

## Localización y Mapeo Simultáneo con Imágenes 3D a Color

Miguel A. Realpe<sup>(1)</sup>, Boris Vintimilla<sup>(1)</sup>, José Gaspar<sup>(2)</sup>, José Santos-Victor<sup>(2)</sup>

<sup>(1)</sup>Centro de Visión y Robótica, <sup>(2)</sup>Instituto de Sistemas e Robótica

<sup>(1)</sup>Facultad de Ingeniería en Electricidad y Computación, <sup>(2)</sup>Instituto Superior Técnico, Torre Norte

<sup>(1)</sup>Escuela Superior Politécnica del Litoral,

<sup>(1)</sup>Campus Prosperina, Km 30.5 vía Perimetral. 0901.5863 Guayaquil, Ecuador, <sup>(2)</sup> Av. Rovisco Pais, 1 1049-001 Lisboa, Portugal

<sup>(1)</sup>mrealpe@fiec.espol.edu.ec, boris.vintimilla@espol.edu.ec, <sup>(2)</sup>{jag,jasv}@isr.ist.utl.pt

### Resumen

*El presente trabajo pretende solucionar los problemas de localización y mapeo del ambiente de navegación de un robot a partir de imágenes 3D a color. Se ha realizado la implementación de un sistema de registro de imágenes tridimensionales mediante el algoritmo ICP, el cual toma ventaja de la información de color de las imágenes 3D y determina la localización del robot en un momento determinado. Además se ha realizado la alineación de todas las imágenes adquiridas por el robot, con lo cual se produce un mapa tridimensional del ambiente por el cual navega. Resultados experimentales usando tres tipos de recorridos con un robot fueron realizados.*

**Palabras Claves:** Localización y mapeo simultáneos, Registro 3D, Imágenes 3D a color, ICP.

### Abstract

*The present work tries to solve the problems of localization and mapping of robot navigation environments from 3D color images. A register system has been implemented by means of ICP algorithm, taking advantage of the color information of the 3D images, this register system determines the robot localization at any time. Also one alignment of all the data acquired by a robot has been done; this produces a three-dimensional map of the navigation environment. Experimental results by using three types of navigation with a robot were performed.*

## 1. Introducción

La finalidad de los métodos de localización y mapeo simultáneos (SLAM) es que un robot pueda construir de forma incremental un mapa de su medio de navegación, y mientras utiliza este mapa estime la trayectoria realizada por el robot al recorrer el ambiente desconocido sin ninguna información previa.

El proceso de construcción de una imagen 3D de un ambiente real involucra dos fases, la primera consiste en la adquisición del medio y la segunda es una reconstrucción de los datos adquiridos [1].

En la fase de adquisición se obtiene información del ambiente desde varios puntos de vista por lo que es necesario realizar una reconstrucción de todos los datos adquiridos para formar un solo modelo o mapa que represente al ambiente completo. En la fase de reconstrucción se realiza el registro de todos los datos obtenidos, el registro es el proceso de alinear espacialmente los datos de dos o más muestras que pertenecen a una misma escena [2].

Los métodos de registro de imágenes 3D se pueden clasificar en dos grupos principales: métodos que registran un grupo de características extraídas de las

imágenes, y métodos que registran los puntos presentes en las imágenes [6].

Los métodos que registran un grupo de características extraídas de las imágenes buscan una forma de caracterizar las imágenes, para calcular la transformación de las coordenadas de las imágenes en el espacio de características (contornos, superficies, formas, etc.). En [3] se realiza la integración de imágenes por medio de un subconjunto de superficies, para esto se divide cada imagen 3D en varias porciones representadas por modelos locales de superficie. Luego, se proyectan los modelos locales a la vista de referencia. Los modelos provistos por distintas vistas son fusionados por una función de diferencia y finalmente son triangulados, obteniendo un modelo triangulado que representa a las imágenes de distintos puntos de vistas en un solo sistema de coordenadas dado por la vista de referencia.

Los métodos que registran los puntos presentes en las imágenes se basan en la minimización de una función de distancia entre los puntos presentes en imágenes de distintos puntos de vista. Uno de los primeros trabajos en la minimización de funciones de distancia fue presentado por Chen y Medioni [4] y es considerado como el primer acercamiento al algoritmo

de ICP (Iterative Closest Point), que fue desarrollado por Besl y McKay [5], y que en la actualidad es uno de los algoritmos más utilizados en el registro de imágenes 3D. El algoritmo ICP realiza un proceso iterativo que, para cada punto de una imagen, realiza la búsqueda del vecino más cercano en la imagen a registrar por medio de una función de minimización de distancias. De esta forma, en cada iteración acerca de manera global los dos conjuntos de datos hasta lograr que ambos converjan.

