

Aplicación de un EIF-SLAM en entornos agrícolas basado en detección de troncos de árboles

Fernando A. Auat Cheein¹, Guillermo Steiner², Gonzalo Perez Paina², Ricardo Carelli¹

Resumen

En este trabajo se presenta la implementación de un algoritmo de SLAM (por sus siglas en inglés de *Simultaneous Localization and Mapping*) en un Filtro Extendido de Información para ambientes agrícolas (plantación de olivos). El algoritmo de SLAM se encuentra implementado en un robot *unmanned* tipo car-like no-holonómico. El mapeo del ambiente se basa en la implementación de máquinas de soporte vectorial (SVM, por sus siglas en inglés de *Support Vector Machine*) para la extracción de troncos asociados a los olivos del entorno. Este trabajo incluye resultados experimentales de implementación en entornos agrícolas reales.

Fecha de recepción del original: 21/10/2010 | Fecha de evaluación del original: 26/10/2010

1. Instituto de Automática, Universidad Nacional de San Juan. Av. San Martin Oeste 1109, San Juan, Argentina | fauat@inaut.unsj.edu.ar, rcarelli@inaut.unsj.edu.ar

2. Centro de Investigación en Informática para la Ingeniera. Universidad Tecnológica Nacional - Regional Córdoba. Maestro M. Lopez esq. Cruz Roja Argentina, Ciudad Universitaria - Córdoba, Argentina | gperez@scdt.frc.utn.edu.ar, gsteiner@scdt.frc.utn.edu.ar

1. Introducción

En este trabajo se propone una arquitectura de localización y reconstrucción simultánea de entornos, SLAM (por sus siglas en inglés de *Simultaneous Localization and Mapping*), para la creación de mapas de entornos agrícolas útiles a la agricultura de precisión.

El algoritmo de SLAM es un algoritmo de estimación recursiva que minimiza simultáneamente los errores que intervienen en el proceso de localización de un vehículo y del mapeo del entorno por el cual el vehículo navega. El algoritmo de SLAM tiene sus orígenes en [1, 2], donde por primera vez se comenzó a incluir la información del ambiente en los movimientos de un robot móvil mientras que a su vez el robot interactuaba con el entorno. Los trabajos de [3, 4] ofrecen una introducción más concisa sobre la evolución del algoritmo de SLAM.

Actualmente, el problema de SLAM se puede resolver por medio de diferentes filtros disponibles en la literatura científica. El método más extendido de resolución [3] es mediante la implementación de un filtro Extendido de Kalman (EKF, por sus siglas en inglés de *Extended Kalman Filter*) que minimiza el error cuadrático del proceso. Al utilizar el EKF-SLAM, el vector de estados del sistema pasa a estar compuesto por la pose del vehículo –su posición dentro del ambiente y su orientación absoluta– y por los parámetros que definen los elementos del entorno. En [5], por ejemplo, se resuelve el problema de SLAM con un EKF y un entorno basado exclusivamente en líneas. Esto es, adquiere solamente la información geométrica del ambiente que pueda modelarse mediante líneas (tales como paredes, puertas, etc.). El trabajo de [6] implementa un EKF-SLAM basado en la extracción únicamente de características puntuales (árboles, esquinas geométricas, etc.). Así también el mapa puede estar compuesto por cualquier elemento que pueda ser matemáticamente modelable. Como alternativas al EKF se pueden encontrar en la literatura diversos filtros derivados que tienden a compensar las desventajas del EKF, tal como el UKF (*Unscented Kalman Filter*) [7] usado para compensar las no-linealidades o el EIF (*Extended Information Filter*) [8], que mejora los tiempos de procesamiento de la etapa de corrección del algoritmo de SLAM disminuyendo la misma en un orden de magnitud.

Otra solución al algoritmo de SLAM es el filtro de partículas (PF, por sus siglas en inglés de *Particle Filter*) [7], que a diferencia del EKF, no se restringe a procesos puramente Gaussianos y tiene un mejor manejo de las no-linealidades involucradas, aunque su implementación en tiempo real es aún restringida.

Cualquiera sea el tipo de algoritmo implementado para solucionar el problema de SLAM, el mismo debe asegurar que la estimación sea consistente (el mapa del entorno debe tener un sentido acorde con el ambiente real y la información que el mismo provea, debe ser de utilidad al robot) y convergente (los errores deben ser minimizados, asegurando así la mejor estima del proceso) [9].

La extracción de las características del mapa a ser usadas en el algoritmo de SLAM, depende intrínsecamente del sensor a ser utilizado. Así, un sensor de visión artificial no adquiere la misma información que un ultrasonido, que un sensor de rango láser o que un GPS, o un giróscopo, etc.

La utilización de visión artificial permite extraer información relevante del entorno cuya característica resulta diferente de los demás sensores [10]. Aplicando técnicas de aprendizaje de máquinas se puede obtener información robusta para ser aplicado al algoritmo de SLAM.

Una Máquina de Soporte Vectorial (SVM, por su siglas en inglés de *Support Vector Machine*) [11] permite clasificar los objetos de la escena percibida pertenecientes a cierta clase de interés, como troncos de árboles. La SVM es un método de aprendizaje supervisado, cuya etapa de entrenamiento permite, en base a muestras positivas y negativas de objetos de interés etiquetados, aprender un modelo general de dicho objeto. De esta forma se realiza, luego, la clasificación de un objeto nuevo. Esto se hace mediante la partición de un espacio de características de alta dimensionalidad donde se representan los objetos de la imagen. Esta representación de los objetos en el espacio de las características se conoce como descriptor; siendo el Histograma de Gradientes Orientados (HOG, por sus siglas en inglés de *Histogram of Oriented Gradients*) uno de los más utilizados actualmente, el cual permite representar una porción de la imagen [12].

En este trabajo se presenta la implementación de un algoritmo de SLAM basado en un EIF. El algoritmo de SLAM implementado está orientado a entornos agrícolas (específicamente, una plantación de olivos) con finalidades de agricultura de precisión.

