CHAPTER X

Odometría visual estéreo en exteriores embarcada en un robot Seekur jr.

R. RODRÍGUEZ[1], R. BAREA[1], L.M. BERGASA[1], E. LÓPEZ[1], M. OCAÑA[1], P. F. ALCANTARILLA[1]

[1]Departamento de Electrónica. Universidad de Alcalá. {barea, bergasa, elena, mocana}@depeca.uah.es

Este artículo describe una implementación práctica de odometría visual estéreo embarcada en un robot Seekur jr. Se ha realizado una comparativa entre dos conjuntos de cámaras estéreo y se han estudiado sus características y prestaciones. El sistema está formado por el robot seekur jr, un sistema de cámaras estéreo (Unibrain o Bumblebee) y un GPS, utilizando como interfaz común de comunicación entre ellos el software servidor Player del proyecto Player/Stage. Se ha implementado y probado un algoritmo de odometría visual en los exteriores del edificio de la Escuela Politécnica de la Universidad de Alcalá.

1 Introducción

En el mundo de la robótica es muy importante conocer la posición del robot en todo momento. Existen distintas técnicas que ayudan a la localización de un robot.

Es importante el uso del GPS en exteriores debido a su bajo coste, pero con un error tal que solo permite realizar la planificación tosca del movimiento del robot para llegar a su objetivo.

La odometría clásica mediante encoders asociados a las ruedas del robot, que puede llegar a ser muy útil en interiores, puede llegar a ser muy

ineficaz en la navegación en exteriores, debido a las características del terreno.

Existe gran número de experiencias utilizando odometría visual en espacios interiores de edificios con resultados bastante positivos. Así mismo existen experiencias en exteriores como por ejemplo para la asistencia a vehículos en carretera (Garcia, 2008).

Cabe mencionar el trabajo en el que a bordo de un robot se ha fusionado mediante un filtro de Kalman la odometría visual con una unidad de medida inercial y GPS proyectando los resultados en 2D (Agrawal, 2006). Otro similar que se ha realizado a bordo de un coche y en espacios exteriores (Nourani, 2008).

En otros casos (Howard, 2008) se han conseguido resultados exitosos utilizando la fusión de una unidad de medida inercial con la salida de un sistema de cámaras estéreo con procesamiento de imágenes y cálculo de mapa de disparidad.

Recientemente se ha publicado el trabajo en el que con un robot RHex de patas se ha experimentado en espacios interiores y exteriores realizando navegación y mapeado en 3D (Morisset, 2009).

Se considera tan interesante el uso de la odometría visual que incluso existen trabajos en los que los exteriores visitados están fuera del planeta, en concreto Marte (Maimone, 2007).

Este trabajo se divide en las siguientes secciones: en primer lugar, se va a describir la arquitectura del sistema implementado en la sección 2. En la sección 3 se indica el funcionamiento del algoritmo de odometría visual. Finalmente, la sección 4 muestran los resultados obtenidos con los dos sistemas de cámaras utilizados y finalmente se comentan una serie de conclusiones como resultado de la experiencia.

2 Arquitectura hardware

El estudio se ha realizado sobre una plataforma robótica Seekur Jr. que ha sido acondicionada para llevar a bordo una cámara estéreo, un GPS y un ordenador. La figura 1 muestra una imagen del sistema implementado.

Se han utilizado dos tipos de cámaras para realizar una comparativa entre sus prestaciones: una con configuración estéreo montada con dos cámaras Firewire de la firma Unibrain [Unibrain, 2010] y las Bumblebee [Bumblebee, 2010]. La figura 2 muestra su colocación sobre el robot Seekur jr.

El robot Seekur jr. dispone de movilidad de tracción diferencial con capacidad para mover una carga de hasta 50 Kg, además de los 70 Kg de su peso en vacío. Dispone de un software servidor clásico de la firma Activmedia, en una versión específica para el mismo.