En este artículo se describen el registro y alineación de imágenes 3D a través del algoritmo de ICP, tomando en cuenta el parámetro de color de las imágenes. Se detallan los pasos realizados para obtener una matriz de transformación entre un par de imágenes y la alineación de un grupo de imágenes que representan un ambiente. Al final del artículo se muestran algunos resultados experimentales y las conclusiones obtenidas.

## 2. Localización y Mapeo

Un problema clásico en el trabajo de robots móviles autónomos que operan en entornos desconocidos es la navegación. Las actividades esenciales para llegar a la solución de este problema consisten en determinar donde se encuentra el robot dentro del medio desconocido (localización) y construir incrementalmente un mapa del entorno que ha recorrido el robot (mapeo). El mapa podrá servirle al robot para posteriormente realizar tareas como localizarse dentro de dicho entorno, planear trayectorias y otras actividades.

### 2.1 Localización

El sistema de localización permite al robot determinar en que posición se encuentra dentro de su ambiente de navegación. La localización del robot se la obtiene calculando el movimiento realizado al adquirir imágenes por medio del registro de un par de imágenes 3D consecutivas.

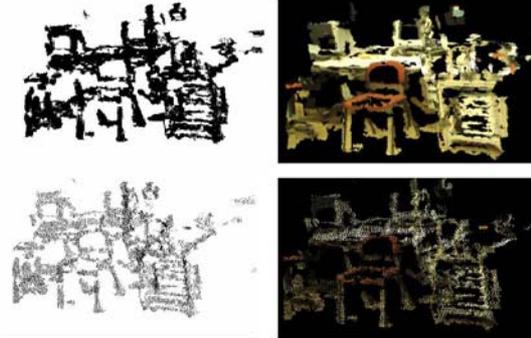
Para el presente trabajo se ha decidido realizar el registro a través del algoritmo de ICP, que registra directamente los puntos presentes en las imágenes, debido a su simpleza en relación al método de registro de un grupo de características extraídas de las imágenes.

Existen 6 etapas en las que se desarrolla el algoritmo de ICP y que se pueden cambiar para mejorar el proceso [7]. En el presente proyecto se aplicaron 4 de los pasos mencionados en [7] siguiendo un procedimiento basado en [8].

**2.1.1 Selección de puntos.** Para disminuir la cantidad de puntos a procesar se ha decidido realizar un muestreo de puntos de forma uniforme, seleccionando como puntos válidos a uno de cada diez de forma

uniforme en toda la nube de puntos, con el objetivo de obtener una información equitativa de toda la imagen.

Al realizar este muestreo se espera disminuir el tiempo de procesamiento de las siguientes etapas, aunque no se minimiza el error existente en las imágenes 3D. En la Figura 1 (fila superior) se muestra una imagen 3D representada a través de una nube de puntos, tanto a color como en blanco y negro, mientras en la fila inferior se aprecia la misma nube de puntos después de aplicar una selección uniforme de puntos



**Figura 1.** (fila superior) Nube de puntos original; (fila inferior) Nube de puntos después de selección uniforme de puntos.

**2.1.2 Correspondencia de Puntos.** Se refiere a encontrar los puntos más cercanos de una imagen en otra. La correspondencia se realiza a través del vecino más cercano. Para el desarrollo del problema del vecino más cercano se tiene como entrada un grupo de puntos en el espacio de dimensión 3. Luego, el grupo de puntos es organizado en un árbol kd ( $k$ -dimensionales) de forma tal que dado un punto cualquiera el punto más cercano pueda ser encontrado con facilidad.

Un árbol kd es un árbol binario de búsqueda en el que cada nodo representa una partición del espacio de dimensión  $d$ , para crear las divisiones en el espacio en cada nodo sólo una de las  $k$  dimensiones es utilizada como discriminador. El nodo raíz representa el espacio entero y las hojas representan subespacios que contienen subconjuntos del grupo de puntos de entrada, los cuales son mutuamente excluyentes.