Las características extraídas del ambiente corresponden a los troncos de los olivos. Los mismos, son tratados como características puntuales del entorno. La extracción de los troncos se basa en un sistema doble de visión artificial y sensor de rango láser. El sistema de visión artificial tiene implementada una máquina de soporte vectorial, SVM (*Support Vector Machine*) para la detección de troncos dentro de la imagen. A partir de esta detección, se obtiene el ángulo de los troncos dentro de dicha imagen que, al coincidir con la orientación del robot, se obtiene el ángulo del tronco con respecto al robot. El sensor de rango láser devuelve la distancia a la que se encuentra el tronco detectado luego de haber buscado en un entorno del ángulo asociado con dicha característica. La información de ángulo y rango asociada a un tronco, es luego enviada al algoritmo de SLAM.

Acompañan a este trabajo resultados de experimentación dentro de una plantación de olivares del Instituto Nacional de Tecnología Agropecuaria - San Juan (INTA-San Juan).

Este trabajo se organiza de la siguiente manera. La sección 2 muestra la arquitectura general del sistema propuesto en este trabajo; la sección 3 explica

en detalle el algoritmo de EIF-SLAM; la sección 4 explica los procedimientos basados en visión artificial y en láser para la detección de los troncos del ambiente; la sección 5 presenta los resultados de implementación de la propuesta. Por último, se presentan las conclusiones de este trabajo.

2. Arquitectura general del sistema

La arquitectura general del sistema se encuentra representada en la Fig. 1. El funcionamiento de esta arquitectura se puede resumir de la siguiente manera:

- El ambiente agrícola es el entorno de navegación del robot. Es una plantación de olivos perteneciente al Instituto Nacional de Tecnología Agropecuaria, San Juan.
- El mecanismo de visión y el láser hacen referencia a los dispositivos usados para la extracción de troncos del ambiente. El sistema de visión monocular provee de información al algoritmo de SVM el cual extrae la posición de los troncos de los árboles dentro de cada imagen. Dicha posición tiene asociada una orientación. El láser, provee la información de rango asociado a cada tronco. La información de rango y orientación de un tronco dado se consideran como parámetros asociados a la característica *tronco*. Dichos parámetros son pasados luego al algoritmo de SLAM.
- El algoritmo de SLAM estima, recursivamente, la pose del vehículo y el mapa del ambiente, minimizando los errores asociados a ambos procesos, basándose en la información adquirida del ambiente (troncos de olivos).

A continuación, se explicará en detalle cada uno de los bloques mostrados en la Fig. 1.

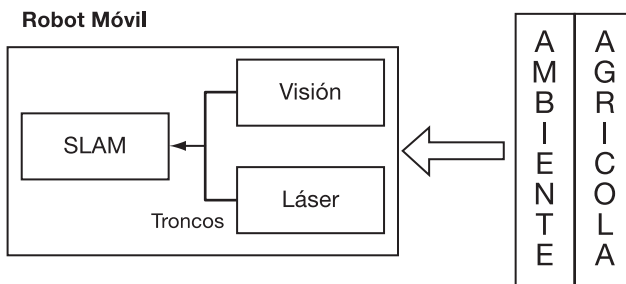


Figura 1 | Esquema de la arquitectura general del sistema propuesto en este trabajo.

3. Algoritmo de SLAM

El algoritmo de SLAM usado en este trabajo se basa en la implementación de un Filtro Extendido de Información (EIF, *Extended Information Filter*) para la estimación recursiva y conjunta de la pose del vehículo dentro del ambiente y del mapa representativo de dicho entorno. El algoritmo de EIF se describe en la Ec. (1).

$$\begin{aligned}
 \mu_{t-1} &= \Omega_{t-1}^{-1} \bar{\xi}_{t-1} \\
 \bar{\Omega}_t &= (G_t \Omega_{t-1}^{-1} G_t^T + R_t)^{-1} \\
 \bar{\xi}_t &= \bar{\Omega}_t g(u_t, \mu_{t-1}) \\
 \Omega_t &= \bar{\Omega}_t + H_t^T Q_t^{-1} H_t \\
 \xi_t &= H_t^T Q_t^{-1} [z_t - h(\bar{\mu}_t) + H_t \bar{\mu}_t]
 \end{aligned} \tag{1}$$

En la ecuación Ec. (1), μ es el vector de estados del sistema, el cual contiene tanto la información relativa a la pose del vehículo dentro del entorno como así también de todas las características encontradas en el ambiente; ξ y Ω –la matriz de información– son los parámetros del EIF, donde ξ tiene las dimensiones de μ y Ω tiene las dimensiones de la matriz de covarianza asociada al vector de estados del sistema; G es la matriz Jacobiana del modelo del sistema, R es su matriz de covarianza y u es el vector de control de entrada; H es la matriz Jacobiana del modelo de las características y Q es la matriz de covarianza asociada con ella; la característica actual es z . Finalmente, $\bar{\xi}$ y $\bar{\Omega}$ son los parámetros predichos del filtro (más información acerca del EIF se puede encontrar en [7]).

La etapa de corrección del algoritmo de EIF secuencial se puede expresar tal como se muestra en la Fig. 2. Esta etapa de corrección es la que atañe a este trabajo pues es aquí donde se implementan las estrategias de optimización de los algoritmos de SLAM.

- 1: Sea N_t el conjunto de características observadas en el instante t
- 2: Sea $M_t \subseteq N_t$ el conjunto de características con asociación correcta en tiempo t
- 3: for $j = 1$ to $\#M_t$ do
- 4: find $z_j^{opt} : \text{arg max}(|\Omega_t|)$
- 5: $\Omega_t = \bar{\Omega}_t + H_t^T Q_t^{-1} H_t$
- 6: $\xi_t = \bar{\xi}_t + H_t^T Q_t^{-1} [z_j - h(\bar{\mu}_t) + H_t \bar{\mu}_t]$
- 7: $\bar{\Omega}_t := \Omega_t; \bar{\xi}_t = \xi_t$
- 8: end for

Figura 2 | Algoritmo del EIF secuencial.