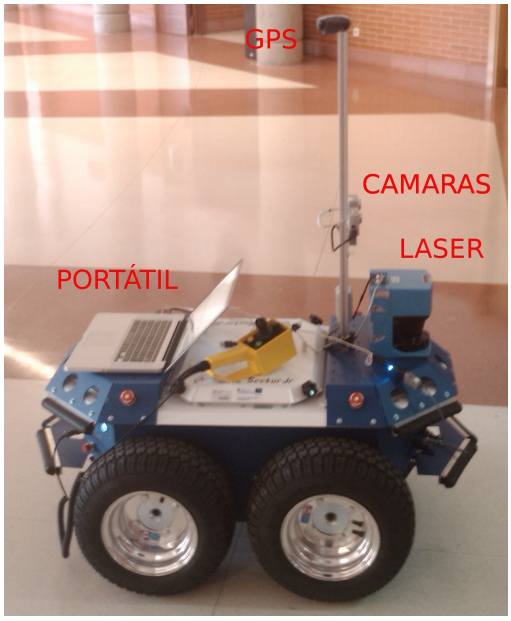

Fig. 1. Plataforma robótica.

Fig. 2. Configuración cámaras estéreo: a) Unibrain. b) Bumblebee.

Se ha modificado un driver ya existente del software Player para hacer de interfaz entre el sistema y el servidor del robot. El fin de esta comunica-

ción en el trabajo ha sido recibir la información de odometría y GPS a través del Player.

La odometría suministrada por el robot se puede considerar muy fiable puesto que no es una odometría clásica, por el contrario entrega una odometría que es el resultado de la fusión de la odometría de las ruedas, bastante precisa, con una unidad de medida inercial, usada principalmente para la corrección en los giros.

En la figura 3 se indica de forma gráfica la arquitectura lógica del sistema para la toma de datos en las pruebas prácticas. Las cámaras se comunican con el ordenador por el bus Firewire, el GPS por Bluetooth y el robot a través de un adaptador de USB a serie.

Las capturas de las cámaras se han tomado a 30 fps para ser reducida la tasa de muestras a valores inferiores en el software posteriormente. Esto se ha realizado con el fin de minimizar el desfase inevitable entre la adquisición de las imágenes de las dos cámaras cuando se utiliza la configuración de dos cámaras independientes sin sincronización (Unibrain).

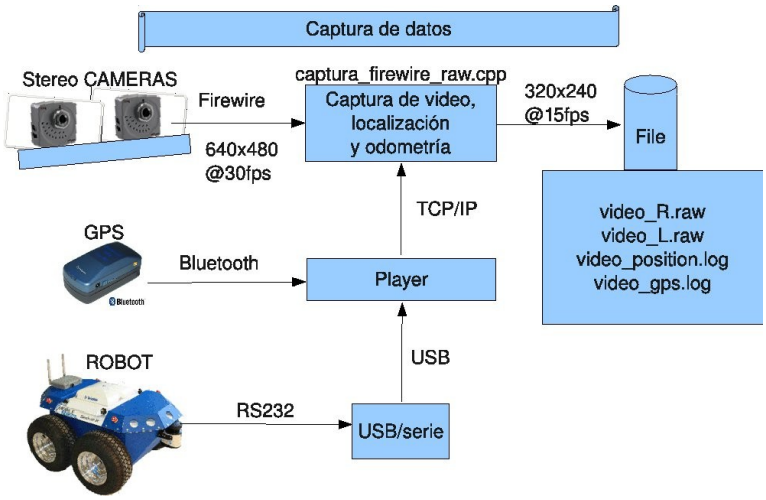

Fig. 3.Arquitectura software para la captura de datos.

La elección del uso del servidor Player viene indicada por el hecho de disponer de una capa de abstracción software para desarrollar software independiente del hardware específico utilizado.

3 Algoritmo de odometría visual

El resultado buscado de odometría visual ha consistido en realizar una estimación de movimiento del robot frame a frame en base únicamente a la información de las cámaras.

