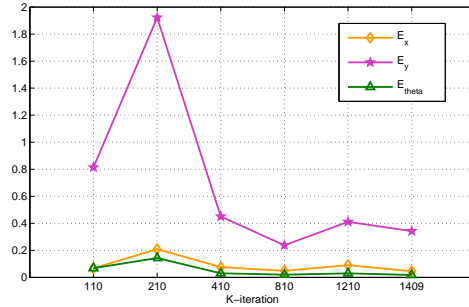
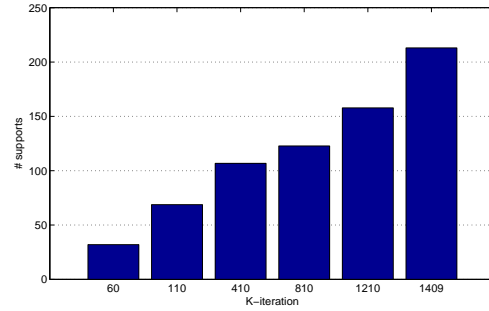


Figura 4: Resultados obtenidos con el algoritmo de RANSAC. Fig. 4(a) muestra el número de apoyos obtenido. Fig. 4(b) muestra el error en cada uno de los parámetros de alineamiento.

de este tipo de mapas. Los resultados obtenidos muestran valores de error muy bajos y un comportamiento estable ante diferente número de marcas en los mapas.

Como trabajo futuro se pretende estudiar la siguiente etapa en el proceso de Fusión de mapas, denominada *Map Merging*. Esta segunda etapa consiste en la integración de la información contenida en los mapas locales para crear un único mapa común a todos los robots.

Agradecimientos

Esta investigación está patrocinada por el Ministerio de Educación y Ciencia (referencia del proyecto: CICYT DPI2007-61107, título 'Sistemas de Percepción Visual Móvil y Cooperativo como Soporte para la Realización de Tareas con Redes de Robots') y por la Generalitat Valenciana con la beca BFPI/2007/096.

Referencias

- [1] K.S. Arun, T.S. Huang, and S.D. Blostein. Least square fitting of two 3d sets. In *IEEE Transactions on Pattern Analysis and Machine Intelligence*. vol. PAMI-9 no. 5, pp.698-700, 1987.
- [2] M. Ballesta, A. Gil, O. Martínez Mozos, and O. Reinoso. Local descriptors for visual slam.

- In *Workshop on Robotics and Mathematics (ROBOMAT07), Coimbra, Portugal, 2007.*
- [3] H. Bay, T. Tuytelaars, and L. Van Gool. Surf: Speeded-up robust features. In *Proc. of the 9th European Conference on Computer Vision, 2006.*
 - [4] P.J. Besl and N. McKay. A method for registration of 3-d shapes. In *IEEE Transactions on Pattern Analysis and Machine Intelligence. vol. PAMI-14 no. 2, pp. 239-256, 1992.*
 - [5] John W. Fenwick, Paul N. Newman, and John J. Leonard. Cooperative concurrent mapping and localization. In *Proc. of the 2002 IEEE International Conference on Intelligent Robotics and Automation, pp.1810-1817, 2002.*
 - [6] D. Fox. Distributed multi-robot exploration and mapping. In *Proc. of the 2nd Canadian conference on Computer and Robot Vision, 2005.*
 - [7] A. Gil, O. Reinoso, W. Burgard, C. Stachniss, and O. Martínez Mozos. Improving data association in rao-blackwellized visual SLAM. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems, 2006.*
 - [8] A. Gil, O. Reinoso, L. Payá, and M. Ballesta. Influencia de los parámetros de un filtro de partículas en la solución al problema de SLAM. In *IEEE Latin America, 2007.*
 - [9] C. G. Harris and M. Stephens. A combined corner and edge detector. In *Alvey Vision Conference, 1998.*
 - [10] A. Howard. Multi-robot simultaneous localization and mapping using particle filters. In *The International Journal of Robotics Research, Vol. 25, No. 12, 1243-1256, 2006.*
 - [11] J. Ko, B. Stewart, D. Fox, K. Konolige, and B. Limketkai. A practical, decision-theoretic approach to multi-robot mapping and exploration. In *Proc. of the IEEE/RSJ Int. Conf. on Intelligent Robots and Systems (IROS), pp. 3232-3238., 2003.*
 - [12] K. Konolige, D. Fox, B. Limketkai, J. Ko, and B. Stewart. Map merging for distributed robot navigation. In *Proc. of the 2003 IEEE/RSJ International Conference on Intelligent Robots and Systems, 2003.*
 - [13] J.J. Leonard and H.F. Durrant-Whyte. Mobile robot localization by tracking geometric beacons. *IEEE Transactions on Robotics and Automation, 7(4), 1991.*
 - [14] J. Little, S. Se, and D.G. Lowe. Global localization using distinctive visual features. In *IEEE/RSJ Int. Conf. on Intelligent Robots & Systems, 2002.*
 - [15] O. Martínez Mozos, A. Gil, M. Ballesta, and O. Reinoso. Interest point detectors for visual slam. In *Proc. of the XII Conference of the Spanish Association for Artificial Intelligence (CAEPIA), Salamanca, Spain, 2007.*
 - [16] M. Montemerlo, S. Thrun, D. Koller, and B. Wegbreit. Fastslam: A factored solution to simultaneous localization and mapping. In *Proc. of the National Conference on Artificial Intelligence (AAAI), pp. 593-598. Edmonton, Canada, 2002.*
 - [17] J. Rieger. On the classification of views of piecewise smooth objects. In *Image and Vision Computing, vol. 5, no. 2, pp. 91-97, 1987.*
 - [18] S. Se, D. Lowe, and J.J. Little. Vision-based global localization and mapping for mobile robots. In *IEEE Transactions on robotics, vol.21, no.3, 2005.*
 - [19] S. Thrun. A probabilistic online mapping algorithm for teams of mobile robots. In *Int. Journal of Robotics Research, 20(5), pp. 335-363, 2001.*
 - [20] S. Thrun and Y. Liu. Simultaneous localization and mapping with sparse extended information filters. In *The International Journal of Robotics Research.; 23: 693-716, 2004.*
 - [21] R. Triebel and W. Burgard. Improving simultaneous mapping and localization. In *Proc. of the National Conference on Artificial Intelligence (AAAI), 2005.*
 - [22] O. Wijk and H. I. Christensen. Localization and navigation of a mobile robot using natural point landmark extracted from sonar data. In *Robotics and Autonomous Systems, 1(31), pp. 31-42, 2000.*
 - [23] S. Williams. Phd dissertation: Efficient solutions to autonomous mapping and navigation problems. Australian Center for Field Robotics, University of Sidney, 2001.
 - [24] Z. Zhang. On local matching of free-form curves. In *Proc. of BMVC, pp. 347-356, 1992.*
 - [25] Xun S. Zhou and Sergios I. Roumeliotis. Multi-robot slam with unknown initial correspondence: The robot rendezvous case. In *Proc. of the 2006 IEEE/RSJ International Conference on Intelligent Robots and Systems, Beijing, China, pp. 1785-1792, 2006.*