3.1. EIF basado en el Filtro Extendido de Kalman (EKF)

La parametrización del EIF puede ser derivada directamente a partir del EKF [7]. Así,

$$\begin{cases} P_t = \Omega_t^{-1} \\ \xi = \Omega_t \mu_t \end{cases} \quad (2)$$

En la Ec. (2), P es la matriz de covarianza asociada al vector de estados del sistema en el EKF (ξ). De acuerdo con [7], se puede probar que la etapa de corrección del EIF es lineal mientras que la etapa de corrección en el EKF es cuadrática –por esta razón el EIF es más rápido que el EKF–. Esta situación se puede ver a partir de las ecuaciones (1) y (3) –esta última representa el Filtro Extendido de Kalman–. Por otro lado, la etapa de predicción del EIF requiere la inversión de una matriz de orden $n \times n$ –donde n es el tamaño del vector de estados del sistema– mientras que en el EKF la etapa de predicción es lineal.

$$\begin{aligned} \hat{\xi}_t^- &= f(\hat{\xi}_t, u_t) \\ P_t^- &= A_t P_{t-1} A_t^T + W_t Q_{t-1} W_t^T \\ K_t &= P_t^- H_t^T (H_t P_t^- H_t^T + R_t)^{-1} \\ \hat{\xi}_t &= \hat{\xi}_t^- + K_t (z_t - h(\hat{\xi}_t^-)) \\ P_t &= (I - K_t H_t) P_t^- \end{aligned} \quad (3)$$

En (3), f es el modelo del sistema, $\hat{\xi}$ es el vector de estados del sistema, u es el vector de entradas de control, P es la matriz de covarianza asociada a $\hat{\xi}$. A es la matriz Jacobiana de f con respecto a $\hat{\xi}$, mientras que Q representa el ruido Gaussiano del proceso y W su matriz Jacobiana; K es la ganancia de Kalman, H es la matriz Jacobiana asociada al modelo de la característica observada y R su covarianza asociada. z es la característica observada por el sistema mientras que h es la predicción de la misma. Más información sobre el EKF y su implementación se puede encontrar en [13].

4. Extracción de características

Las características extraídas del ambiente agrícola de este trabajo corresponden a los troncos de los olivos. Estos troncos están definidos por dos parámetros: rango y ángulo [17]. El rango es la distancia a la que se encuentra el tronco con respecto a la posición del vehículo; mientras que el ángulo es la orientación de dicho tronco con respecto a la pose del vehículo. La Fig. 3 muestra los parámetros asociados a un tronco cualquiera. Además, la Ec. 12 muestra el modelado analítico de los troncos como características puntuales del ambiente –ver [17]–.

$$z_{tronco} = h_i[x_{robot}, y_{robot}, \psi, p_i, \varpi] = \begin{bmatrix} z_R \\ z_\beta \end{bmatrix} = \begin{bmatrix} \sqrt{(x_{robot} - x_{tronco})^2 + (y_{robot} - y_{tronco})^2} \\ \arctan\left(\frac{y_{robot} - y_{tronco}}{x_{tronco} - x_{tronco}}\right) + \psi \end{bmatrix} + \begin{bmatrix} \varpi_R \\ \varpi_\beta \end{bmatrix} \quad (12)$$

En la Ec. 12, z_{tronco} es el vector de parámetros del tronco de olivo, compuesto por la información de rango (z_R) y la información de ángulo asociada a la detección (z_β). Las variables x_{robot} , y_{robot} y ψ representan la posición y la orientación del robot dentro del ambiente, mientras que p_i representa las coordenadas del tronco respecto a un sistema de referencia local al vehículo. El vector ϖ corresponde a ruido Gaussiano asociado con la medición. La Fig. 3 muestra una representación gráfica de las variables de la Ec. 12. Como se puede ver, el tronco es considerado como una característica puntual del ambiente [17].

La extracción de características se divide en dos fases: extracción de orientación de los troncos y extracción de rango. La extracción de la orientación se efectúa mediante el procesamiento de las imágenes adquiridas por el sistema de visión monocular montado sobre el robot. Se utiliza una máquina de soporte vectorial para detectar los troncos de los árboles, obteniéndose la posición del tronco dentro de la imagen a partir de la cual, mediante geometría proyectiva, se calcula el ángulo del tronco respecto al eje principal de la cámara. Debido a que de esta forma se obtienen solo mediciones de ángulo, se utilizó también un sensor láser para estimar el rango del tronco detectado.

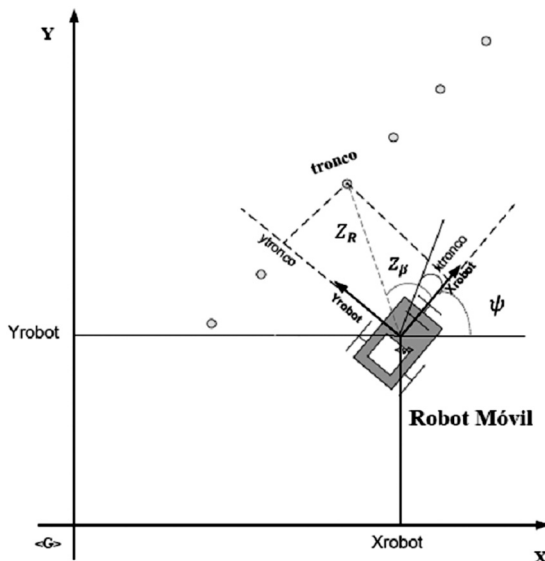


Figura 3 | Representación gráfica de los troncos del ambiente agrícola.

El uso de dos sensores diferentes para las mediciones de rango y ángulo de los troncos de olivos se basa en que:

- Un único sensor de visión monocular no devuelve mediciones de profundidad.
- El sensor láser puede adquirir mediciones de rango y orientación, pero no distingue entre troncos, personas u obstáculos afines a la morfología del tronco del árbol.
- La fusión de ambos sensores robustece la estima inicial de los parámetros del tronco.