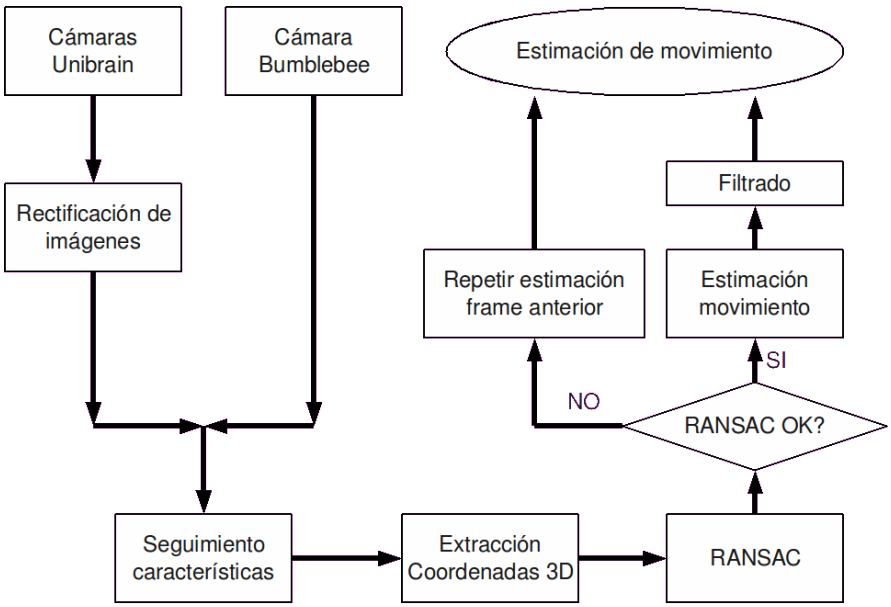

Fig. 4. Diagrama general del algoritmo de odometría visual.

Para ello se realizan distintas etapas como se puede observar en la figura 4. En primer lugar se realiza una rectificación de las imágenes, transformando las imágenes procedentes de las cámaras en imágenes equivalentes a las que hubieran sido capturadas por cámaras ideales sin distorsión, perfectamente coplanares y sus ejes ópticos completamente paralelos. Se realiza antes de todo una calibración de las cámaras para obtener los parámetros necesarios en la rectificación de imágenes.

En el caso de la cámara Bumblebee se toman las imágenes ya rectificadas de la cámara.

Con las imágenes rectificadas se realiza una detección y seguimiento de características frame a frame. Las características a seguir son puntos de la imagen detectados con el algoritmo de Shi-Tomasi (Shi y Tomasi, 1994) implementado en la librería OpenCV. En la figura 5 se muestra el diagrama de bloques del funcionamiento del algoritmo de seguimiento de características.

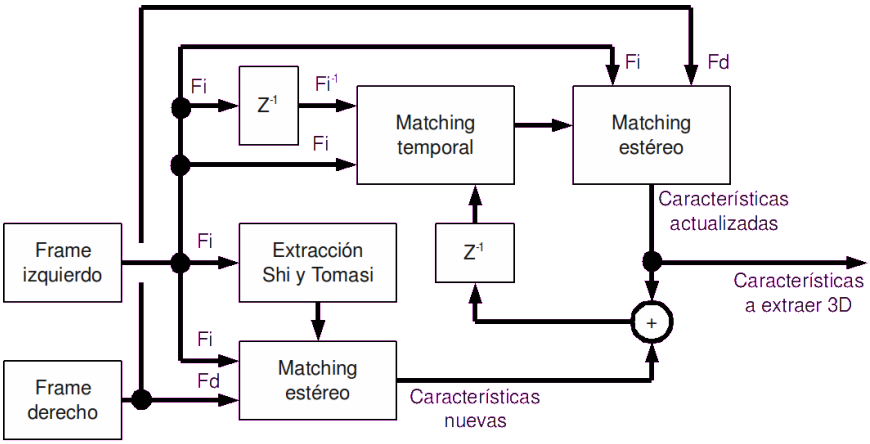

Fig. 5. Diagrama de bloques del seguimiento de características.

En el diagrama se muestra como por un lado se extraen características nuevas, mientras que por el otro se hace seguimiento de las del frame anterior en la imagen izquierda, para finalmente, si pasan el matching estéreo, sumarse con las nuevas, conformando el total de las características a seguir el frame siguiente.

En la figura 6 se indica el comportamiento del seguimiento de características. Se puede apreciar como hay características que pasando los matching temporal y estéreo, sucesivamente, consiguen sobrevivir para ser evaluadas en el siguiente frame.