Se ha utilizado la librería ANN [9] para realizar la creación y búsqueda del árbol kd. Esta librería crea el árbol kd recursivamente dividiendo el conjunto de puntos en la dimensión de mayor varianza, hasta obtener hojas con un determinado número de puntos.

El proceso de búsqueda opera recursivamente, cuando se encuentra un nodo del árbol se visita el nodo hijo que se encuentra más cerca al valor del punto consultado en la dimensión discriminante del nodo. Si el otro nodo hijo se encuentra a una distancia dentro de un porcentaje de error determinado de la distancia del vecino cercano encontrado hasta el momento, entonces el otro nodo hijo también es visitado de forma recursiva. Cuando se encuentra una hoja todos los puntos de esa hoja son evaluados en relación al punto consultado y se determina el punto más cercano.

**2.1.3 Rechazo de pares.** Consiste en desechar los pares que no cumplen con un parámetro de correspondencia. Si el registro se lo realiza entre imágenes 3D de buena calidad esta etapa no influye mucho en el desarrollo y optimización del algoritmo. Para realizar el rechazo de pares que no representan una correspondencia válida se utiliza como filtro el color de los puntos en las imágenes 3D.

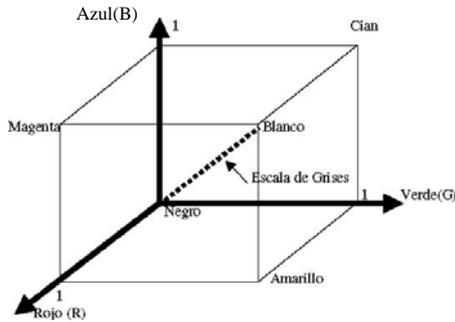


Figura 2. Espacio de color RGB.

Cuando se trabaja con imágenes de color existen grandes variaciones para representar el color, las dos más utilizadas son: RGB y HSI. Por medio de la representación RGB se descompone la imagen de color en 3 imágenes correspondientes a los componentes de color rojo, verde y azul (Red, Green, Blue), es la representación de mayor uso en el mundo informático y se encuentra orientado a los equipos. El modelo RGB puede representarse como un cubo cuyos ejes son los componentes de color, tal como se muestra en la Figura 2.

El problema que existe con la representación RGB es que los seres humanos no describimos un determinado color considerando los componentes RGB, sino mediante otras tres características que son tono, saturación y brillo. Estas tres componentes forman la representación HSI del color, la cual se asocia a la forma en que percibimos el color. El modelo HSI puede representarse como dos conos unidos en su base tal como se aprecia en la Figura 3.

Las tres componentes que utiliza el modelo HSI son descritas a continuación. El tono (Hue) indica el color, en la Figura 3 esta representado por el ángulo. La saturación (Saturation) muestra la pureza de un color determinado, lo que es igual al radio en la Figura 3. La intensidad o brillo (Intensity) es la iluminación del color, aparece como la altura en el cono de color de la Figura 3.

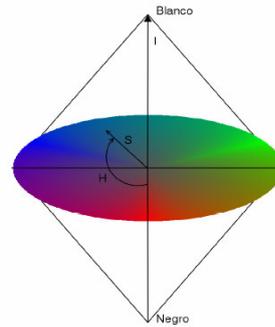


Figura 3. Espacio de color HSI.

A diferencia del modelo de color RGB, el espacio HSI separa la iluminación de la información de color, esto es, el tono (H) y la saturación (S) permanecen estables bajo variaciones en la intensidad de la iluminación, [10], permitiendo realizar un filtrado más robusto al ignorar la componente I, que representa de la iluminación de la escena.

La relación existente entre RGB y HSI se muestra en las siguientes ecuaciones:

$$H = \arccos\left(\frac{(1/2)(R-G) + (R-B)}{\sqrt{(R-G)^2 + (R-B)(G-B)}}\right) \quad \text{si } B \leq G \quad \text{Ec. 1}$$

