https://doi.org/10.17979/spudc.9788497498081.0049

GENERACIÓN DE *DATA SETS* SIMULANDO DIFERENTES TIPOS DE CÁMARAS EN ENTORNOS VIRTUALES

Y. Berenguer, L. Payá, L. M. Jiménez, M. Ballesta y O. Reinoso
Departamento de Ingeniería de Sistemas y Automática.
Universidad Miguel Hernández de Elche. Avda. de la Universidad s/n. 03202, Elche (Alicante), Spain.
{yberenguer, lpaya, luis.jimenez, m.ballesta, o.reinoso}@umh.es,

Resumen

En este artículo se presenta una plataforma para crear diferentes tipos de data sets a partir de entornos virtuales. Estos data sets contienen información sobre la apariencia visual del entorno y sobre la distancia desde algunas posiciones de referencia hasta todos los objetos. Los algoritmos diseñados para la creación de mapas y la localización de robots deben ser probados y testados con diferentes conjuntos de imágenes para validarlos. Esto puede llevarse a cabo utilizando imágenes reales, sin embargo, un cambio en los parámetros del sistema de visión supondría su sustitución y se tendrían que volver a capturar nuevos data sets. Esto supone un alto coste y frena el avance en las primeras fases del desarrollo. El objetivo de este trabajo es desarrollar una herramienta versátil que permita generar data sets para testar eficientemente estos algoritmos. Otra ventaja de esta plataforma es que pueden generarse imágenes desde cualquier posición del entorno y con cualquier rotación. Además, las imágenes generadas no tienen ruido; lo que permite llevar a cabo una prueba preliminar en condiciones ideales. El entorno virtual se puede crear y modificar fácilmente. Por último, la plataforma permite realizar otras tareas avanzadas utilizando las imágenes generadas y el entorno virtual.

Palabras clave: *data sets*, generación de mapas, entornos virtuales, sistemas de visión, descripción de imágenes, imágenes omnidireccionales, nube de puntos.

1. INTRODUCCIÓN

Hoy en día, hay muchos tipos de robots móviles que tienen que llevar a cabo diferentes tareas de forma autónoma en un entorno desconocido, por lo tanto, deben llevar a cabo dos pasos fundamentales. Por un lado, el robot debe crear una representación interna del entorno (es decir, un mapa) y por otra parte debe ser capaz de utilizar este mapa para estimar su pose actual (posición y orientación). El robot extrae información del entorno desconocido utilizando los diferentes sensores que lleve equipados. Esta información se compara con los datos de los mapas para estimar la pose del robot. Para llevar a cabo estas tareas se pueden utilizar diferentes tipos de sensores como por ejemplo: sensores láser, táctiles o visuales.

A lo largo de los últimos años han habido infinidad de investigaciones sobre creación de mapas y localización de robots utilizando diferentes tipos de sensores. En muchas de estas investigaciones se han utilizado sensores visuales, ya que permiten muchas configuraciones posibles y le proporcionan al robot información muy completa del entorno que se puede utilizar en otras tareas de alto nivel (por ejemplo, detección de personas, identificación de semáforos, etc.) [6, 18]. Entre ellos, algunos trabajos se centran en las imágenes con información visual y métrica, como imágenes RGB-d. Dos ejemplos de esto se muestran en [16, 13] que utilizan este tipo de información en las tareas de creación de mapas y seguimiento.

Este trabajo se centra en la generación de imágenes virtuales que utilizan diferentes tipos de sensores visuales con diferentes configuraciones. Esto tiene un propósito muy útil; la generación de *data sets* de imágenes para diseñar y mejorar los algoritmos que utilizan cualquier tipo de información visual. Para llevar a cabo esta generación, se ha desarrollado una plataforma para crear *data sets* de imágenes que cambian el tipo de cámara y todos los parámetros del sistema de visión. Estos sistemas pueden ser cámaras simples, cámaras estéreo, cámaras panorámicas o sistemas de visión catadióptricos, que proporcionan al robot escenas omnidireccionales del entorno [19]. Podemos encontrar muchos trabajos previos que usan imágenes omnidireccionales en tareas de creación de mapas y localización. Por ejemplo, [17] presenta una comparación entre dos métodos diferentes de SLAM (*Simultaneous Localization And Mapping*) visual mediante el uso de imágenes omnidireccionales y en [8] se propone un sistema de navegación topológico mediante visión omnidireccional. El sistema virtual catadióptrico se compone de una cámara que apunta a un espejo hiperbólico.