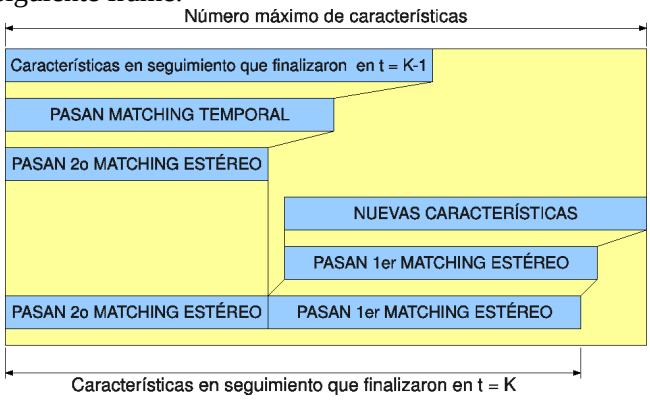

Fig. 6. Comportamiento del seguimiento de características frame a frame.

Dado que se hace el seguimiento de las características, en vez de como se hace en otros trabajos que se buscan características nuevas en cada frame, se ha podido estudiar el comportamiento de la experiencia variando la tasa de captura de frames a posteriori, una vez realizada la captura.

De las características en seguimiento actualizadas se extraen sus coordenadas 3D en ambos instantes de tiempo, mediante la información obtenida en la calibración estéreo.

Conocidas las coordenadas 3D con respecto al robot, de un conjunto de puntos estáticos en dos instantes distintos, se puede obtener el vector de movimiento asociado al desplazamiento realizado por el robot. Este vector es de seis componentes puesto que incluye tanto la traslación realizada, como la rotación con respecto a cada uno de los ejes, fruto del cambio de orientación que haya sufrido el robot.

No se pueden usar todos los puntos correspondientes a las características en seguimiento dado que algunos no son correctos. Por ejemplo se puede estar haciendo seguimiento a un punto que no sea estático.

Se implementa un algoritmo de selección de 'inliers' (puntos correctos) y rechazo de 'outliers' (incorrectos) denominado RANSAC (Fischler &Bolles, 1981). En la figura 7 está el diagrama de flujo del algoritmo de RANSAC.