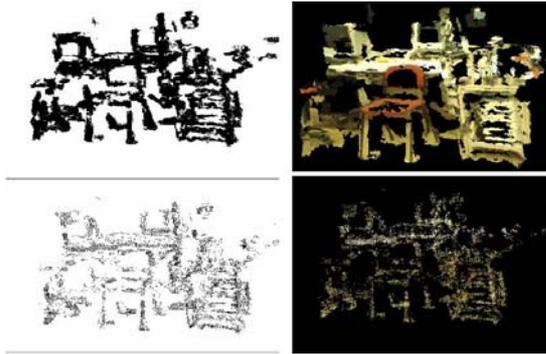
$$(2\pi - H) \quad \text{si } B > G$$

$$S = 1 - \frac{3 \min(R, G, B)}{I} \quad \text{Ec. 2}$$

$$I = R + G + B \quad \text{Ec. 3}$$

La conversión de un modelo de color a otro es un proceso muy costoso computacionalmente ya que se deben realizar tantos cálculos como número de puntos existan en la imagen 3D, por lo que se ha optado por realizar una tabla de búsqueda LUT (Look Up Table), la cual se almacena en la memoria del computador con  $2^4$  las combinaciones de todos los valores de tono y saturación posibles para cada componente RGB utilizando 8 bits.

Una vez que se han obtenido las correspondencias entre dos imágenes 3D se comparan los valores de H y S mediante una operación de lectura de la LUT y se evalúan en relación a la media de todos los puntos de la imagen, si la diferencia de los valores excede la media se descarta el par como una correspondencia, de esa forma se realiza el registro sólo con las correspondencias más robustas. En la Figura 4 se aprecia una nube de puntos después de pasar por la fase de rechazo de pares a través del método de color descrito anteriormente.



**Figura 4.** (fila superior) Nube de puntos original; (fila inferior) Nube de puntos después de rechazo de pares.

**2.1.4 Asignación y minimización de la métrica de error.** La asignación de la métrica de error se refiere a la forma en que se medirá el error entre los pares. En el presente trabajo se utiliza la suma del cuadrado de las distancias. La finalidad es encontrar la transformación, compuesta por Rotación y Translación, que minimiza la siguiente función:

$$F = \sum_{i=0}^N \|p_i - (R \cdot q_i + T)\|^2 \quad \text{Ec. 4}$$

Donde  $\{p\}$  y  $\{q\}$  son los conjuntos de pares correspondientes entre las dos imágenes 3D y R, T son las matrices de Rotación y Translación respectivamente

Para hallar la matriz de transformación se ha empleado el método basado en la descomposición de valores singulares (SVD) propuesto en [11]. Cuyo primer paso para hallar la matriz de transformación es separar la Rotación, R, de la matriz de Translación, T, para lo cual se utilizan los centros de masas o centroides de  $\{p\}$  y  $\{q\}$  mediante las siguientes ecuaciones:

$$C_p = \frac{1}{N} \sum_{i=1}^N p_i \quad \text{Ec. 5} \quad C_q = \frac{1}{N} \sum_{i=1}^N q_i \quad \text{Ec. 6}$$

Una vez que se han encontrado los centroides,  $C_p$  y  $C_q$ , se representan los dos grupos de puntos (imágenes 3D) en relación a los centros de masa para que el origen de cada grupo de puntos se encuentre en su respectivo centroide, tal como se expresa en Ec. 7 y Ec. 8

$$r_i = p_i - C_p \quad \text{Ec. 7}$$

$$s_i = q_i - C_q \quad \text{Ec. 8}$$

Al realizar el cambio de coordenadas se elimina la translación de la función a minimizar consiguiendo que la transformación entre los dos grupos de puntos sea únicamente de rotación, obteniendo así una nueva función a minimizar expresada como se muestra en Ec. 9:

$$F = \sum_{i=0}^N \|r_i - R \cdot s_i\|^2 \quad \text{Ec. 9}$$

El SVD de una matriz H de  $m \times n$  (donde  $m \geq n$ ) es una descomposición de H tal que  $H = U \cdot D \cdot V'$  donde U es una matriz de  $m \times n$  tal que  $U' \cdot U$  es igual a la identidad, D es una matriz diagonal de  $n \times n$  y V es una matriz ortogonal de  $n \times n$ .

En [11] se muestra que la rotación óptima puede ser obtenida por  $R = VU'$  donde V y U son derivadas por medio de la SVD de la matriz de correlación H y U' es la transpuesta de U. La matriz de correlación H es dada por Ec. 10

$$H = \sum_{i=1}^N s_i \cdot r_i' \quad \text{Ec. 10}$$

Una vez obtenida la Rotación, R, se procede a calcular la Translación, T, que esta dada por la siguiente expresión:

$$T = p - R \cdot q \quad \text{Ec. 11}$$

Al terminar el proceso de localización se obtiene una matriz de transformación para cada par de imágenes consecutivas, mediante la cual se representa la translación y rotación que se ha producido entre las mismas, estimando así el movimiento del robot.

