Desarrollo de un Sistema SLAM VISUAL con reconstrucción 3D monocular de marcas orientadas para un Humanoide

Juan M. IBARRA ZANNATHA, Eric HERNÁNDEZ, Rafael CISNEROS, Jorge E. LAVÍN Departamento de Control Automático Centro de Investigación y de Estudios Avanzados del IPN D.F. 07360. México

> José NEIRA Departamento de Informática e Ingeniería de Sistemas Centro Politécnico Superior Universidad de Zaragoza Zaragoza 50018, España

RESUMEN

En este trabajo se presentan los resultados de un sistema de localización y mapeo simultáneo desarrollado para un robot humanoide con base en información sensorial producida por un sistema de visión monocular (VSLAM). La medición de la ubicación de las marcas con respecto al robot se hace exclusivamente mediante un sistema de visión artificial (VA) el cual permite obtener su orientación además de su posición; en efecto, el método de reconstrucción 3D monocular desarrollado permite utilizar por vez primera en un sistema de SLAM información sobre la orientación de las marcas. Por su parte, el sistema SLAM implementado utiliza el sistema de estimación estocástica clásico basado en el Filtro de Kalman Extendido (FKE) el cual se aplica a los siguientes dos casos: i) Marcas 2D sin odometría; ii) Marcas 2D orientadas sin odometría. Se analizan los resultados obtenidos en dos experimentos aplicados a ambos casos: a) Caminata a lo largo de una línea recta aproximadamente paralela a una pared con marcas, y b) Caminata en bucle cerrado dentro de un claustro con cuatro paredes con marcas.

Palabras Claves: SALM, MonoSLAM, Marcas orientadas, Reconstrucción 3D, Localización de Robots, Filtro de Kalman.

1. INTRODUCCIÓN

Con el fin de navegar de forma autónoma en un entorno desconocido, un robot debe de ser capaz de construir una representación del mapa de dicho entorno y al mismo tiempo auto-localizarse dentro de este. A este problema se le conoce dentro de la comunidad de la robótica como SLAM (*Simultaneous Localization and Mapping*), cuya solución se considera como una pieza clave hacia la consecución de la autonomía del robot en cuestión. En los últimos años, el campo de SLAM ha proporcionado muchos algoritmos prácticos y entendibles para hacerle frente a este problema, con miras, en particular a aplicaciones dentro de la robótica móvil autónoma. La mayoría de estos algoritmos se basan en métodos de estimación Bayesiana como por ejemplo el Filtro de Kalman Extendido (FKE), Filtro de Partículas (FP), estimación Monte-Carlo (MC), entre otros [1], [2], [3].

En nuestra aplicación, se empleo el algoritmo clásico de SLAM-FKE. En efecto, existen en la literatura numerosos trabajos basados en este algoritmo, así como las pruebas de su convergencia y de su excelente desempeño [3]. Pero ¿Cómo es que surgió el problema del SLAM?; pues bien, a finales de los años 80 comienzan a aparecer los primeros sistemas que

intentan construir o actualizar un mapa del entorno con base en la información de los sensores exteroceptivos del robot, el cual se mueve dentro de un entorno desconocido. Los algoritmos utilizados desacoplaban el problema de la construcción del mapa y el de la localización del robot, de tal forma que construían el mapa con la suposición que la posición del robot es correcta y se localizan en el mapa suponiendo que éste es correcto. Sin embargo, pronto se descubre que la solución rigurosa del problema no es posible sin considerar ambos aspectos simultáneamente [4].

El planteamiento probabilístico para resolver este problema fue hecho en el marco del ICRA 1986 organizado por IEEE en San Francisco a raíz de las discusiones generadas por Peter Cheeseman, Jim Crowley, y H. F. Durrant-Whyte junto con otros investigadores como Raja Chatila, Oliver Faugeras y Randal Smith. Así, las bases estadísticas para describir la relación entre las marcas del entorno (*landmarks*) y la manipulación de la incertidumbre geométrica fueron establecidas en los trabajos pioneros de R. Smith y P. Cheeseman [2] así como de H. F. Durrant Whyte [3]. En donde se demuestra el hecho clave de que existe un alto grado de correlación entre la estimación de la localización de las diferentes marcas en un mapa y que, efectivamente, estas correlaciones crecerían con observaciones sucesivas [5].

Al mismo tiempo en Francia Avache y Faugeras desarrollaban los primeros trabajos en navegación visual, mientras que Crowley, Chatila y Laumond lo hacían con un sensor ultrasónico; ambas investigaciones empleaban algún tipo de algoritmo basado en el Filtro de Kalman. Estas dos líneas de investigación tenían mucho en común y pronto dieron lugar al trabajo seminal en este tema debido a R. Smith, M. Self y P. Cheeseman [2]. En este trabajo se puso de manifiesto que, cuando un robot móvil navega en un entorno desconocido capturando observaciones de la posición relativa de marcas en dicho entorno, las estimaciones de la posición de dichas marcas están todas necesariamente correlacionadas entre sí a causa del error común en la estimación de la localización del vehículo. La implicación de esto resultó de mucha importancia: una solución completa coherente para el problema combinado de localización y mapeo requiere un estado compuesto de la pose del vehículo y la posición de cada una de las marcas, el cual se actualiza después de la observación de cada una de las marcas. A su vez, esto requeriría que el estimador utilizara un vector de estados grande, del orden del número de marcas mantenidas en el mapa [5].

Desde hace más de diez años SLAM ha sido uno de los campos de investigación más activos en la robótica, así como, el tema principal dentro de las conferencias internacionales más importantes. Durante este tiempo, se han reportado excelentes trabajos por muchos investigadores en los cuales los temas han sido: i) SLAM en entornos grandes tanto interiores como exteriores, ii) SLAM multirrobot, iii) El problema de la asociación de datos en las diversas aplicaciones a fin de generar aplicaciones en tiempo real, entre otros. Pero hasta hace poco, todos estos sistemas requerían el uso de sensores de rango como lo es el sensor laser, por lo que predominaban la construcción de mapas 2D de entornos en un solo plano. Recientemente, sea considerado que es más difícil el tratar sistemas de SLAM los cuales emplean cámaras estándar como principal sensor de entrada, y se ha tomado como un reto dar solución a este problema [1].

El interés de utilizar cámaras en el SLAM se debe principalmente a que estas son mucho más baratas que los sensores de rango, además, proporciona información de textura de los elementos presentes en la escena independientemente de la distancia a la que se encuentren de la cámara. Además, estas representan la mejor solución en aplicaciones donde no es práctico colocar sensores pesados y abultados como por ejemplo en aplicaciones de odometría visual en robots móviles y en aplicaciones de egomotion para el seguimiento de personas o en operaciones de rescate, en las que la cámara se puede adaptar sobre el casco del rescatista [1]. Otra aplicación es el caso de los robots humanoides de pequeño formato como los utilizados en la categoría Kid Size de RoboCup (que es nuestro caso), los cuales, no tienen las capacidades mecánicas suficientes como para soportar un sistema con sensores de rango que hagan las mediciones utilizadas en los sistemas SLAM o que permitan tener una odometría con un mínimo de precisión; de ahí el interés de implementar un sistema de SLAM visual basado en una cámara para este tipo de robots. Más aún, al aplicar los métodos de reconstrucción 3D monoculares desarrollados por nuestro equipo [4], [5] podemos contar con un parámetro de orientación de las marcas que, a nuestro conocimiento, es la primera vez que se utiliza en un esquema de SLAM. Otro logro importante si se emplea sensores visuales como lo son las cámaras es que estas ofrecen tanto precisión en la estimación de los movimientos locales empleando información odométrica visual, como una detección robusta de elementos en lazo-cerrado cuando se han observado previamente partes del entorno que han sido vistas otra vez [1].