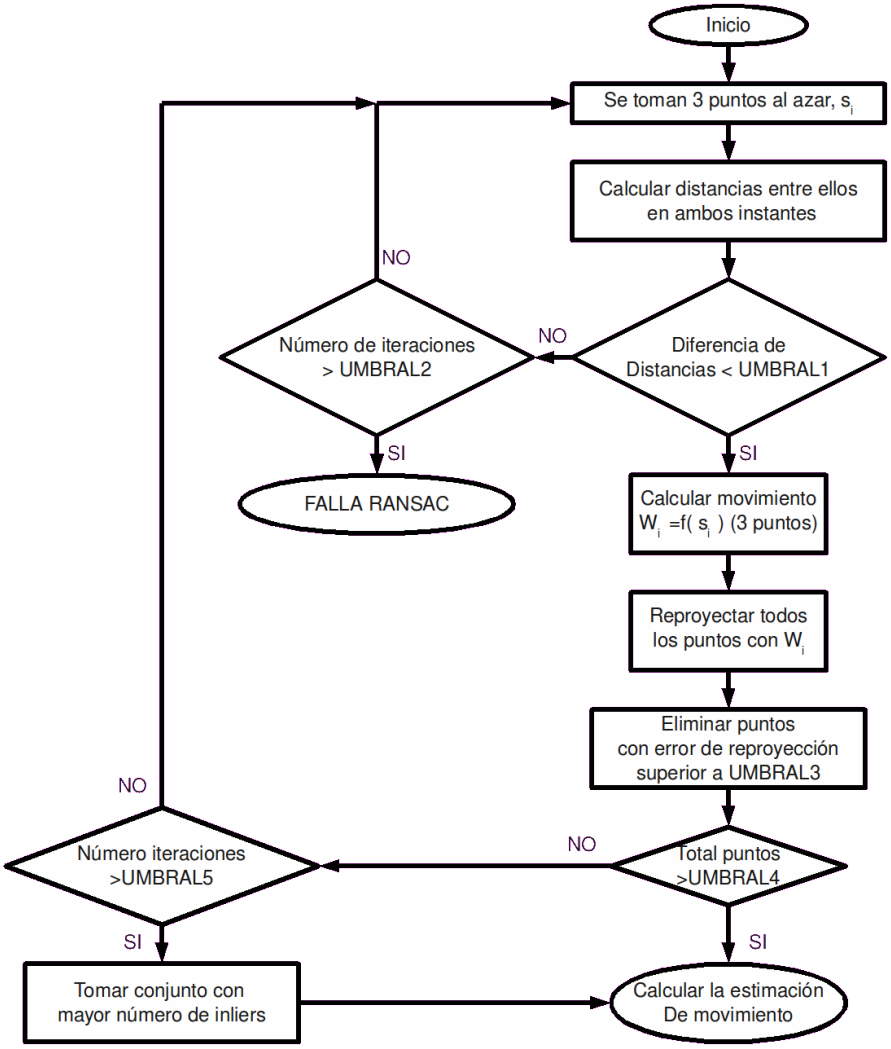

Fig. 7. Diagrama de flujo del algoritmo de RANSAC.

Inicialmente, se toman muestras al azar de tres puntos (s_i) a las que se les comprueba que cumplan con la restricción de no movimiento. Esta comprobación consiste en comprobar que los tres puntos mantienen una distancia constante entre ellos en ambos instantes. Esto se hace mediante la

comparación de la diferencia de distancias en ambos instantes con un umbral de error de la distancia (UMBRAL1). Si esta comprobación falla más veces de un número determinado (UMBRAL2) se considera que el RANSAC ha fallado. En este caso la estimación de movimiento será el mismo valor de la última iteración.

Cuando una tripla de puntos cumpla la restricción, se calcula el vector de movimiento inicial asociado mediante el algoritmo de Besl (Besl y McKay, 1992).

Con el vector de movimiento obtenido (w_i) se calcula el error de reproyección de todos los puntos en el segundo instante, como la diferencia entre la posición medida, y la obtenida aplicando el vector de movimiento a los puntos en su ubicación inicial. Se consideran outliers aquellos cuyo error de reproyección supere un cierto umbral (UMBRAL3). Se realizará esta operación con nuevas muestras de tres puntos al azar hasta obtener un conjunto de inliers superior a un umbral fijado (UMBRAL4), o cuando se haya iterado más veces que otro valor umbral máximo de iteraciones (UMBRAL5). En este último caso se toma como conjunto de inliers el mayor de los obtenidos en las distintas iteraciones.

A continuación, si el RANSAC no ha fallado, se realiza una estimación de movimiento de nuevo pero teniendo en cuenta todos los puntos del conjunto de inliers. En este caso se usa una solución de mínimos cuadrados no lineales que precisa de un valor inicial para de forma iterativa aproximarse a la solución final. Ese valor inicial es el que había sido calculado con la muestra de tres puntos al azar que dio origen al conjunto de inliers (w_i).

Con todo lo anterior se pueden obtener estimaciones erróneas por diversas causas. Para evitarlo se realiza un filtrado de las estimaciones aplicando criterios de cumplimiento de restricciones como pueden ser la velocidad y aceleración máximas del robot, que ponen cota a los valores que se pueden dar en las componentes del vector de movimiento.

Así mismo se filtra en función del valor final del error de estimación que da el algoritmo de mínimos cuadrados no lineales, y del error de reproyección de los puntos.

4 Resultados

Se ha experimentado con los dos tipos de cámaras sobre los jardines cercanos a la Escuela Politécnica Superior. Se ha considerado como 'ground-truth' de las pruebas la odometría suministrada por el robot.

Todas las tomas de datos mostradas se han realizado con las cámaras sobre el soporte que produce un ángulo de las mismas de unos 33 grados hacia abajo con respecto a la horizontal.

Se han tomado los videos en condiciones de luz natural, pero evitando la exposición en zonas muy soleadas, principalmente porque el ajuste automático de la cámara Bumblebee no actúa adecuadamente, y se saturan las imágenes. Además en las cámaras Unibrain el ajuste automático funciona de forma independiente en cada cámara, produciendo graves trastornos posteriores en los procesos de emparejamiento estéreo.