A continuación, se detallará cada uno de los métodos de extracción de características usados en este trabajo.

4.1. Detección de troncos mediante sensores de visión monocular

Para la detección de los troncos de árboles se utilizan técnicas de *machine learning* aplicadas al reconocimiento del entorno. El reconocimiento de entorno se focaliza en establecer un mapa de estructuras, objetos rígidos, personas, etc., que rodea un dispositivo. Este reconocimiento puede realizarse por medio de visión artificial, ultrasonido, láser, etc. En particular, el presente trabajo pretende reconocer troncos de árboles por medio de visión artificial.

El proceso de reconocimiento podemos separarlo en dos etapas:

- extracción de la característica de una imagen
- clasificación mediante las características extraídas

Para el caso de la representación del espacio de características se utiliza los descriptores HOG (*Histogram of Oriented Gradient*) [18]. Este método, utilizado generalmente en detección de personas, se basa en un método simplificado de descriptores SIFT [19] pero aplicado en forma densa sobre la superficie a caracterizar.

La técnica de aprendizaje utilizada se conoce como SVM (*Support Vector Machine*), y se utiliza para clasificar las zonas de la imagen que corresponden al objeto de interés.

4.1.1. Descriptores HOG

La extracción de características, consiste en caracterizar o describir a la imagen por medio de valores extraídos de ella, para esto, el método HOG realiza los siguientes pasos:

- Calcula las derivadas espaciales de la imagen en dirección x e y , obteniendo luego la dirección y magnitud del gradiente en cada punto de la imagen

- Se subdivide la imagen en bloques descriptores. Estos bloques de un tamaño prefijado están dispuestos de forma tal que cubran toda la imagen a describir con una superposición entre ellos
- Se subdividen cada uno de estos bloques en celdas
- Por cada celda se calcula un vector que posee la suma de las contribuciones de cada punto en una cantidad de direcciones dada o bins
- Los valores resultantes en este vector varían en un amplio rango, debido a variaciones en el contraste e iluminación, es necesario entonces una normalización, en este caso la normalización L_2 -norm dio mejores resultados

$$L_2(v) = \frac{v}{\sqrt{\|v\|_2^2}}$$

- Los vectores de bins de las celdas que forman un bloque son concatenados entre sí, resultando en un vector descriptor del bloque

Finalmente, se concatenan los vectores de cada bloque en un único vector descriptor de la imagen.

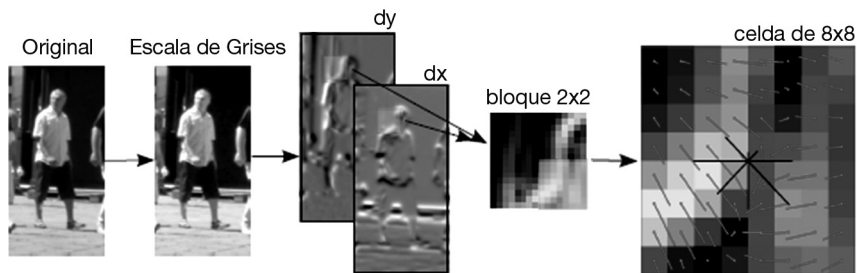


Figura 4 | Detalle de celda de 8 x 8 con su vector descriptor de 8 bins en un bloque de 2 x 2 aplicado al reconocimiento de una persona. Como se puede ver, la técnica presentada en este trabajo no se restringe a un tipo especial de patrón.

4.1.2. Clasificación con SVM

Las máquinas de soporte vectorial (SVM, por sus siglas en inglés de *support vector machines*) [20] son un tipo de máquina núcleo (o máquina *kernel*) [21] utilizadas como método de aprendizaje supervisado para tareas de clasificación.

Una tarea de clasificación generalmente consiste de una etapa de entrenamiento y una de evaluación con alguna instancia del conjunto de datos. Cada instancia de los datos de entrenamiento contiene un valor objetivo o etiqueta y un conjunto de atributos o características. El objetivo de la SVM es generar un modelo que pueda

predecir el valor de etiqueta (o clase) en el conjunto de evaluación dado solamente de las características.

De manera formal, dados un conjunto de n puntos de los datos de entrenamiento D de la forma

$$D = \left\{ (x_i, y_i) \mid x_i \in \mathfrak{R}^p, y_i \in \{-1, 1\} \right\}_{i=1}^n$$

donde y_i indica la clase a la que pertenece el punto x_i , y los x_i son vectores reales de dimensión p . La SVM encuentra el hiperplano que maximiza el margen que divide los puntos cuyos $y_i = 1$ de aquellos cuyos $y_i = -1$.

Dado un conjunto separable existe al menos un hiperplano $\pi : \mathbf{w} \cdot \mathbf{x} + b = 0$ que separa los vectores $x_i, i = 1, \dots, n$. La SVM, por lo tanto, resuelve el siguiente problema de optimización cuadrática

$$\begin{aligned} \min_{w,b} \quad & \frac{1}{2} \|w\|^2 \\ \text{s.a.} \quad & y_i (w \cdot x_i + b) \geq 1 \end{aligned} \quad (13)$$

Se puede realizar una clasificación no lineal reemplazando el producto punto por una función kernel, esto permite que el algoritmo ajuste el hiperplano al máximo margen en el espacio de características transformado. El kernel se relaciona con la transformación $\phi(x_i)$ por medio de $K(x_i, x_j) \equiv \phi(x_i) \phi(x_j)$.

4.1.3. Implementación

Para la descripción mediante HOG se utilizan una serie de parámetros a configurar para encontrar la mejor manera de describir una imagen. En este trabajo se utilizaron los siguientes:

- tamaño del bloque (2x2 celdas)
- superposición entre bloques (8 bits)
- tamaño de la celda (utilizada 8x8 pxel)
- cantidad de bins (9 bins, además se toma la dirección y no el sentido del gradiente, disponiendo los 9 bins en 180 grados)
- tamaño de la imagen a describir (32x72 píxel)

Luego, con este vector descriptor, se pasa a la etapa de clasificación, para esto se utiliza una SVM con kernel lineal, la cual fue previamente entrenada con un conjunto de imágenes positivas (para este caso imágenes con árboles) y un conjunto de imágenes negativas (sin árboles). Las imágenes utilizadas en el entrenamiento de la SVM y luego las clasificadas, deben tener el mismo tamaño, de esta forma el vector generado en todo los casos posee la misma dimensión. Para otro tamaño de imagen, se deberá generar una grilla que particione la misma

en bloques del tamaño del clasificador para luego clasificar cada uno de estos bloques de manera aislada.