La madurez tanto de los algoritmos para SLAM como de las herramientas de procesamiento y análisis de imágenes para extraer información geométrica relevante de estas y el constante incremento en el desarrollo de los procesadores de las computadores y de la tecnología de las cámaras se ha conjuntado para dar como resultado el surgimiento de una nueva técnica llamada SLAM Visual o simplemente VSLAM, que se define como: *El proceso de crear mapas del entorno y de localizar al robot dentro de este usando exclusivamente información visual.*

En los últimos cinco años ha habido un aumento en las investigaciones dentro del campo de VSLAM, gracias al interés que han puesto tanto la comunidad de Robótica dedicada a SLAM utilizando sensores láser como la comunidad de visión artificial. Esto ha sido evidente debido a que gran parte del estado del arte actual de las investigaciones en localización y mapeo se basa principalmente o exclusivamente en el uso de cámaras como sensor exteroceptivo. Sistemas robóticos recientes que utilizan una sola cámara, sistemas estereoscópicos o cámaras omnidireccionales frecuentemente en combinación con odometría o sensores inerciales han mostrado una estimación muy precisa y confiable de la localización y de la construcción del mapa, lo que nos dice que el empleo de cámaras como sensor principal para aplicaciones de SLAM es altamente viable. También han demostrado su potencial para construir mapas detallados a gran escala como por ejemplo del tamaño de un edificio o de una cuadra de alguna ciudad. VSLAM se ha implementado en una gran variedad de plataformas operando en diferentes entornos como por ejemplo en la tierra, el aire o bajo el agua con la posibilidad de implementarse en tiempo real tanto para entornos pequeños como grandes [6] [7] [8]. Este algoritmo tiene la capacidad de guiar robots autónomos en su operación y exploración dentro de entornos grandes y complejos o dentro de entornos más limitados como por ejemplo en los hogares donde los robots de servicio de bajo costo son empleados. Sin embargo, VSLAM tiene otras aplicaciones potenciales pertenecientes a escenarios dentro de la robótica móvil tradicional, debido a la capacidad que nos da utilizar una cámara como sensor de posición 3D de propósito general [1].

Las investigaciones actuales dentro de VSLAM se han enfocado en el uso tanto de sistemas de visión monoculares como sistemas estereoscópicos para obtener información 3D del entorno. Se ha demostrado que los sistemas de VSALM monocular son viables para emplearse en entornos pequeños [8]. Mientras que los sistemas estereoscópicos (BI-SLAM) [7] y el SLAM visual jerárquico [6] se emplean para entornos grandes; y existe otras áreas dentro del VSLAM que se le denomina SLAM-multicámara [7], Wearable Robotics y Parametrización inversa de la profundidad (Inverse Depth Parametrization IDP) [10]. La mayoría de estos sistemas de VSLAM se basan esencialmente en el filtro de Kalman extendido estándar y solo varían en la técnica que utilizan para inicializar las características, dando solo información parcial de orientación provista por una cámara o en el tipo de puntos de interés extraídos de las imágenes (esquinas Harris, o Shi-Tomasi, scale-invariant feature transform SIFT, o alguna combinación de estas) [6]. En algunos otros trabajos también se consideran segmentos de características [9].

SLAM Monocular (Mono-SLAM)

Este resultado es la consolidación del trabajo realizado por A. J. Davison [8]. En los inicios del SLAM visual, para obtener una buena estimación de las marcas 3D, se requerían muchas observaciones de diferentes puntos, lo que dificultaba la inicialización de la marca. Esto se debía a que una nueva marca se incorporaba al estado con un retardo, por lo que, para obtener un estimado completo 3D se necesitaban varias observaciones antes de la inicialización. Además, en estos esquemas sólo pueden inicializar marcas que cuenten con suficiente paralaje, es decir, aquellas que están cerca de la cámara y situadas perpendicularmente a su trayectoria de movimiento; por lo tanto era necesario operar en escenarios interiores realizando movimientos laterales. Recientemente se han desarrollado métodos satisfactorios para explotar toda la información geométrica provista por la cámara. Estos son los métodos de inicialización de la marca sin retardo (en donde se mapean las marcas desde su primera observación), esto permitió la inclusión de las marcas con bajo paralaje, es decir aquellas que están alejadas o situadas cerca del eje de movimiento. Con lo cual se pueden realizar operaciones en entornos exteriores con trayectoria frontal.



Figura 1. Robot y Marcas empleadas

Parametrización inversa de la profundidad

Recientemente se desarrolló la técnica llamada Parametrización Inversa de la Profundidad, (Inverse Depth Parametrization) [10], con la cual se ha podido trazar correctamente cualquier marca en el mapa que comience de un punto cercano y termine en el infinito, es decir marcas con paralaje muy pequeño. Esta nueva parametrización para puntos característicos (landmarks) en un sistema de SLAM monocular permite una representación eficiente y precisa de la incertidumbre durante la inicialización retrasada de una nueva característica, pues posee un modelo casi lineal y con distribución gaussiana. Además, puede lidiar con características dentro de un gran rango de profundidad, aun cuando estas se encuentren lejos de la cámara y posean bajo paralaje. Mientras, la parametrización euclidiana presenta una alta incertidumbre en la estimación de la profundidad de puntos que posean bajo paralaje. Esto significa que la situación está mal caracterizada por la distribución Gaussiana que se asume en el FKE. El concepto en el que se basa esta parametrización es la inversa de la profundidad de la característica relativa a la localización de la cámara en la que fue vista por primera vez, con lo que se obtienen ecuaciones de medición con un alto grado de linealidad, lo cual mejora el desempeño del FKE, permitiendo obtener una mejor estimación tanto de la posición de la cámara como la de la característica.

En este artículo usamos una nueva técnica para extraer información de posición y orientación de las marcas que se encuentran en el entorno del robot. La ventaja principal de esta técnica en comparación con otras como lo son SIFT implementado con IDP como en [10] es que esta provee información de orientación de la marca con respecto a la posición actual del robot. Se muestra que esta información ayuda a mejorar el mapa final de la estimación de la trayectoria del robot y del mapa de su entorno [11] [12]. También se utilizo un sensor inercial IMU, con el cual solo podemos hasta hora estimar la orientación del robot. Mostraremos que esta información en conjunto con la información de posición y orientación de la marca nos ayudara a obtener una mejor estimación del mapa que las que se tienen hasta ahora.

Para poder llevar a cabo la implementación del algoritmo de visual SLAM basado en el FKE empleamos una plataforma robótica que se describe en [13]. Recientemente le agregamos a la plataforma una IMU comercial (3DM-GX1) creada por MicroStrain, la cual combina tres giróscopos con tres acelerómetros DC ortogonales y tres magnetómetros ortogonales junto con un mico-controlador fijo a la salida, el cual nos da la orientación de esta tanto en entornos estáticos como dinámicos. Finalmente la organización del trabajo es: En la sección 2 se presenta el sistema de percepción visual

incluyendo el método de reconstrucción 3D monocular que se utiliza para hacer la medición de la posición relativa robotmarcas así como la orientación de estas últimas. La sección 3 se dedica a la presentación de la implementación del sistema de SLAM visual basado en el algoritmo FKE, mientras que en la sección 4 se presentan los resultados obtenidos de los experimentos realizados, mismos que se comentan en la sección 5.