4.1 Ejemplo1. Cámara Bumblebee.

En este trayecto se realiza un bucle cerrado de unos 43 metros sobre césped con una cámara Bumblebee. La figura 5 muestra la trayectoria estimada en base a los datos obtenidos realizando un submuestreo de 5, esto es, el equivalente a una captura realizada a 3 frames por segundo.

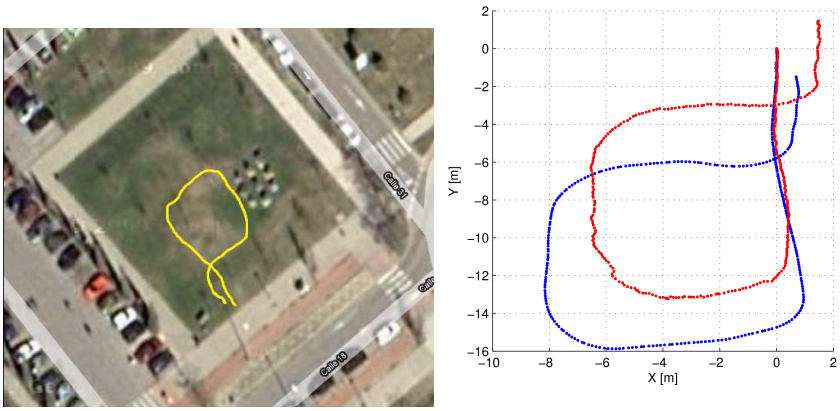

Fig. 5. Ejemplo1: trayectoria estimada (rojo) con un submuestreo de 5 y ground-truth (azul)

Se aprecia como la forma de la trayectoria es similar, si bien, los errores longitudinales a la trayectoria producidos en el primer y segundo tramos rectos hacen que la posición final de la trayectoria estimada y real se alejen unos 3 metros (error<10%).

La trayectoria realizada es de 43.31 m y la estimada de 40.76 m, lo que da un error de estimación del 10%.