## 2.2 Mapeo

El mapeo se refiere a la creación de una representación del medio que rodea al robot. El sistema de mapeo utiliza las transformaciones obtenidas por el sistema de localización e imágenes 3D de entrada para crear los mapas 3D del ambiente de navegación del robot.

En el desarrollo de este trabajo se propone un algoritmo mediante el cual se van a alinear dos nubes de puntos tridimensionales, expresadas en distintas referencias y correspondientes a un mismo medio dado. El objetivo es utilizar la matriz de transformación de cada par de imágenes 3D para representar varias nubes de puntos en un mismo sistema de coordenadas.

Para realizar la alineación de todas las nubes de puntos se debe representar las matrices de transformación en coordenadas homogéneas. Mediante las coordenadas homogéneas se trata a todas las transformaciones con operaciones de multiplicación, para esto se utiliza una coordenada adicional para cada par  $(X, Y, Z)$ , de esa forma una coordenada homogénea se define como  $(kX, kY, kZ, k)$ , donde k es una constante arbitraria distinta de cero.

**2.2.1 Translación.** La translación de un punto  $(X, Y, Z)$  utilizando coordenadas homogéneas esta dado por el producto matricial:

$$\begin{bmatrix} X_1 \\ Y_1 \\ Z_1 \\ 1 \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & T_x \\ 0 & 1 & 0 & T_y \\ 0 & 0 & 1 & T_z \\ 0 & 0 & 0 & 1 \end{bmatrix} \begin{bmatrix} X_0 \\ Y_0 \\ Z_0 \\ 1 \end{bmatrix} \quad \text{Ec. 12}$$

Donde  $(T_x, T_y, T_z)$  es el vector de translación.

**2.2.2 Rotación.** Para establecer la rotación de un punto en el espacio es necesario, primero definir el eje de coordenadas alrededor del cual se va a rotar. Así existen tres posibles rotaciones para un punto (X,Y,Z) utilizando coordenadas homogéneas que son:

$$R(x, \alpha) = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & \cos \alpha & \sin \alpha & 0 \\ 0 & -\sin \alpha & \cos \alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 13}$$

$$R(y, \beta) = \begin{bmatrix} \cos \beta & 0 & \sin \beta & 0 \\ 0 & 1 & 0 & 0 \\ -\sin \beta & 0 & \cos \beta & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 14}$$

$$R(z, \theta) = \begin{bmatrix} \cos \theta & \sin \theta & 0 & 0 \\ -\sin \theta & \cos \theta & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 15}$$

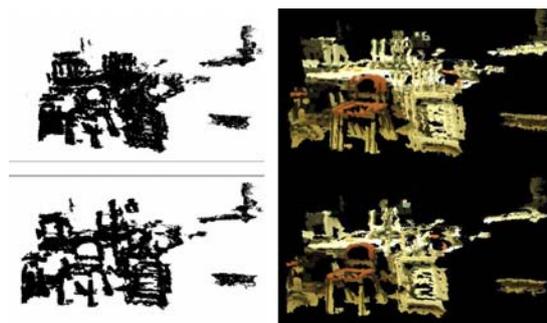
Debido a que la variación en el orden de la rotación de un punto produce resultados distintos es necesario utilizar un orden en la composición de rotaciones tal como se muestra en Ec. 16.

$$R = \begin{bmatrix} c\theta c\beta & -s\theta c\alpha + c\theta s\beta s\alpha & s\theta c\alpha + c\theta s\beta c\alpha & 0 \\ s\theta c\beta & c\theta c\alpha + s\theta s\beta s\alpha & -c\theta c\alpha + s\theta s\beta c\alpha & 0 \\ -s\beta & c\beta s\alpha & c\beta c\alpha & 0 \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 16}$$

**2.2.3 Alineación.** Las matrices de Rotación, R, y Translación, T, son obtenidas mediante el método de SVD y luego son representadas en coordenadas homogéneas mediante una sola matriz como se muestra en Ec. 17

$$T(R,T) = \begin{bmatrix} c\theta c\beta & -s\theta c\alpha + c\theta s\beta s\alpha & s\theta c\alpha + c\theta s\beta c\alpha & Tx \\ s\theta c\beta & c\theta c\alpha + s\theta s\beta s\alpha & -c\theta c\alpha + s\theta s\beta c\alpha & Ty \\ -s\beta & c\beta s\alpha & c\beta c\alpha & Tz \\ 0 & 0 & 0 & 1 \end{bmatrix} \quad \text{Ec. 17}$$