2. SISTEMA DE PERCEPCIÓN

El humanoide Robonova-I fue equipado con una cámara que permite captar imágenes y enviarlas a una computadora huésped en donde se hace el procesamiento adecuado para estimar la posición relativa de las marcas observadas con respecto a la cámara del robot. Esta cámara fue montada sobre un mecanismo de 2 gdl (*pan-and-tilt*) que hace las veces del cuello del robot y que permite orientar la cámara en azimut y elevación. Las marcas utilizadas son motivos cuadrados de dimensión conocida cuyo aspecto puede verse en la fotografía de la Figura 1.

Cabe mencionar que el proceso de estimación correspondiente equivale a una reconstrucción 3D monocular. Es decir, se trata de efectuar el cálculo de una posición 3D a partir de la información 2D contenida en una sola imagen. En efecto, la posición relativa de una marca con respecto al referencial asociado a la cámara del robot está determinada por tres parámetros (ρ , ϕ , θ), los cuales representan respectivamente la longitud ρ de un segmento de recta que une los referenciales asociados a la marca observada y a la cámara (medida en el plano horizontal), el ángulo ϕ que dicho segmento hace con el eje óptico de la cámara y el ángulo θ que forma con el plano en el que se encuentra la marca, denominado también orientación de la marca, tal como se muestra en la Figura 2.



Figura 2. Posición relativa entre el robot y las marcas dada por (ϕ, ρ) así como la orientación θ de las marcas.

A partir de la posición de una marca con respecto a la cámara (ρ, ϕ, θ) puede calcularse la posición de dicha marca ahora con respecto al robot conociendo el modelo cinemático cámararobot, el cual depende básicamente de dos variables articulares: el movimiento acimutal y de elevación del cuello del robot. Para efecto de simplificación de los cálculos, pero sin pérdida de generalidad del método, se asume que el robot no utiliza su movimiento de elevación y todo lo mira con el eje óptico de su cámara permanentemente horizontal, con lo cual la posición relativa de una marca con respecto al referencial asociado al robot está dada por tres coordenadas: (ρ, ϕ, θ) , en donde ϕ se calcula a partir de ϕ , el ángulo entre el eje óptico y el segmento de recta que une el referencial de la cámara con la marca en cuestión, y de ψ , el ángulo acimutal del cuello del robot.

Para hacer el cálculo de los tres parámetros (ρ , ϕ , θ) que determinan la posición de una marca cuadrada de dimensiones conocidas expresada en el referencial de la cámara hemos desarrollado dos diferentes métodos de reconstrucción 3D monocular. El primero se basa en la estimación de la distorsión geométrica que introduce la proyección en perspectiva de la marca cuadrada de dimensiones conocidas, mientras que el segundo utiliza la Homografía inducida por dicho proceso de proyección.

Reconstrucción 3D monocular basada en la perspectiva

Si bien es posible hacer la reconstrucción 3D monocular de un entorno completamente desconocido aunque relativamente estructurado, los métodos existentes son bastante costosos desde un punto de vista computacional. Por ejemplo, Kanatani propuso un método basado en los puntos y líneas de fuga que aparecen en las imágenes de entornos estructurados (por ejemplo interiores en edificios con paredes verticales y suelos perfectamente planos), tratando de obtener la posición 3D de puntos en la imagen a partir de la obtención previa del modelo de distorsión en perspectiva basado en el análisis de los puntos y las líneas de fuga [14], [15], [16].

A fin de aligerar el costo computacional de la reconstrucción 3D monocular hemos decidido hacer algunas suposiciones en cuanto al conocimiento a priori del entorno. Sin pérdida de generalidad suponemos que el robot evoluciona en un entorno estructurado (interiores de edificios con paredes verticales y suelos planos) y que se analizarán exclusivamente imágenes de las paredes de su entorno; además, supondremos que en las paredes se tienen marcas cuadradas de dimensiones conocidas. Finalmente requerimos de una tercera suposición: las marcas se encuentran a la misma altura que la cámara del robot; es decir, el centro geométrico de las marcas se encuentran a la misma altura que el eje óptico de la cámara del robot. La primera suposición, que el robot se desenvuelva en interiores bien estructurados, no entraña restricción alguna; que las marcas sean de dimensión conocida es lo que permite determinar su posición 3D con respecto al referencial de la cámara con cálculos simples a partir de una sola imagen. Finalmente, que las marcas y la cámara se encuentren a la misma altura permite mantener simples los cálculos pues sólo se debe calcular un ángulo de rotación acimutal entre el plano de la pared y el plano imagen pues no habría ángulo de elevación [17], [12]



Figura 3. Efecto de la distorsión de proyección en perspectiva

Así, para calcular los tres parámetros (ρ , ϕ , θ) que determinan la posición de una marca cuadrada de dimensiones conocidas expresada en el referencial de la cámara, el sistema de visión dispone como entradas los valores de las coordenadas imagen de las cuatro esquinas de una marca cuadrada de dimensiones conocidas. Dicha imagen se verá distorsionada por efectos de la proyección en perspectiva que caracteriza a la cámara, tal como se ilustra en el diagrama de la Figura 3. Haciendo la medición de dicha distorsión y empleando el modelo de proyección en perspectiva inversa es posible encontrar la posición relativa de una marca con respecto al referencial asociado a la cámara dada por (ρ , ϕ , θ).

Una vez que se tiene el conjunto de cuatro puntos de interés (esquinas de un patrón cuadrado en la pared) en coordenadas imagen, se propone la aplicación de una transformación geométrica que permita representarlos como el vector (ρ , ϕ , θ) que necesita el proceso de SLAM. Considerando que se conoce la distancia focal f de la cámara utilizada, que ésta ha sido previamente calibrada para conocer sus parámetros intrínsecos y que la distorsión radial introducida por la cámara ya ha sido corregida, se puede obtener la información 3D del punto de interés a partir de la información visual, expresada en el referencial de la cámara, mediante el análisis de la deformación con el modelo de proyección en perspectiva [18], [19]. Esto es, a partir del conocimiento de algunos parámetros como la longitud de los lados de los cuadrados (E) y de la distancia focal de la cámara, midiendo las coordenadas imagen de las esquinas del cuadrado es posible reconstruir las coordenadas 3D de ese cuadrado expresadas en el referencial imagen. A continuación se desarrollan las ecuaciones correspondientes a esta reconstrucción 3D.

Los puntos de interés serán los puntos centrales de cada uno de los cuadrados de color que decoran cada pared. Como la paredes son verticales pero el ángulo entre la pared y el eje óptico de la cámara puede ser cualquiera (siempre y cuando alguno de los cuadrados permanezca visible), dicho cuadrado se deforma de modo que sus aristas verticales permanecen verticales y paralelas mientras que sus aristas horizontales se ven deformadas por la proyección en perspectiva que caracteriza al proceso de formación de imágenes en la cámara, con rectas que se cortan en un punto de fuga, dejando de ser paralelas. Por facilidad en la derivación del modelo pero sin pérdida de generalidad, se supone, además, que el centro de los cuadrados en las paredes está a la misma altura sobre el suelo que el eje óptico de la cámara del robot. Dicha altura es uno de los parámetros extrínsecos de la cámara obtenidos en la fase de calibración. En la Figura 3 se muestra un esquema de la manera en que aparece dentro de la imagen una de las marcas de la pared. En dicha figura las esquinas de cada uno de los cuadros se han denominado A, B, C y D, de modo que las coordenadas imagen de cada una de estas esquinas se denotan (x_A, y_A) , (x_B, y_B) , (x_C, y_C) y (x_D, y_D) respectivamente. Debido a que, bajo las suposiciones hechas, en la imagen el paralelismo de las líneas verticales se mantiene se tiene que:

$$x_A = x_C = x_L$$

$$x_B = x_D = x_R$$
(1)

En donde las coordenadas x_L y x_R se refieren, respectivamente, al lado izquierdo y al derecho del cuadrado analizado. En la Figura 3 se muestra, además, la longitud en la imagen de

dichos lados denotadas h_L y h_R , las cuales se calculan de la siguiente manera:

$$h_L = y_C - y_A$$

$$h_R = y_D - y_B$$
 (2)

Primeramente se debe calcular la posición relativa de las marcas con respecto al referencial de la cámara, de acuerdo con el diagrama geométrico mostrado en la Figura 4. Ahí aparece una vista en planta del robot y de una de las paredes situada a su izquierda, de modo que se puedan situar las coordenadas de las esquinas de una marca cuadrada con respecto al referencial de la cámara, a fin de obtener una expresión para los parámetros que definen la posición relativa de la marca con respecto a la cámara.