Otro problema surge con respecto a las escalas. La búsqueda en una grilla de bloques de un tamaño determinado, solo podrá detectar árboles de las dimensiones con que fue entrenado. Se crea entonces una representación de la imagen en multi-escala, esto es, generar por medio de sucesivos filtros de suavizado y posterior *downsampling* una pirámide de imágenes, donde cada nivel de esta pirámide, representa una escala de la imagen original. Aplicando el criterio de búsqueda a partir de una grilla de imágenes en cada nivel, resultará en una serie de detecciones de árboles en diferentes escalas.

4.1.4. Medición de ángulo

Para realizar la medición de ángulo se utiliza el modelo *pin-hole* de la cámara perspectiva. La proyección de la escena en la imagen se describe mediante una matriz homogénea de 3×4 llamada *matriz de proyección de la cámara*

$$P = KR[I, -C] \quad (14)$$

donde K es la matriz de parámetros intrínsecos, y R (matriz una rotación) y C (centro óptico) son los parámetros extrínsecos. La matriz de proyección tiene 11 grados de libertad, siendo los mismos que una matriz de 3×4 definida hasta un factor de escala arbitrario. Los parámetros de la cámara se pueden obtener mediante el proceso de calibración [22].

La matriz de parámetros intrínsecos tiene la forma:

$$K = \begin{bmatrix} fk_u & -fk_u \cot \beta & u_0 \\ 0 & fk_v / \sin \beta & v_0 \\ 0 & 0 & 1 \end{bmatrix} \quad (15)$$

donde los parámetros más importantes la distancia focal f y las coordenadas del punto principal (u_0, v_0) .

La Fig. 5 representa como se proyectan segmentos de líneas verticales en la cámara. Los puntos 3D del entorno X_i se proyectan a los puntos imágenes m_i sobre el plano imagen Π , este se encuentra desplazado del sistema de coordenadas de la cámara una distancia f , de forma que el plano imagen sea perpendicular al eje o rayo principal. Los puntos 3D se proyectan a la imagen como

$$m_i \cong PX_i$$

donde \cong significa que son iguales hasta un factor de escala.

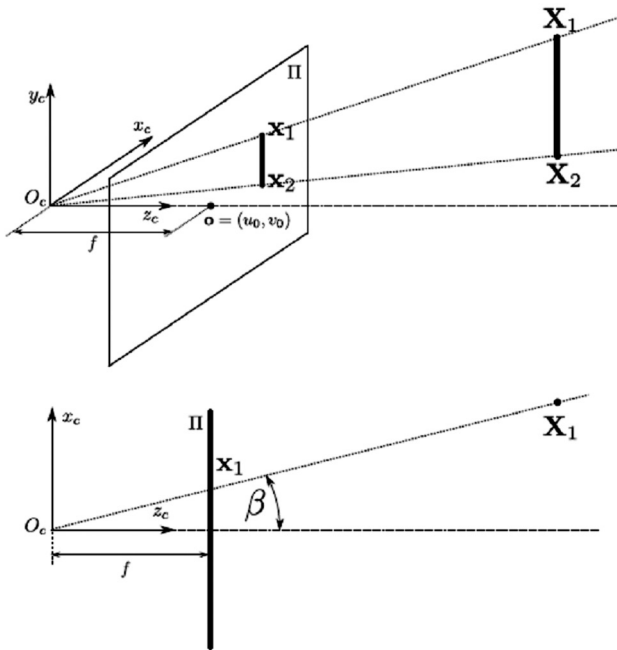


Figura 5 | Esquema para la medición de ángulo

Teniendo la descripción de la línea del objeto de interés de la escena en la imagen, y conociendo la matriz de parámetros intrínsecos de la cámara se puede obtener los puntos imagen respecto al sistema de coordenadas de la cámara como $x_i = K^{-1}m_i$, donde $x = (x, y, f)$; con esto se puede calcular el ángulo de la línea vertical como $\beta = \arctan(x/f)$. Este ángulo corresponde con la rotación sobre el eje y_c de la cámara, sobre el plano $x_c z_c$.

4.2. Detección de troncos mediante láser

La detección de troncos del ambiente mediante un sensor láser aporta la información de rango asociada a la detección de los troncos. El sensor láser usado en este trabajo corresponde a un láser SICK que adquiere 181 mediciones entre 0 y 180 grados, en un rango máximo de 30 metros. El algoritmo de extracción de troncos fue previamente presentado por [15, 17] y consiste en considerar al tronco como un objeto puntual, tal como se presentó en la Ec. 12. El algoritmo consiste en extraer la distancia a la que se encuentra el tronco con respecto al vehículo a partir del histograma asociado a la medición, tal como se muestra en la Fig. 6.

Información adicional respecto a la extracción de troncos mediante sensores de rango láser, se puede encontrar en [15, 17].

Para integrar este algoritmo de detección de troncos con el algoritmo basado en visión, se realiza el siguiente procedimiento:

- Mediante el algoritmo de SVM se determinan los troncos detectados dentro de la imagen y sus ángulos respectivos.
- Como la cámara está fija en el vehículo, el ángulo asociado a cada tronco se asocia directamente al marco de referencia local al robot móvil (por ejemplo, si un tronco se determina a 30 grados en la imagen, corresponde también a 30 grados respecto a la pose del robot, ver sección 5).
- Una vez que se tiene la información de ángulo, se constata el rango asociado al tronco mediante las mediciones del láser. Así, si se encontró un tronco a 30 grados, y debido a que el láser adquiere mediciones entre 0 y 180 grados, se busca un tronco en el entorno de 30 grados. Si es encontrado, entonces dicho tronco tiene asociado un rango y su parametrización se completa. Caso contrario, el tronco es desechado.
- El proceso se repite para todos los troncos detectados en la imagen.