Para alinear en un mismo sistema de coordenadas a varias imágenes 3D pertenecientes a un ambiente se multiplica la matriz de transformación correspondiente y se une de forma recursiva pares de imágenes. En la Figura 5 se muestran dos nubes de puntos de un mismo medio, representadas en su propio sistema de coordenadas (fila superior) y una vez alineadas al multiplicar una de las nubes de puntos por su respectiva matriz de transformación (fila inferior).



**Figura 5.** (fila superior) Par de nubes de puntos consecutivas; (fila inferior) Nubes de puntos alineadas a partir de la matriz de transformación.

### 3. Resultados

El robot utilizado para realizar las pruebas experimentales fue el TRC Labmate de HelpMate Robotics Inc, el cual consiste en una plataforma móvil de control diferencial al que se le ha montado la cámara de estéreo visión Digiclops. La Figura 6 muestra el montaje de la cámara sobre el TRC.



**Figura 6.** Sistema de visión estéreo montado sobre robot móvil.

Las pruebas experimentales consistieron en tres recorridos con el TRC, donde se adquirieron una serie de imágenes rectificadas de la cámara izquierda, derecha y superior de la cámara de estéreo visión Digiclops y se obtuvieron imágenes 3D a través de nubes de puntos, tal como se explica en [12]. A continuación se presentan tres pruebas realizadas, cada una con un recorrido diferente.

#### 3.1 Prueba 1: Translación lateral

En la primera prueba el robot realizó una trayectoria de translación lateral, teniendo un desplazamiento total de aproximadamente 2.0 metros en el que se obtuvo 76 imágenes 3D. En este recorrido el error acumulado obtenido es aproximadamente 4 cm. en el eje de desplazamiento y 14 cm. en el eje ortogonal. En la Figura 7 se muestra el recorrido realizado por el robot (rojo) y la estimación realizada en base al algoritmo de SLAM (azul).

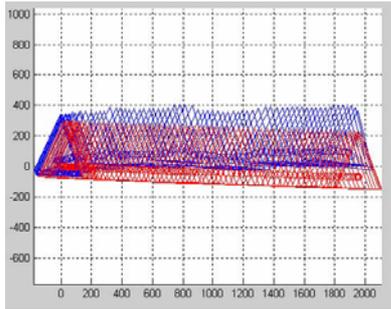


Figura 7. Trayecto lateral realizado por el robot.

En la Figura 8 se muestran la primera y la última imagen obtenida en el recorrido del robot y sus respectivas nubes de puntos



Figura 8. (columna izquierda) Imagen a color del ambiente; (columna derecha) nubes de punto resultantes.

La Figura 9 muestra la reconstrucción completa del ambiente (mapa 3D del ambiente) realizada a partir de las 76 imágenes obtenidas en el movimiento lateral del robot realizado en la prueba 1.



Figura 9. Mapa 3D de ambiente de navegación de la prueba 1.

### 3.2 Prueba 2: Translación frontal

Durante el desarrollo de la segunda prueba el robot recorrió una trayectoria de translación hacia delante, desplazándose un total aproximado de 1.5 metros y obteniendo 60 imágenes 3D. El error acumulado obtenido es aproximadamente 5 cm. en el eje de desplazamiento y 12 cm. en el eje ortogonal. El recorrido realizado por el robot (rojo) y la estimación realizada en base al algoritmo de SLAM (azul) son mostrados en la Figura 10.

En la Figura 11 se muestran la primera y la última imagen obtenida en el recorrido del robot y sus respectivas nubes de puntos.

La Figura 12 muestra la reconstrucción completa del ambiente (mapa 3D del ambiente) realizada a partir de las 60 imágenes obtenidas en el movimiento de translación hacia delante del robot.

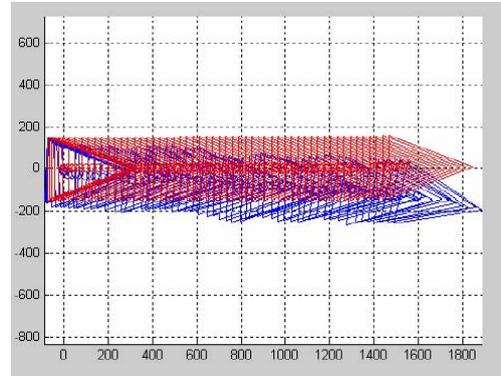


Figura 10. Trayecto hacia delante realizado por el robot.