Figura 4. Posición relativa de una marca con respecto a la cámara.

Sean X_M y Z_M las coordenadas del punto central de la marca expresadas en el referencial de la cámara, sean X_L y Z_L las coordenadas del lado izquierdo del cuadrado y, finalmente, X_R y Z_R las del lado derecho de dicho cuadrado. Entonces, las coordenadas buscadas, o sea las del punto central del cuadrado, se calculan como sigue:

$$X_{M} = \frac{X_{R} - X_{L}}{2}$$
 y $Z_{M} = \frac{Z_{R} - Z_{L}}{2}$ (3)

Mientras que la posición relativa buscada (ρ, ϕ) queda expresada en términos de las coordenadas del punto medio del cuadrado de la siguiente manera:

$$\rho = \sqrt{X_M^2 + Z_M^2} \quad \text{y} \quad \phi = \arctan \frac{Z_M}{X_M} \tag{4}$$

A partir del mismo diagrama y con las mismas ecuaciones se puede calcular también el ángulo θ que forma la pared vista con respecto al plano imagen mediante la siguiente expresión:

$$\theta = \arctan \frac{Z_R - Z_L}{X_R - X_L} \tag{5}$$

Ahora se procede a calcular el valor de todas estas coordenadas expresadas en el referencial de la cámara pero utilizando las coordenadas imagen de las esquinas extraídas de la imagen, que son los únicos datos cuyos valores se conocen. Para ello se utiliza el modelo de proyección en perspectiva definido como sigue:

$$\frac{X_L}{Z_L - f} = \frac{x_L}{f} \qquad \frac{X_R}{Z_R - f} = \frac{x_R}{f}$$
$$\frac{E}{Z_L - f} = \frac{h_L}{f} \qquad \frac{E}{Z_R - f} = \frac{h_R}{f}$$
(6)

A partir de estas expresiones se pueden despejar los valores de X_L , X_R , Z_L y Z_R en función de los datos conocidos, obteniéndose:

$$X_{L} = \frac{Z_{L} - f}{f} x_{L} \qquad X_{R} = \frac{Z_{R} - f}{f} x_{R}$$
$$Z_{L} = f\left(\frac{E}{h_{L}} + 1\right) \qquad Z_{R} = f\left(\frac{E}{h_{R}} + 1\right)$$
(7)

Entonces, midiendo en el plano imagen las coordenadas de las cuatro esquinas (x_A, y_A) , (x_B, y_B) , (x_C, y_C) y (x_D, y_D) , conociendo los valores *E* y *f* y aplicando las ecuaciones (1) y (2) pueden calcularse los valores de Z_L y Z_R mediante (7) y luego se calculan X_L y X_R usando la misma ecuación. Con esto se está en medida de calcular X_M y Z_M con (3) y, finalmente se calcula la posición relativa buscada (ρ, ϕ) con la ecuación (4) y aun el ángulo θ que forman la pared y el plano imagen usando la ecuación (5).

Este método de reconstrucción 3D [12] tiene una leve restricción: las marcas deben estar a la misma altura que la cámara para evitar que las líneas verticales de la marca pierdan su paralelismo. Para salvar dicha restricción se generó otro método basado en la identificación de la homografía entre el plano de la pared donde se encuentran las marcas y el plano imagen [20]. Las dos alternativas del sistema de percepción descrito se implementaron en Matlab, donde se realizó una interfaz de usuario que permite el monitoreo de su funcionamiento así como la teleoperación del humanoide, la cual se muestra en la Figura 5.



Figura 5. Interfaz para la teleoperación del robot y para la adquisición y procesamiento de datos.

Con respecto al referencial del robot

Aplicando ahora el modelo cinemático de la cabeza del robot, es decir del mecanismo *pan-and-tilt* de 2 gdl, basado en la medición del ángulo ψ (*pan*) ya se puede calcular el ángulo φ .

En la Figura 6 se muestra un diagrama de la cabeza con los movimientos *pan-and-tilt* que fue desarrollada para orientar la

cámara de visión del robot. El movimiento de rotación alrededor del eje vertical (*pan*) se genera de manera directa mediante un servo miniatura HS-55 de HiTec, mientras que el movimiento de elevación (*tilt*) se genera de manera indirecta mediante un mecanismo de 4 barras cuyo eslabón de entrada es actuado por un servo miniatura S3114 de Futaba. En [21] se encuentra una descripción completa de este sistema así como el análisis cinemático de sus movimientos. Con base en dicho análisis se procedió a generar un modelo cinemático de la cabeza incluyendo sus movimientos (*pan-and-tilt*), que permita la transformación de la posición 3D de los puntos de interés representados en el referencial de la cámara (ρ , ϕ , θ) a fin de obtener su representación en el referencial del robot, localizado en el centro de masa del eslabón del cuerpo [22].



Figura 6. Cabeza con 2 gdl (*pan-and-tilt*) para orientar la cámara.

Para ello se considera el diagrama cinemático mostrado en la Figura 7, en donde aparecen los referenciales R_B , R_P , R_T , R_I y R_C , que son respectivamente los referenciales asociados al robot, a la articulación de rotación alrededor del eje vertical (*pan*), a la articulación de elevación (*tilt*), a la imagen y a la cámara, además de los referenciales R_U , R_N y R_S , que sirven como auxiliares: R_U es un referencial ubicado en el cruce de la línea vertical del cuerpo y la línea que une ambas extremidades superiores, R_N y R_S se ubican en la parte superior del robot, justo en la ubicación del soporte del mecanismo *pan-and-tilt*.



Figura 7. Diagrama cinemático de la cabeza y de la cámara.

De especial interés son los referenciales del robot y de la cámara. El referencial del robot orienta el eje *X* en la dirección frontal al robot, mientras que el eje *Y* en la dirección vertical. El referencial de la cámara se orienta de acuerdo al modelo de proyección en perspectiva manejado en [23]. Con base en estos diagramas y siguiendo la convención de Denavit-Hartenberg se obtuvo la siguiente matriz de transformación, donde s_i y c_i representan la abreviatura de $sin(\phi_i)$ y $cos(\phi_i)$, siendo i = PAN ó *TILT*. Cabe mencionar que ϕ_{TILT} se obtiene a partir del análisis del mecanismo de cuatro barras anteriormente explicado. En cuanto a las variables de longitud su valor se puede extraer directamente de las cotas mostradas en la Figura 7.

$$T_{B}^{C} = \begin{bmatrix} s_{PAN} & c_{PAN} s_{TILT} & c_{PAN} c_{TILT} & l_{IC} c_{PAN} c_{TILT} + l_{PT} c_{PAN} \\ 0 & -c_{TILT} & s_{TILT} & l_{IC} s_{TILT} + l_{BU} + l_{UN} + l_{SP} \\ c_{PAN} & -s_{PAN} s_{TILT} & -s_{PAN} c_{TILT} & -l_{IC} s_{PAN} c_{TILT} - l_{PT} s_{PAN} \\ 0 & 0 & 0 & 1 \end{bmatrix}$$
(8)