También hay que tener en cuenta el hecho de que, hoy en día, vehículos aéreos no tripulados (UAVs (*Unmanned Aerial Vehicles*)) se han convertido en plataformas muy populares y versátiles que pueden realizar infinidad de tareas. Algunos investigadores se han enfrentado anteriormente el problema de la localización con

este tipo de plataforma, como [9].

Tradicionalmente, los avances en creación de mapas y localización de robots móviles utilizando sensores visuales se basan en la extracción y la descripción de algunos puntos característicos de las escenas, como por ejemplo descriptores SIFT (*Scale-Invariant Feature Transform*) [7] y SURF (*Speeded-Up Robust Features*) [3]. Más recientemente algunos autores proponen el uso de la información global de las escenas para crear descriptores. Estas técnicas han demostrado ser una buena opción para resolver los problemas de localización y de navegación. [5], [12] y [20] proponen tres ejemplos de ello.

En todos estos trabajos, es necesario contar con un conjunto completo de escenas para validar los algoritmos de creación de mapas y localización que utilizan información visual. Tradicionalmente, el robot está equipado con un sistema de visión y es controlado a lo largo del entorno para asignar posiciones a cada una de las imágenes que va capturando para hacer la creación del mapa.

Este método presenta algunas desventajas, como por ejemplo que el proceso para obtener los *data sets* es lento y caro. Además, es bastante difícil saber con precisión las coordenadas de las posiciones en las que se adquiere cada imagen. Además, para probar el efecto que pueden tener sobre el algoritmo algunos cambios en la geometría del sistema de visión, es necesario capturar nuevos conjuntos de escenas para cada nueva geometría.

En este trabajo se implementa una plataforma para crear información visual que simula diferentes sistemas visuales montados en el robot. Esta plataforma es útil para crear conjuntos de escenas sin ningún coste adicional. Estos conjuntos pueden utilizarse para validar cualquier nuevo algoritmo de creación de mapas y localización que utilice información visual. Estos mapas pueden ser creados usando diferente número de imágenes y diferentes tipologías de mapas, como los mapas de trayectoria o de cuadrícula. Por otra parte, estos mapas pueden contener tantas imágenes como sea necesario. La plataforma también permite cambiar la geometría y la configuración del sistema de visión, además de otras características de los entornos.

Además, para probar los algoritmos de localización en un mapa previamente construido, es posible generar imágenes de prueba desde cualquier posición en el mapa simulando cualquier rotación u orientación de la plataforma robótica en el entorno (6 grados de libertad). Otra ventaja de esta plataforma es que las imágenes creadas no tienen ningún tipo de ruido o imperfección porque son generadas a partir de un entorno virtual definido previamente. Permite probar los diferentes algoritmos en condiciones ideales, lo que puede ser útil en las etapas iniciales del diseño y puesta a punto de un nuevo algoritmo. Además, el ruido y las oclusiones se pueden añadir después de probar la robustez de los algoritmos, una vez que ya se han ajustado con imágenes ideales.

De esta manera, esperamos que el uso de esta plataforma ahorre tiempo y dinero durante el desarrollo de nuevos algoritmos de creación de mapas y localización de robots. Gracias a ello, los experimentos iniciales pueden ser llevados a cabo rápidamente, en una gran variedad de entornos y con precisión.

El resto de este trabajo se estructura de la siguiente manera. En la Sección 2 se introducen los sistemas de visión que utilizamos para crear las imágenes: La cámara sencilla, la cámara estéreo, la cámara panorámica y el sistema de visión omnidireccional. La Sección 3 presenta el algoritmo que hemos diseñado para simular los sistemas de visión. La Sección 4 describe algunas opciones adicionales de la plataforma ofrece. En la Sección se presentan los experimentos y resultados. Y por último en la Sección se exponen las conclusiones.

2. SIMULACIÓN DE SISTEMAS VISUALES

Hoy en día, existen varios tipos de sensores visuales. La plataforma que se presenta en este trabajo simula algunos de ellos, en particular, cámaras simples, cámaras estéreo, cámaras panorámicas y sistemas de visión catadióptrico.

La simulación de todos estos sistemas se basa principalmente en la trayectoria del haz de luz que parte de los objetos y viaja hacia el foco de la cámara. Todos estos sistemas de visión se utilizan muy comúnmente en innumerables trabajos. El sistema catadióptrico presenta el proceso de formación de la imagen más complejo. Por esta razón, el algoritmo de simulación se explica utilizando este tipo de sistema de visión como base. El resto de los sistemas de visión se simulan utilizando los mismos conceptos.