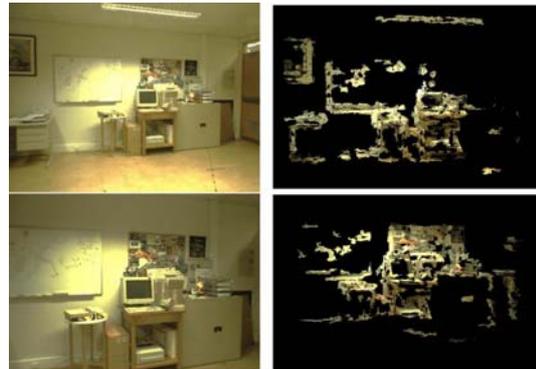


Figura 11. (columna izquierda) Imagen a color del ambiente; (columna derecha) nubes de punto resultantes.

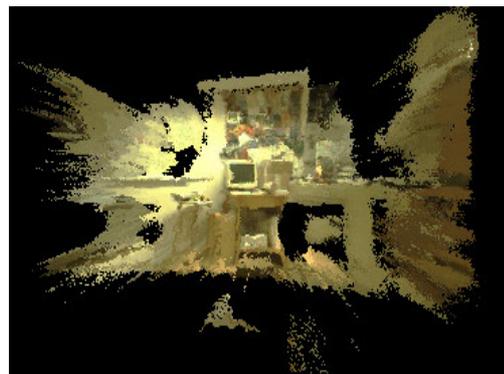
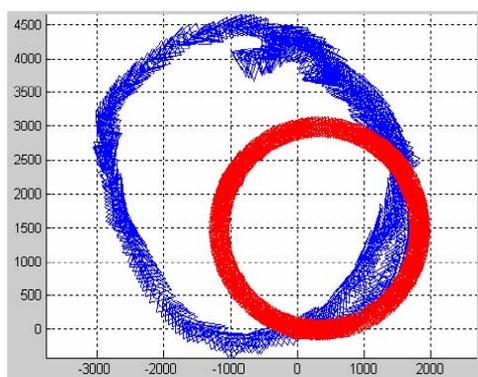


Figura 12. Mapa 3D de ambiente de navegación del robot de la prueba 2.

### 3.3 Prueba 3: Movimiento circular

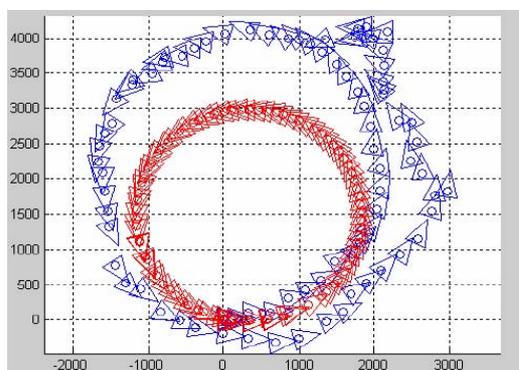
En la tercera prueba se realizó un recorrido que mezcla rotación y translación, el recorrido elegido fue

una trayectoria de dos vueltas alrededor de una circunferencia de aproximadamente 3 metros de diámetro. En la Figura 13 se muestra el recorrido realizado por el robot y la estimación realizada.



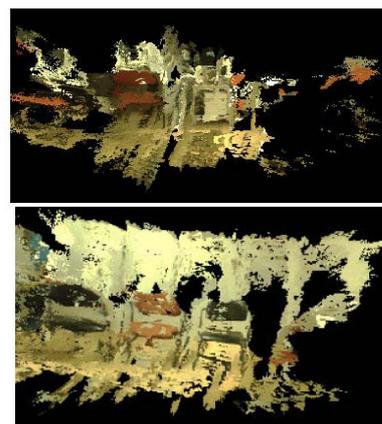
**Figura 13.** Trayectoria circular realizada por el robot.