Con este modelo cualquier punto tridimensional X_C , Y_C , Z_C en el referencial de la cámara se transforma en un punto tridimensional X_B , Y_B , Z_B en el referencial del robot, pudiéndose obtener la posición relativa buscada en coordenadas polares pero con respecto al vehículo (robot) y no la cámara, información que es necesaria para SLAM:

 $\begin{bmatrix} X_B \\ Y_B \\ Z_B \end{bmatrix} = T_B^C \begin{bmatrix} X_C \\ Y_C \\ Z_B \end{bmatrix}$ (9)

3. ALGORITMO FKE IMPLEMENTADO

En esta sección se describe la implementación del algoritmo de SLAM EKF con el cual se estima la posición del robot dentro de su entorno y, además, se construye simultáneamente un mapa del mismo, que posteriormente el robot usará para navegar. El problema de construcción de mapas consiste en los siguientes tres pasos: *i*) Sensar el entorno del vehículo en el instante *k* con el sensor embarcado (cámara); *ii*) Representar los datos del sensor; y *iii*) Agregar las observaciones hechas en el instante *k* con las que ya se tenían en el instante previo k-1.

Una aproximación simple de la construcción de mapas depende de la estimación del la posición actual del robot (odometría, *dead-reckoning*). Sin embargo, esto no es muy apropiado para navegaciones en entornos grandes debido a la deriva (*drift*), la cual se incrementa con el tiempo y produce una mala estimación de la posición del robot y del mapa. Para evitar esta deriva y debido a que el humanoide utilizado carece de sensores de odometría, se implementará un SLAM-EKF sin odometría.

La técnica de SLAM basada en EKF se caracteriza por la existencia de un vector de estados aumentado en tiempo discreto, formado por la localización del robot y la de los objetos del mapa, los cuales se estiman recursivamente a partir de las observaciones disponibles de los sensores en el tiempo k, y de un modelo de movimiento del robot entre los instantes k-1 y k. Dentro de este marco, la incertidumbre se representa por funciones de densidad de probabilidad asociadas con el vector de estado, el modelo de movimiento, y las observaciones del sensor.

SLAM basado en EKF

Dentro del SLAM el entorno se modela como un conjunto de características geométricas (segmentos de recta, planos, puntos) en función del tipo de entorno de interés y del sensor utilizado. En nuestro caso se implementó un sistema de visión para obtener ya sea puntos 2D o puntos 2D orientados correspondientes a marcas ubicadas en la pared. En la aproximación estándar de SLAM-EKF, la información del entorno se relaciona con un conjunto de elementos $\{B,R,F_1,...,F_n\}$ que se representan por un mapa $M^B = (\hat{\mathbf{x}}^B, \mathbf{P}^B)$, donde \mathbf{x}^B es un vector de estados estocástico con media estimada $\hat{\mathbf{x}}^B$ y un error de covarianza estimado \mathbf{P}^B :

$$\hat{\mathbf{x}}^{B} = E[\mathbf{x}^{B}] = \begin{bmatrix} \hat{\mathbf{x}}^{B} \\ \vdots \\ \hat{\mathbf{x}}^{B} \\ \hat{\mathbf{x}}^{B} \end{bmatrix}$$
$$\mathbf{P}^{B} = E[(\mathbf{x}^{B} - \hat{\mathbf{x}}^{B})(\mathbf{x}^{B} - \hat{\mathbf{x}}^{B})^{T}] = \begin{bmatrix} \mathbf{P}^{B}_{R} & \cdots & \mathbf{P}^{B}_{RF_{n}} \\ \vdots & \ddots & \vdots \\ \mathbf{P}^{B}_{F_{n}R} & \cdots & \mathbf{P}^{B}_{F_{n}} \end{bmatrix}$$
(10)

El vector $\hat{\mathbf{x}}^{B}$ contiene la localización estimada del robot *R* y de las *n* marcas del entorno F_i , *i*=1,2...*n*, todo con respecto al referencial base *B*. En nuestra aplicación el vector de estados tendrá dos versiones, puntos 2D o puntos 2D orientados; para las cuales está constituido respectivamente por:

$$\hat{\mathbf{x}}_R = \begin{bmatrix} x_R & y_R & \phi_R \end{bmatrix}^T \quad \& \quad \hat{\mathbf{x}}_{F_i}^R = \begin{bmatrix} x_{F_i} & y_{F_i} \end{bmatrix}^T \tag{11}$$

$$\hat{\mathbf{x}}_{R} = \begin{bmatrix} x_{R} & y_{R} & \phi_{R} \end{bmatrix}^{T} & \& \quad \hat{\mathbf{x}}_{F_{i}}^{R} = \begin{bmatrix} x_{F_{i}} & y_{F_{i}} & \phi_{F_{i}} \end{bmatrix}^{T} (12)$$

Los elementos de la diagonal de la matriz \mathbf{P}^{B} representan el error de covarianza estimado de las diferentes características del vector de estados; mientras que los elementos fuera de la diagonal representan la correlación entre dichos estados. La estimación recursiva de los dos primeros momentos de las funciones de densidad probabilística de \mathbf{x}^{B} se desarrolla siguiendo el algoritmo de SLAM mostrado en la Tabla 1. En este algoritmo el mapa se inicializa empleando la ubicación inicial del robot como referencial base, con esto se conoce la ubicación del robot en el instante k=0. Antes de que se mueva el robot, se hace la adquisición de una imagen y se inicializa el mapa con las características observadas, con esto se maximiza la precisión del mapa resultante. Posteriormente se llevan a cabo, interactivamente, la etapa de cálculo o predicción del movimiento sin usar odometría {pre} y la actualización del mapa {act} utilizando las mediciones visuales.

Inicialización

La creación de un nuevo mapa estocástico requiere de un referencial base. Una práctica común en SLAM es asignar la posición de partida del robot: $B=R_0$, para que la estimación de posición se inicie sin incertidumbre alguna [24], [25] de modo que:

$$\hat{\mathbf{x}}_{0}^{B} = \hat{\mathbf{x}}_{R_{0}}^{B} = 0$$
; $\mathbf{P}_{0}^{B} = \mathbf{P}_{R_{0}}^{B} = 0$ (13)