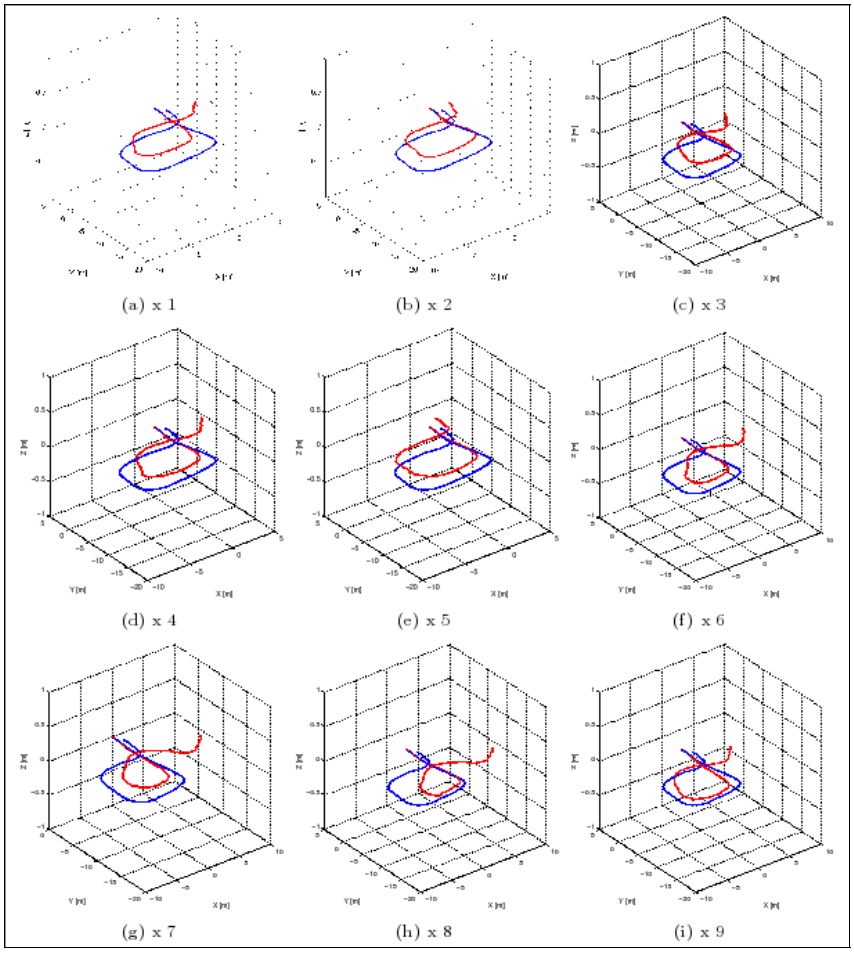

Fig. 6. Ejemplo 2: trayectorias con distintos submuestreos

De las experiencias mostradas en la figura 6, la que menor error de estimación de la longitud de la trayectoria ofrece es la del submuestreo por 9. En concreto para una trayectoria de 42.96 m de longitud, da una estimación de 41.38 m, esto es, un error del 3.66%.

4.2 Ejemplo 2. Unibrain.

En este trayecto se realiza un bucle cerrado de unos 60 metros sobre césped, como se puede observar en la figura 7.

Fig. 7. Ejemplo 2.- Trayectoria trazada sobre foto aérea.

En la tabla 1 se muestra el error de estimación de la longitud de la trayectoria para distintos valores de submuestreo.

Se puede observar cómo el error de estimación varía en las distintas estimaciones para cada valor del factor de submuestreo. Se llega incluso a un valor de error de 1.68% en el caso de submuestreo x2, equivalente a una captura de datos a 7.5 muestras por segundo. Si bien es el más bajo valor de longitud de la trayectoria, la forma de la misma está distorsionada con

respecto a la realizada, debido a la acumulación de errores de estimación de los giros.

En la figura 8 se muestra el resultado de la odometría visual comparado con la odometría propia del robot, mostrando el resultado obtenido con un submuestreo de 8, con un resultado de error de estimación de la trayectoria de 6.47%.

Tabla 1. Ejemplo 2: comparativa del error de la longitud de la trayectoria, a distintas frecuencias de muestreo equivalentes mediante el submuestreo.

Factor	**Fps equ. [fps]**	**Distancia [m]**	**Estimada [m]**	**Error [%]**
1	15	61.62	68.06	10.4
2	7.5	61.61	60.59	1.68
3	5	61.61	58.55	4.98
4	3.75	61.61	56.99	7.5
5	3	61.61	57.64	6.45
6	2.67	61.61	58.05	5.79
7	2.14	61.61	56.51	8.28
8	1.88	61.58	57.6	6.47
9	1.67	61.58	57.12	7.25
10	1.5	61.52	56.46	8.22
15	1	61.14	56.39	7.78
20	0.75	61.03	53.88	11.73
25	0.6	60.7	57.86	4.69
30	0.5	60.24	58.65	2.64
35	0.43	59.56	53.79	9.69
40	0.38	59.16	56.45	4.58

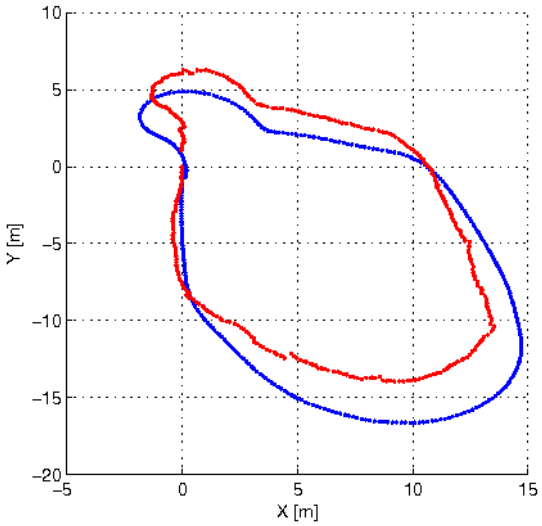

Fig. 8. Ejemplo2: trayectoria estimada (rojo) con un submuestreo de 8 y ground-truth (azul)

5 Conclusiones

En este trabaja se ha desarrollado y probado en la práctica un sistema de odometría visual para exteriores integrado en una plataforma robótica Seekur Jr.