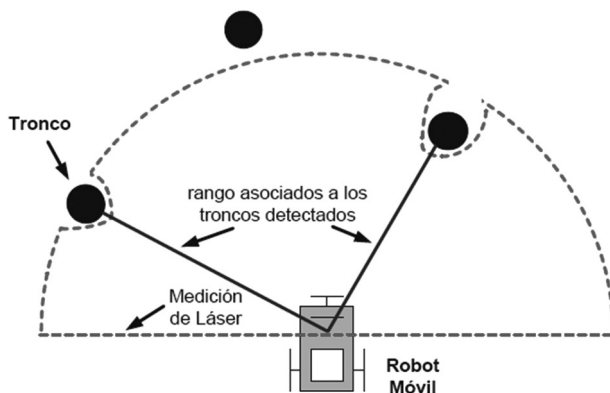


Figura 6 | Diagrama representativo de la detección de troncos realizada por el sensor de rango láser usado en este trabajo.

5. Resultados de implementación

En esta sección se mostrarán los resultados de implementar el algoritmo de SLAM propuesto junto con la detección de troncos basada en visión artificial y láser.

La Fig. 7 muestra una fotografía del vehículo usado en este trabajo. Este vehículo es del tipo *unmanned* con una cinemática tipo *car-like* [23]. El mismo cuenta con

dos sensores láser **SICK**, ambos en su parte frontal pero uno apuntando en la dirección de movimiento y otro de forma periférica. Cuenta además con cámaras estéreo tipo *Bumblebee 2* –aunque se utiliza una sola cámara, cuya apertura de campo es apropiada para la propuesta de este trabajo– y un GPS *Novatell* diferencial. El vehículo es totalmente autónomo y fue construido en el *Instituto de Automática de la Facultad de Ingeniería de la Universidad Nacional de San Juan*. Tanto el sensor de rango láser como la cámara se ajustan para estar alineados sobre el mismo eje vertical así el láser toma información de rango en el ángulo indicado por la información visual (sin necesitar transformaciones extras).



Figura 7: Fotografía del vehículo autónomo usado en este trabajo.

5.1. Extracción de troncos

La Fig. 8 muestra diferentes casos de la detección de troncos de la plantación de olivos mientras el vehículo navega entre los surcos de la misma. Los casos mostrados en la Fig. 8 corresponden a distintas situaciones y posiciones del robot dentro del ambiente, como así también de la inclinación del mismo dadas las condiciones del terreno. Como se puede ver, se detectan todos los troncos próximos y algunos de los surcos adyacentes. La múltiple detección de un mismo tronco obedece a que el mismo es detectado bajo diferentes valores de escala, tal como se explicó en la sección 4. Para calcular la orientación asociada al tronco, se tiene en cuenta el promedio de los centros de masas asociados a la detección de un mismo tronco.

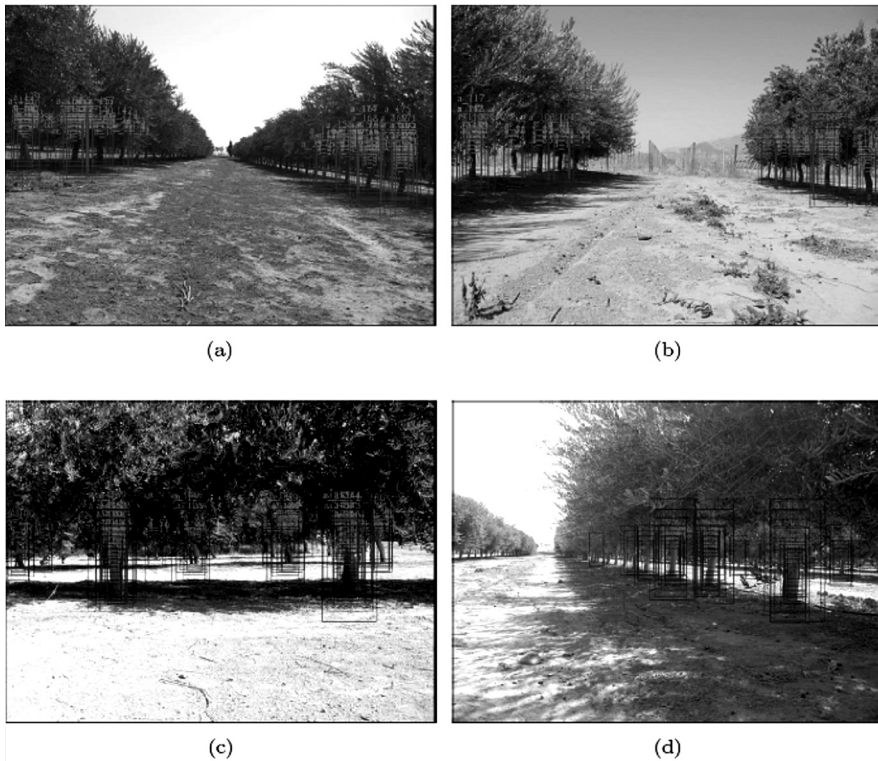


Figura 8: Ejemplos de detecciones de troncos mediante la SVM propuesta en este trabajo. Los rectángulos en azul encierran la porción de la imagen en que fue detectado el tronco y las diferentes escalas en las que sucedió dicha detección. En rojo, son las etiquetas asignadas a cada tronco.