$$\begin{aligned} \mathbf{x}_{0}^{B} &= 0; \ \mathbf{P}_{0}^{B} &= 0 \ \{\text{inicialización del Mapa}\} \\ \begin{bmatrix} \mathbf{z}_{0}, \mathbf{R}_{0} \end{bmatrix} &= \text{hacer_medición} \\ \begin{bmatrix} \mathbf{x}_{0}^{B}, \mathbf{P}_{0}^{B} \end{bmatrix} &= \text{agregar_rasgo} \left(\mathbf{x}_{0}^{B}, \mathbf{P}_{0}^{B}, \mathbf{z}_{0}, \mathbf{R}_{0} \right) \\ \text{for } k &= 1 \ \text{to steps do} \\ \begin{bmatrix} \mathbf{x}_{R_{k}^{B-1}}^{R}, \mathbf{Q}_{k} \end{bmatrix} &= \text{hacer_odometría} \\ \begin{bmatrix} \mathbf{x}_{k,k-1}^{B}, \mathbf{P}_{k/k-1}^{B} \end{bmatrix} &= \text{calc_mvto} \left(\mathbf{x}_{k-1}^{B}, \mathbf{P}_{k-1}^{B}, \mathbf{x}_{R_{k}}^{R_{k-1}}, \mathbf{Q}_{k} \right) \ \{ \text{pre} \} \\ \begin{bmatrix} \mathbf{z}_{k}, \mathbf{R}_{k} \end{bmatrix} &= \text{hacer_medición} \\ \\ H_{k} &= \text{asociación_datos} \left(\mathbf{x}_{k/k-1}^{B}, \mathbf{P}_{k/k-1}^{B}, \mathbf{z}_{k}, \mathbf{R}_{k} \right) \\ \begin{bmatrix} \mathbf{x}_{k}^{B}, \mathbf{P}_{k}^{B} \end{bmatrix} &= \text{act_mapa} \left(\mathbf{x}_{k/k-1}^{B}, \mathbf{P}_{k/k-1}^{B}, \mathbf{z}_{k}, \mathbf{R}_{k}, H_{k} \right) \ \{ \text{act} \} \\ \begin{bmatrix} \mathbf{x}_{k}^{B}, \mathbf{P}_{k}^{B} \end{bmatrix} &= \text{agregar_nuevo_rasgo} \left(\mathbf{x}_{k}^{B}, \mathbf{P}_{k}^{B}, \mathbf{z}_{k}, \mathbf{R}_{k}, H_{k} \right) \ \} \\ \text{end for} \end{aligned}$$

Tabla 1. Algoritmo de Slam-EKF utilizado.

Ahora, utilizando la función hacer_medición creada en Matlab, se hace una búsqueda en la Base de Datos para calcular el vector de observación $\mathbf{z}_{k,i}$ con i=1,2,...s, donde *s* es el número de características observadas. Para los casos de marcas 2D simples y 2D orientadas, este vector tiene la siguiente forma:

$$\mathbf{z}_{k,i} = \begin{bmatrix} x_{F_i} & y_{F_i} \end{bmatrix}^T \quad \& \quad \mathbf{z}_{k,i} = \begin{bmatrix} x_{F_i} & y_{F_i} & \theta_{F_i} \end{bmatrix}^T \quad (14)$$

Aplicando ahora la Ec. (10) se obtienen las observaciones expresadas en el referencial robot y se crea la matriz de covarianza para cada observación que codifica la incertidumbre introducida por el sistema de visión. Al no tener todos los parámetros de su expresión analítica, estas matrices se inicializan experimentalmente con una incertidumbre en profundidad de σ_{ρ} =0.05 m. y lateral de σ_{ϕ} = σ_{θ} = 2° . Así, las matrices de covarianza para los casos de marcas 2D simples y 2D orientadas son:

$$\mathbf{R}_{k,i} = \begin{bmatrix} \sigma_{\rho}^{2} & 0 \\ 0 & \sigma_{\varphi}^{2} \end{bmatrix}; \mathbf{R}_{k,i} = \begin{bmatrix} \sigma_{\rho}^{2} & 0 & 0 \\ 0 & \sigma_{\varphi}^{2} & 0 \\ 0 & 0 & \sigma_{\theta}^{2} \end{bmatrix}$$
(15)

Con el número de observaciones y la identificación de las marcas de acuerdo con su posición en la BD ya se sabe que marca ha sido observada, lo cual será de utilidad en la fase de Asociación de datos. Finalmente, con la función agregar_rasgo desarrollada en Matlab se inicializa el mapa con las marcas observadas con referencia a la posición inicial del robot.

Etapa de predicción

El sistema odométrico del robot permite estimar el desplazamiento del robot de la posición k-1 a la k:

$$\mathbf{X}_{R_k}^{R_{k-1}} = \hat{\mathbf{X}}_{R_k}^{R_{k-1}} + \mathbf{V}_k \tag{16}$$

En donde $\hat{\mathbf{x}}_{R_k}^{R_{k-1}}$ es la transformación relativa estimada entre los puntos *k*-1 y *k*, mientras que \mathbf{v}_k es ruido blanco de media cero

y covarianza \mathbf{Q}_k . En nuestro caso no se dispone de odometría, $\hat{\mathbf{x}}_{n-1}^{R_{k-1}} = [0]$

 $\hat{\mathbf{x}}_{R_k}^{R_{k-1}} = [0]$, pero se conoce aproximadamente el avance obtenido con un paso del robot y se tiene una estimación experimental de la incertidumbre de esta aproximación: $\sigma_{\phi}=5^{\circ}$ y $\sigma_x=\sigma_y=0.075$ m., con lo cual se construye \mathbf{Q}_k :

$$\mathbf{Q}_{k} = \begin{bmatrix} \sigma_{x}^{2} & 0 & 0\\ 0 & \sigma_{y}^{2} & 0\\ 0 & 0 & \sigma_{\phi}^{2} \end{bmatrix}$$
(17)

Ahora, dado $M_{k-1}^{B} = (\hat{x}_{k-1}^{B}, P_{k-1}^{B})$, el mapa en el instante k-1, su predicción para el instante k, después del próximo movimiento del robot será:

$$\hat{\mathbf{x}}_{k/k-1}^{B} = \begin{bmatrix} \hat{\mathbf{x}}_{R_{k-1}}^{B} \bigoplus \hat{\mathbf{x}}_{R_{k}}^{B} \\ \hat{\mathbf{x}}_{F_{1,k-1}}^{B} \\ \vdots \\ \hat{\mathbf{x}}_{F_{m,k-1}}^{B} \end{bmatrix}$$
$$\mathbf{P}_{k/k-1}^{B} \cong \mathbf{F}_{k} \mathbf{P}_{k-1}^{B} \mathbf{F}_{k}^{T} + \mathbf{G}_{k} \mathbf{Q}_{k} \mathbf{G}_{k}^{T}$$

En donde \oplus es la composición de transformaciones y

$$\mathbf{F}_{k} = \frac{\partial \mathbf{x}_{k/k-1}^{B}}{\partial \mathbf{x}_{k-1}^{B}} \bigg|_{\left(\hat{\mathbf{x}}_{R_{k-1}}^{B}, \hat{\mathbf{x}}_{R_{k}}^{R_{k-1}}\right)} = \begin{bmatrix} \mathbf{J}_{1\oplus} \left\{ \hat{\mathbf{x}}_{R_{k-1}}^{B}, \hat{\mathbf{x}}_{R_{k}}^{R_{k-1}} \right\} & 0 & \cdots & 0 \\ 0 & \mathbf{I} & & \\ \vdots & \ddots & \vdots \\ 0 & & \cdots & \mathbf{I} \end{bmatrix}$$
$$\mathbf{G}_{k} = \frac{\partial \mathbf{x}_{k/k-1}^{B}}{\partial \mathbf{x}_{R_{k}}^{R_{k-1}}} \bigg|_{\left(\hat{\mathbf{x}}_{R_{k-1}}^{B}, \hat{\mathbf{x}}_{R_{k}}^{R_{k-1}}\right)} = \begin{bmatrix} \mathbf{J}_{2\oplus} \left\{ \hat{\mathbf{x}}_{R_{k-1}}^{B}, \hat{\mathbf{x}}_{R_{k}}^{R_{k-1}} \right\} \\ 0 \\ \vdots \\ 0 \end{bmatrix}$$

En donde $J_{1\oplus}$ y $J_{2\oplus}$ son los Jacobianos de las transformaciones. Este procedimiento se programó en Matlab como la función predicción_EKF.