De los resultados obtenidos se extraen distintas conclusiones que se exponen a continuación:

- Los resultados obtenidos en la estimación de la longitud de la trayectoria pueden considerarse satisfactorios. Esto contrasta con la estimación del ángulo de giro en las curvas, donde los errores obtenidos son considerables.
- La relación entre la velocidad del robot y el periodo de muestreo es muy importante en la obtención de valores de estimación de movimiento con poco error. En la práctica cuando la navegación sea reactiva y en situaciones de actuación útil del robot es posible que se deba trabajar a diferentes velocidades, con lo que será difícil la elección de periodo de muestreo adecuado. Es por ello que en los trabajos futuros

se propondrá la implementación de un periodo de muestreo dinámico y adaptativo.

- Se ha comprobado como muy relevante el ángulo de la orientación de la cámara con respecto del suelo. Se debe disponer de un número mayor de referencias cercanas que permiten que la estimación de movimiento se realice de una forma más precisa. No obstante la detección del ángulo de giro se realiza mejor con referencias lejanas. Por lo tanto, en la práctica hay que llegar a un equilibrio que permita disponer tanto de referencias cercanas, útiles para la precisión del desplazamiento, como referencias lejanas que tienen una gran importancia en la corrección de los errores angulares de la orientación. En el caso de la cámara bumblebee, con el ángulo tomado en la práctica (inclinación de 33º hacia abajo), se obtienen mejores resultados que con las Unibrain, debido a que estas últimas disponen de un menor campo de visión, y por tanto pierden la posibilidad de hacer seguimiento de referencias lejanas, mientras que la bumblebee se dispone de ambos tipos.
- Por la comparación de los resultados de los dos tipos de cámaras utilizados se deduce que se obtienen mejores resultados con un mayor campo de visión, como es el caso de las cámaras Bumblebee. Esto es debido a que un mayor campo de visión permite detectar mayor variedad de puntos y obtener mejores estimaciones de movimiento. Es especialmente importante la detección de puntos lejanos para realizar buenas estimaciones de los giros.
- El número de características en seguimiento es crítico para realizar una buena estimación de movimiento. Se observa en las experiencias que cuando este número es bajo la estimación es incorrecta. Este problema es especialmente importante en los giros, dado que la vida media de las características es menor y disminuye el número total de éstas.

Agradecimientos

Los autores desean expresar su gratitud a la Comunidad de Madrid por su financiación a través del proyecto RoboCity2030 II: Robots de servicios

para la mejora de la calidad de vida de los ciudadanos en áreas metropolitanas II (CAM-S2009/DPI-1559)

Referencias

Agrawal, M. 2006. Real-time Localization in Outdoor Environments using Stereo Vision and Inexpensive GPS, ICPR(3): 1063-1068.

Besl, P. y McKay, N. 1992. A method fro registration of 3-d shapes, IEEE Trans. PAMI, pp. 239-256

Bumblebee. 2010. http://www.ptgrey.com/products/stereo.asp

Fischler, M. A. Bolles, R. C. 1981. Random Sample Consensus: A Paradigm for Model Fitting with Applications to Image Analysis and Automated Cartography. Comm. of the ACM, Vol 24, pp 381-395.

García, R. 2008. 3D Visual Odometry for Road Vehicles, Journal of Intelligent and Robotic Systems, vol 51, pp. 113-134.

Howard, A. 2008. Real-Time Stereo Visual Odometry for Autonomous Ground Vehicles, IEEE/RSJ.

Morisset, B. 2009. Leaving Flatland: Toward real-time 3D navigation, ICRA: 3786-3793.

Nourani, N. 2008. IMU Aided 3D Visual Odometry for Car-Like Vehicles, Australasian Conference on Robotics and Automation.

Shi, J. y Tomasi, C. 1994. Good Features to track, IEEE Conference on Computer Vision and Pattern Recognition, pp. 593-600

Unibrain. 2010.
http://www.unibrain.com/Products/VisionImg/Fire_i_DC.htm