5.2. EIF-SLAM

En la Fig. 9(a) se muestra el recorrido realizado por el vehículo dentro de la plantación de olivos. El camino en negro es el camino real seguido por el robot (obtenido mediante el sensor diferencial de GPS incorporado en el robot); mientras que el camino en magenta es el camino estimado por el algoritmo de SLAM basado en los troncos del ambiente. La posición de cada tronco es previamente conocida (también mediante medición de posicionamiento global). Los troncos están representados por rectángulos azules; las cruces rojas asociadas a los troncos son las estimas efectuadas por el algoritmo de SLAM acerca de la posición del tronco. La posición del tronco se obtiene directamente transformando la Ec. 12 a coordenadas cartesianas globales al robot móvil. Cada tronco tiene una elipse de incertidumbre asociada, extraída directamente de la matriz de información del vector de estados del

sistema [7]. La Fig. 9(b) muestra las elipses de incertidumbres de dos troncos de la Fig. 9(a). La Fig. 9(c) muestra la evolución del determinante de la matriz de covarianza (el cual es una medida del volumen de incertidumbre asociado) de algunos troncos detectados en el entorno. Como se puede ver la incertidumbre disminuye a medida que el algoritmo de SLAM se ejecuta y el vehículo revisita características pasadas.

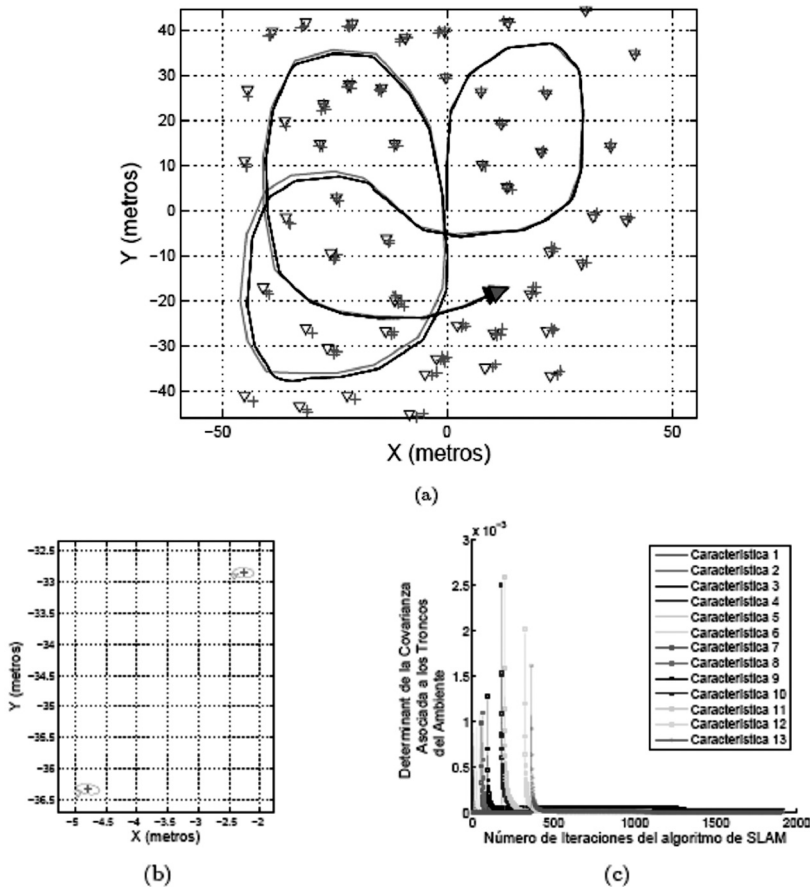


Figura 9: Resultados del algoritmo de SLAM presentado en este trabajo. La Fig. 9(a) muestra el mapa construido del entorno basándose en la información provista por el vector de estados del algoritmo de SLAM; como se puede ver, el mapa final es consistente. También se grafica la trayectoria seguida por el vehículo (en magenta) en función de la trayectoria real obtenida mediante datos del GPS diferencial montado en el robot (camino en negro). La Fig. 9(b) muestra una ampliación de una porción del mapa mostrado en la Fig. 9(a), donde la posición estimada del tronco está representada por una cruz roja con una elipse de varianza asociada a la misma. La Fig. 9(c) muestra la evolución del determinante de la covarianza asociada a algunas características del entorno. Como se puede ver, los determinantes tienden a cero a medida que el SLAM se ejecuta, lo que implica que el proceso de estimación es consistente y convergente.

5.3. Comentarios finales

El tiempo de procesamiento asociado a la extracción de troncos por cada imagen fue, en promedio, de 1 segundo. Además, el algoritmo de SLAM requirió por iteración de 0.25 segundos de procesamiento. El tiempo de procesamiento total fue por tanto de 1.25 segundos, lo cual limita la velocidad máxima de avance lineal del vehículo, la cual fue fijada en 0.3 m/s. Los algoritmos de procesamiento de imágenes como así también el algoritmo de SLAM, fueron implementados en C++ usando las librerías de *OpenCV* y *VXL*.

6. Conclusiones

En este trabajo se presentó un algoritmo de SLAM basado en el Filtro Extendido de Información, aplicado a la reconstrucción de entornos agrarios.

Las características extraídas del entorno agrícola corresponden a troncos asociados a las plantas de olivos del ambiente. El método de detección de troncos se basó en dos etapas. Una primera etapa basada en máquinas de soporte vectorial, fue la encargada de detectar los troncos en el plano de la imagen, de la cual se obtiene la información de ángulo del tronco, en el sistema de visión monocular. Una segunda etapa basada en un sensor láser fue la encargada de obtener el rango asociado a los troncos detectados por el sistema de visión artificial. La información de rango y ángulo de los troncos fue luego incorporada al algoritmo de SLAM pasando a constituir *el mapa* del entorno.

Los resultados experimentales presentados en este trabajo, muestran la viabilidad de la propuesta para aplicaciones de agricultura de precisión.

Agradecimientos

Los autores quieren agradecer a la Universidad Nacional de San Juan y a la Universidad Tecnológica Nacional - Regional Córdoba por su apoyo y por la financiación parcial de este proyecto.

Referencias

[1] R. Chatila and J. P. Laumond, "Position referencing and consistent world modeling for mobile robots," *Proc. of the IEEE International Conference on Robotics and Automation*, pp. 138–145, 1985.