Asociación de datos

Cada vez que el robot avanza y hace una nueva medición se obtiene un conjunto de medidas $\mathbf{z}_{k,i}$ correspondientes a *m* características del entorno E_i , i=1,2,...m. Ahora, la asociación de datos consiste en determinar el origen de cada una de estas medidas en términos de las *n* características del mapa F_j , j=1,...n. con lo cual se obtiene la hipótesis siguiente:

$$H_k = [j_1, j_2, ..., j_m]$$
 (18)

que asocia cada medida $\mathbf{z}_{k,i}$ con su correspondiente características en el mapa F_{ji} . Para ello, se realiza una predicción de la medida que cada característica debería generar y se compara con la medida actual hecha por el sistema de visión del robot. Esta comparación puede hacerse con base en muchos criterios: el vecino más cercano, mínima distancia de Mahalanobis, o con algoritmoas más fiables pero complejos como el JCBB (*Joint Compatibility Branch and Bound*) [2], [6]. En nuestro caso esto no es necesario pues las marcas son todas distintas pues cada una utiliza cuadritos de un color

diferente (rojo, verde, azul y negro) como se puede apreciar en la Figura 5; por tanto, cada vez que se observa una marca se sabe de qué marca se trata.

Actualización del mapa

Una vez que tienen las correspondencias para las medidas z_k se emplean para mejorar la estimación del vector de estados con la ecuación de actualización del filtro EKF como sigue:

$$\mathbf{x}_{k}^{B} = \mathbf{\hat{x}}_{k/k-1}^{B} + \mathbf{K}_{H_{k}} \mathbf{v}_{H_{k}}$$

$$\mathbf{K}_{H_{k}} = \mathbf{P}_{k/k-1}^{B} \mathbf{H}_{H_{k}}^{T} \mathbf{S}_{H_{k}}^{-1}$$
(19)

En donde K_{Hk} es la ganancia del filtro. Finalmente, la covarianza del error de estimación del vector de estado es:

$$\mathbf{P}_{k}^{B} = \left(\mathbf{I} - \mathbf{K}_{H_{k}} \mathbf{H}_{H_{k}}\right) \mathbf{P}_{k/k-1}^{B}$$
$$= \left(\mathbf{I} - \mathbf{K}_{H_{k}} \mathbf{H}_{H_{k}}\right) \mathbf{P}_{k/k-1}^{B} \left(\mathbf{I} - \mathbf{K}_{H_{k}} \mathbf{H}_{H_{k}}\right)^{T} \qquad (20)$$
$$+ \mathbf{K}_{H_{k}} \mathbf{H}_{H_{k}} \mathbf{K}_{H_{k}}^{T}$$

Esta parte del algoritmo se incluyó en la función actualización_EKF y se usa en la sección siguiente. Agregar nuevos datos

Las medidas a las que no se les encuentra alguna correspondencia con características del mapa mediante la asociación de datos se consideran como observaciones nuevas y se agregan directamente al vector de estados mediante la transformación relativa entre el robot R_k y la nueva característica observada E y x_k^B se actualiza como sigue:

$$\mathbf{x}_{k}^{B} = \begin{bmatrix} \mathbf{x}_{R_{k}}^{B} \\ \vdots \\ \mathbf{x}_{F_{n,k}}^{B} \end{bmatrix} \Rightarrow \mathbf{x}_{k+}^{B} = \begin{vmatrix} \mathbf{x}_{R_{k}}^{B} \\ \vdots \\ \mathbf{x}_{F_{n,k}}^{B} \\ \mathbf{x}_{E_{k}}^{B} \end{vmatrix} = \begin{vmatrix} \mathbf{x}_{R_{k}}^{B} \\ \vdots \\ \mathbf{x}_{F_{n,k}}^{B} \\ \mathbf{x}_{E_{k}}^{B} \\ \mathbf{x}_{E_{k}}^{B} \oplus \mathbf{x}_{E}^{R_{k}} \end{vmatrix}$$
(21)

Esto se hace con la función desarrollada en Matlab agregar_nuevo_rasgo. Finalmente, la actualización de la matriz de covarianza P^{B}_{k+} se calcula usando la linealización de la Ec. (21).



Figura 8. Primer experimento: Caminata a lo largo de una pared con marcas.

4. RESLTADOS EXPERIMENTALES

Para evaluar el algoritmo de SLAM Visual propuesto en este trabajo se realizaron cuatro experimentos en los que no se considera ningún tipo de odometría: *i*) Caminata a lo largo de una línea recta aproximadamente paralela a una pared con marcas 2D; y *ii*) utilizando marcas 2D orientadas. *iii*) Caminata en bucle cerrado dentro de un claustro con cuatro paredes con marcas 2D; y *iv*) utilizando marcas 2D orientadas.

En la Figura 8 se presenta una fotografía correspondiente al primer par de experimentos, mientras que en las Figuras 9 y 10

se muestran los resultados de dichos experimentos; se trata del mapa del entorno obtenido por el algoritmo de SLAM visual EKF implementado para dos de los casos considerados: marcas 2D y marcas 2D orientadas. Por su parte, en la Figura 11 se muestra una fotografía del segundo grupo de experimentos, caminata en bucle cerrado, acompañada con el resultado del mapa obtenido y de la localización del robot para dos de los casos considerados: marcas 2D orientadas (Figura 13). Nótese que en ambas fotografías aparecen las huellas del robot en cada uno de los puntos a partir de los cuales se hace la medición visual de la posición relativa marca-robot, 10 pasos en el primer grupo de experimentos y 49 pasos en el segundo, pudiendo apreciarse que el robot no se mueve de manera precisa a lo largo de una línea recta, sino que avanza con algún error sobre dicha línea.



Figura 9. Mapa y trayectoria del robot en el caso de marcas 2D, del primer experimento.



Figura 10. Mapa y trayectoria del robot para el caso de marcas 2D orientadas, del primer experimento.

Caminata a lo largo de una Línea Recta

En el primer grupo de experimentos se ve que, en el caso de marcas 2D, Figura 9, tanto la trayectoria del robot como la pared han sido estimadas correctamente, aunque las elipses de incertidumbre asociadas a las marcas presentan un leve traslape, lo cual no es muy adecuado para la reconstrucción 3D de la pared pues no permite distinguir correctamente las marcas entre ellas; mientras que, en el caso de marcas 2D orientadas ilustrado en la Figura 10, esta situación se mejoró sustancialmente.



Figura 11. Segundo experimento: Caminata dentro de un recinto cuadrado cuyas paredes tienen marcas



Figura 12. Mapa y trayectoria del robot en el caso de marcas 2D del segundo experimento.



Figura 13. Mapa y trayectoria del robot en el caso de marcas 2D orientadas del segundo experimento.

Caminata en Bucle Cerrado dentro de un claustro cuadrado

En el segundo grupo de experimentos se obtuvieron resultados similares, sólo que, además, se tiene que las estimaciones se van haciendo menos buenas a lo largo del camino recorrido por tratarse de un recorrido más largo. Esto se puede apreciar en la manera en que crecen las elipses de incertidumbre construidas con los valores propios de la matriz de covariancia calculada en cada punto de medición. Sin embargo, cuando el robot regresa a su posición original y ve de nuevo las primeras marcas, la incertidumbre tanto del mapa como de su localización en dicha zona se reduce enormemente. Así, cabe esperar que si se hace un nuevo recorrido, las incertidumbres se reduzcan en toda la trayectoria.

Del mismo modo que en el primer grupo de experimentos, aquí se aprecia que, al haber considerado la orientación de las marcas en el caso número 2, Figura 13, los resultados mejoraron sustancialmente reduciéndose bastante las elipses de incertidumbre obtenidas en el algoritmo de SLAM visual EKF.