En la Figura 13 se aprecia que el error de cada par de imágenes se acumula creando un error total cada vez mayor. Se trató de reducir la acumulación de error utilizando mejores estimaciones iniciales en el cálculo de las matrices de rotación y translación, para esto se tomaron las matrices obtenidas anteriormente y se obtuvo el movimiento realizado cada 5 imágenes, luego se utilizó esta información como estimación inicial en un nuevo cálculo de la matriz de movimiento, obteniendo así menores errores (Figura 14).



**Figura 14.** Trayectoria calculada de forma iterativa cada 5 imágenes.

La Figura 15 muestra la reconstrucción de algunos sectores del entorno de navegación del robot, estimados durante la realización de la prueba 3.



**Figura 15.** Segmentos del mapa 3D del ambiente de navegación de la prueba 3.

#### 4. Conclusiones

Durante el recorrido de la prueba 1 y 2, el error total de la estimación se encuentra entre el 7% y 9%, obteniendo en cada movimiento del robot una localización bastante aproximada a la realidad y un mapa lo suficientemente bueno como para que le robot navegue autónomamente sin ningún problema, aunque aparecen obstáculos extras debido a la redundancia de datos al momento de realizar el registro y alineación de imágenes, pero esta información extra no influirán mayormente en el robot al momento de navegar autónomamente ya que son valores pequeños. Se puede apreciar, también, que el error aumenta notablemente al introducir la rotación en el movimiento, esto se debe principalmente a los puntos aislados (outliers) que se obtienen durante la etapa de adquisición, los cuales mantienen el mismo patrón durante el desplazamiento, pero varían al cambiar el ángulo de adquisición de la imagen. Al obtener una gran cantidad de outliers en dos imágenes consecutivas se dificulta el registro, obteniendo así mayores errores.

Para mejorar el registro de imágenes se prevé utilizar un método de muestreo que permita eliminar puntos aislados (outlier) en la imagen. De esta forma, el proceso de registro de nubes de puntos trabajará sobre valores con poco ruido y se obtendrán resultados con mayor precisión. Una opción puede ser el algoritmo de RANSAC.

Además se plantea implementar un registro simultáneo de varias imágenes 3D, logrando una disminución del error global de registro al reducir o eliminar la acumulación del error para cada par de conjuntos de datos

#### 5. Referencias

- [1] T. Masuda, N. Yokoya, "A robust method for registration and segmentation of multiple range images", *Computer Vision and Image Understanding*, pp. 295-307, vol.61, n°3, 1995.

- [2] T. Jost, H. Hugli, "A multi-resolution scheme icp algorithm for fast shape registration", International Symposium on 3D Data Processing Visualization and Transmission, pp 540– 543, 2002.
- [3] M. Soucy, D. Laurendeau, "A general surface approach to the integration of a set of range views", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 17, no. 4, pp. 344– 358, 1995
- [4] Y. Chen, G. Medioni, "Object modelling by registration of multiple range images", IEEE Conf. on Robotics and Automation, 1991
- [5] P. Besl, N.D. McKay, "A method for registration of 3-d shapes", IEEE Transactions on Pattern Analysis and Machine Intelligence, vol. 14, no. 2, pp. 239–256, Febrero, 1992
- [6] Brian Lee Curless, "New methods for surface reconstruction from range images a dissertation for the degree of doctor of philosophy", Junio 1997
- [7] S. M. Rusinkiewicz, M. Levoy. "Efficient variants of the ICP algorithm", Reporte técnico, Stanford University, <http://graphics.stanford.edu/papers/fasticp/>, 2001
- [8] H. Surmann, A. Nuchter, K. Lingemann, J. Hertzberg, "6D SLAM with Approximate Data Association", ICRA, 2005
- [9] David Mount, Sunil Arya, "ANN: A Library for Approximate Nearest Neighbor Searching", <http://www.cs.umd.edu/~mount/ANN/>, 2006.
- [10] F. Perez, C. Kock, "Toward color image segmentation in analog VLSI: Algorithm and hardware", International Journal of Computer Vision, vol. 12, pp.17–42, 1994
- [11] K. S. Arun, T. S. Huang, S. D. Blostein, "Least square fitting of two 3-d point sets", IEEE Transactions on Pattern Analysis and Machine Intelligence, 9(5):698-700, 1987.
- [12] M.A. Realpe, B.X.Vintimilla, "Modelado 3D de Superficies Mediante Visión Estereo", 1er Congreso de Ciencia y Tecnología, Escuela Superior Politécnica del Ejercito (ESPE), Ecuador, Octubre 2005.