- [2] N. Ayache and O. Faugeras, “Maintaining a representation of the environment of a mobile robot,” *IEEE Transactions on Robotics and Automation*, vol. 5, pp. 804–819, 1989.
- [3] H. D. Whyte and T. Bailey, “Simultaneous localization and mapping (slam): part i essential algorithms,” *IEEE Robotics and Automation Magazine*, vol. 13, pp. 99–108, 2006.
- [4] H. Whyte and T. Bailey, “Simultaneous localization and mapping (slam): part ii state of the art,” *IEEE Robotics and Automation Magazine*, vol. 13, pp. 108–117, 2006.
- [5] A. Garulli, A. Giannitrapani, A. Rosi, and A. Vicino, “Mobile robot slam for line-based environment representation,” *IEEE Conference on Decision and Control*, 2005.
- [6] J. Guivant, E. Nebot, and S. Baiker, “High accuracy navigation using laser range sensors in outdoor applications,” in *Robotics and Automation, 2000. Proceedings. ICRA '00. IEEE International Conference on*, vol. 4, 2000, pp. 3817–3822 vol.4.
- [7] S. Thrun, W. Burgard, and D. Fox, *Probabilistic Robotics*. Cambridge: MIT Press, 2005.
- [8] C. Cadena and J. Neira, “Slam in $o(\log n)$ with the combined kalman - information filter,” in *Intelligent Robots and Systems, 2009. IROS 2009. IEEE/RSJ International Conference on*, 10-15 2009, pp. 2069–2076.
- [9] G. Dissanayake, P. Newman, S. Clark, H. F. Durrant-Whyte, and M. Csorba, “A solution to the simultaneous localisation and map building (slam) problem,” *IEEE Transactions on Robotics and Automation*, vol. 7, pp. 229–241, 2001.
- [10] G. R. C. and W. R. E., *Digital Image Processing*. Addison-Wesley Publishing Company, 1993.
- [11] H. S., *Neural Networks: A comprehensive foundation*. Prentice Hall, 1999.
- [12] N. He, J. Cao, and L. Song, “Scale space histogram of oriented gradients for human detection,” vol. 2, dec. 2008, pp. 167–170.
- [13] F. A. Auat Cheein, F. di Sciascio, G. Scaglia, and R. Carelli, “Towards features updating selection based on the covariance matrix of the slam system state (in press),” *ROBOTICA (Cambridge)*, 2010.
- [14] D. A. Harville, *Matrix Algebra from a Statistician's Perspectiv*. New York: Springer-Verlag, 1997.
- [15] F. A. Auat Cheein, *Simultaneous Localization and Mapping of a Mobile Robot based on Uncertainty Zones Navigation*. San Juan Argentina: PhD Thesis, 2009.

- [16] C. K. Li and R. Mathias, “The determinant of the sum of two matrices,” *Bull. Austral. Math. Soc.*, vol. 52, pp. 425–429, 1995.
- [17] F. Masson, J. Guivant, J. Nieto, and E. Nebot, “The hybrid metric map: a solution for precision farming,” *Latin American applied research*, vol. 32, no. 2, pp. 105 – 110, 2005.
- [18] N. Dalal and B. Triggs, “Histograms of oriented gradients for human detection,” in *CVPR '05: Proceedings of the 2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05) - Volume 1*. Washington, DC, USA: IEEE Computer Society, 2005, pp. 886–893.
- [19] D. G. Lowe, “Distinctive image features from scale-invariant keypoints,” *Int. J. Comput. Vision*, vol. 60, no. 2, pp. 91–110, 2004.
- [20] V. N. Vapnik, *Statistical Learning Theory*. Wiley-Interscience, 1998.
- [21] K.-R. Muller, S. Mika, G. Ratsch, K. Tsuda, and B. Scholkopf, “An introduction to kernel based learning algorithms,” *Neural Networks, IEEE Transactions on*, vol. 12, no. 2, pp. 181–201, mar. 2001.
- [22] Z. Zhang, “Flexible camera calibration by viewing a plane from unknown orientations,” in *ICCV*, 1999, pp. 666–673.
- [23] R. Siegwart and I. Nourbahsh, *Introduction to Autonomous Mobile Robots*. Cambridge: MIT Press, 2004.

Síntesis del las instrucciones para los autores

Revista Ciencia y Tecnología

Facultad de Ingeniería. Universidad de Palermo. Buenos Aires. Argentina

Los trabajos deberán presentarse en versión impresa (original y 2 copias) y en versión digital (CD) a la Coordinadora Editorial de la revista.

La página inicial se reservará para indicar:

Título del trabajo (en negrita), centrado y en letras mayúsculas

Subtítulo, si lo hubiere, en negrita y letra minúscula

Nombre y apellido del (de los) autor (es)

Correo electrónico de el/los autor(es) y la(s) institución(es) a la(s) que pertenece(n).

El trabajo será precedido por un resumen (*abstract*) de 50 a 100 palabras en fuente cursiva (*itálica*), en donde se delinearán los aspectos más sobresalientes, que permitirán tener una comprensión general del contenido del artículo.

- Formato: textos en Word que no presenten ni sangrías ni efectos de texto o formato especiales.
- Extensión: entre 10.000 y 40.000 caracteres (con espacio).
- Títulos y subtítulos: en negrita y en Mayúscula y minúscula.
- Fuente: Times New Roman.
- Estilo de la fuente: normal.
- Tamaño: 12 pt.
- Interlineado: sencillo.
- Alineación: justificada
- Tamaño de la página: A4.
- Márgenes: superior e inferior, 2,5 cm; derecho e izquierdo, 3 cm
- Normas: se debe tomar en cuenta las normas básicas de estilo de la publicación de la American Psychological Association (normas APA).

Bibliografía y notas: en la sección final del artículo con el formato de las normas APA.

Fotografías, cuadros o figuras: deben ser presentados en formato tif a 300 dpi en escala de grises.

Importante: tener en cuenta que la imagen debe ir acompañando el texto a modo ilustrativo y dentro del artículo hacer referencia a la misma.

En caso de necesitar información adicional escribir a la Coordinadora Editorial: ingenieria@palermo.edu