5. CONCLUSIONES

En este trabajo se reporta una aplicación del algoritmo de SLAM FKE estándar a para generar el mapa de un entorno desconocido pero estructurado y, simultáneamente, calcular la trayectoria de un robot que cuenta con varias innovaciones.

Primeramente, las mediciones utilizadas son exclusivamente visuales y utilizan el modelo de proyección en perspectiva para calcular la posición relativa de las marcas con respecto a la cámara mediante una medición de su distorsión. Este método de análisis de imágenes para la medición de la posición relativa permite, además, la obtención de la orientación de las marcas observadas. A nuestro conocimiento, es la primera vez que se utiliza esta información en un sistema de SLAM.

En segundo lugar, el esquema de SLAM visual FKE desarrollado utiliza un hardware muy compacto y de bajo costo, lo cual permite aplicarlo a un humanoide comercial de pequeño formato, el Robonova-I. No es muy común encontrar sistemas de SLAM para robots humanoides del tipo utilizado en nuestros experimentos.

Finalmente, los resultados obtenidos con marcas 2D y marcas 2D orientadas y sin odometría nos hace pensar que en nuestros trabajos actuales en donde comenzamos a utilizar una odometría parcial (orientación del robot obtenida con una brújula electrónica) serán aún mejores. Además, como pensamos instalar en el robot una unidad de mediciones inerciales, estaremos en medida de generar una odometría más completa.

6. **REFERENCIAS**

- José Neira, Andrew J. Davison, and John J. Leonard, "Guest editorial special issue on Visual SLAM", IEEE Transactions on Robotics, Vol. 24, No. 5, 2008 pp 929-931.
- [2] R. Smith, M. Self, P. Cheeseman, "Estimating uncertain spatial relationships in Robotics", En: J.F. Lemmer y L.N. Kanal (Eds.), Uncertainty in Artificial Intelligence, Vol. 2, 1988, pp. 435-461. Elsevier Science Pub.
- [3] H. F. Durrant-Whyte. "Uncertain Geometry in Robotics", IEEE International Conference on Robotics and Automation, Vol. 4, 1987 pp. 851-856.
- [4] Diego R. Losada González. "Slam Geométrico en tiempo real para robots móviles en interiores basado en EKF". Tesis doctoral, Escuela Técnica Superior de Ingenieros Industriales, España, 2004.

- [5] T. Bailey, H. F. Durrant-Whyte, "Simultaneous Localisation and Mapping (SLAM): Part I, The essential algorithms". IEEE Transactions on Robotics and Automation, Vol. 13, 2006, pp. 1-9
- [6] L. M. Paz, P. Piniés, J. D. Tardós y J. Neira, "Large Scale 6DOF Slam with Stereo-in-Hand", IEEE Transactions on Robotics, (to appear), 2008.
- [7] Solà, Joan: "Multi_camera VSLAM: from former information losses to self-calibration" SRI International, workshop, 2008,
- [8] Davison, A.J, "Real-Time Simultaneous Localisation and Mapping with a Single Camera" IEEE International Conference on Computer Vision, 2003.
- [9] Andrew P. Gee, Denis Chekhlov, Andrew Calway, and Walterio Mayol-Cuevas, "Discovering higher level structure in visual slam" IEEE Transactions on Robotics, Vol. 24, No.5, Octubre 2008, pp:980–990.
- [10] Javier Civera, Andrew J. Davison, and J. M. Martínez Montiel, "Inverse depth parametrization for monocular slam", **IEEE Transactions on Robotics**, Vol. 24, No. 5, , Octubre 2008, pp: 932–945.
- [11] Eric Hernández Castillo. "Slam visual para un robot humanoide". Tesis de Maestria, Centro de Investigación y de Estudios Avanzados del IPN, Noviembre 2008.
- [12] J. M. Ibarra Zannatha, R. Cisneros, E. Hernández, J. E. Lavín, M. A. Trujano, "Proposición de una técnica de análisis de imágenes para generar los vectores clásicos usados en SLAM", CIINDET 2008, IEEE, 6º Congreso Internacional en Innovación y Desarrollo Tecnológico. Cuernavaca, México. ISBN: 978-968-9152-01-9, Octubre, 2008
- [13] J. M. Ibarra Zannatha, E. Hernández, J. Neira, R. Cisneros, y J. E. Lavín, "SLAM visual tipo EKF aplicado a un robot humanoide", Congreso Mexicano de Robótica, (*ComRob*), Septiembre 2008.
- [14] J. M. Ibarra Zannatha, Eduardo de Jesús Iturbe Córdova. "Generación de Mapas Virtuales Interactivos para Robots Móviles usando la Reconstrucción 3D de Kanatani" Quinto Congreso Mexicano de Robótica CoMRob 2003 de la AMRob. San Luis Potosí, agosto, 2003, pp. 182-188.
- [15] Eduardo Jesús Iturbe Córdova, J. M. Ibarra Zannatha. "Reconstrucción 3D Monocular basada en Cosenos Directores". Memorias del Sexto Congreso Mexicano de Robótica de la AMRob 2004. Torreón, Coahuila, 2004.
- [16] Eduardo Jesús Iturbe Córdova "Generación de mapas para la navegación de robots móviles" Tesis de Maestría en Ciencias especialidad Control Automático. Cinvestav. México, DF, septiembre, 2004.
- [17] Jorge Enrique Lavín Delgado "Reconstrucción 3D monocular y estereoscópica". Tesis de Maestría en Ciencias en Control Automático.Cinvestav México, DF, febrero, 2009

- [18] J. M. Ibarra Zannatha, Visión Artificial para Robots, Notas de Curso, México, DF, Cinvestav, 2005
- [19] Carlos M. Soria, Ricardo Carelli, Rafael Kelly, Juan M. Ibarra Zannatha, "Coordinated Control of Mobile Robots Based on Artificial Vision", International Journal of Computers, Communications & Control, Vol. 1, No. 2, 2006, pp. 85-94
- [20] Ibarra Zannatha J. M, Lavín Delgado J. E., Hernández Castillo E., Cisneros Limón R., "Reconstrucción 3D Monocular para el Sistema de SLAM Visual de un Robot Humanoide", 10° Congreso Mexicano de Robótica, México D.F. Universidad Anáhuac México Sur, Septiembre, 2008.
- [21] J. M. Ibarra Zannatha, L. D. García Cobos, A. D. Gómez Sánchez, R. Cisneros Limón, L. F. Lupián Sánchez, R. A. Gil Rivera, "Diseño, construcción, modelado, control y programación de una cabeza con 2 gdl para un humanoide", Memorias del Congreso Mexicano de Robótica de la AMRob 2009, Celaya, Gto., Octubre, 2009
- [22] J. M. Ibarra Zannatha, R. Cisneros Limón, "Modelado y Simulación de un Humanoide" CISCI, Orlando, Florida, Julio, 2008
- [23] J. M. Ibarra Zannatha, R. Cisneros Limón, L. García Cobos, J. E. Lavín Delgado, I. Yáñez Díaz. "Evasión de obstáculos basada en visión artificial y en sensores infrarrojos para un humanoide", 6° Congreso Internacional en Innovación y Desarrollo Tecnológico, México, 2008.
- [24] Castellanos J.A., J. Neira, J.D. Tardós. "Map Building and SLAM Algorithms", in Autonomous Mobile Robots: Sensing, Control, Decision Making and Applications. Shuzhi Sam Ge, Frank L. Lewis (eds) T&F, May 2006.
- [25] Hernández Castillo E., "SLAM para un humanoide en interiores basado en EKF". Reporte RR-08-07, Depto. de Informática e Ingeniería de Sistemas de la Universidad de Zaragoza, España. Julio 29 de 2008.