2.1. Sistema de Visión Omnidireccional

Las imágenes omnidireccionales (Figura 1) se crean usando un sistema de visión catadióptrico que consta de una cámara montada en el robot que apunta generalmente a un espejo hiperbólico (Figura 2). El eje óptico de la cámara y el eje del espejo, están alineados. La cámara captura la imagen reflejada en el espejo hiperbólico que forma la imagen omnidireccional. La figura 3 muestra como un punto del espacio se refleja en el espejo $\overrightarrow{P}$ y posteriormente proyectada al plano de la imagen, p. (i, j) es el sistema de referencia del plano imagen y (x_c, y_c, z_c) es el sistema de referencia del mundo. La figura muestra algunos de los parámetros más relevantes del espejo hiperbólico y de la cámara.

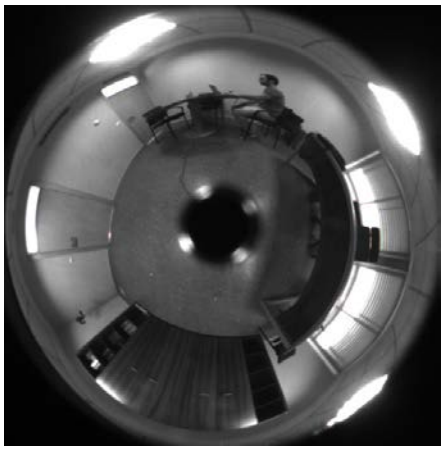


Figura 1: Imagen omnidireccional real.

Figura 2: Sistema real de adquisición de imágenes omnidireccionales.

El algoritmo está basado en la definición de la trayectoria del rayo de luz que va desde los objetos del entorno hasta el foco de la cámara (F). Primero, el rayo parte desde los objetos y llega al espejo hiperbólico. Tras ello, el rayo rebota en el espejo y parte hacia el foco de la cámara (F) pasando por algún punto del plano imagen.

La Ecuación 1 define el espejo hiperbólico. Este espejo es simétrico y sus dimensiones se definen a partir de a y b. Estas variables también definen la distancia entre el foco del espejo hiperbólico y el origen del sistema de coordenadas del mundo, c (Ecuación 2) .

$$\frac{x_c^2+y_c^2}{a^2}-\frac{z_c^2}{b^2}=-1 \tag{1}$$

$$c=\sqrt{a^2+b^2} \tag{2}$$

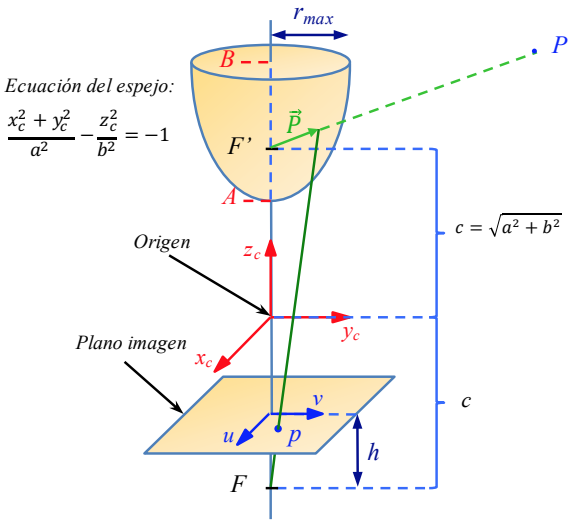


Figura 3: Sistema catadióptrico utilizado para capturar las imágenes omnidireccionales sintéticas usando un espejo hiperbólico.

Los espejos hiperbólicos son espejos muy utilizados en la creación imágenes omnidireccionales gracias a sus propiedades [21]. Sin embargo, hay muchos otros tipos de sistemas de visión que proporcionan imágenes omnidireccionales, tales como espejos parabólicos, espejos esféricos o espejos cónicos [10] e incluso matrices polares formadas por cámaras [14].

La Ecuación 3 define la ecuación de espejos parabólicos y la Ecuación 4 define la ecuación de los espejos esféricos..

$$\frac{z_c}{c}=\frac{x_c^2}{a^2}+\frac{y_c^2}{b^2} \tag{3}$$

$$(x_c-x_0)^2+(y_c-y_0)^2+(z_c-z_0)^2=r^2 \tag{4}$$

donde (x_0,y_0,z_0) son el origen de coordenadas de la esfera con respecto al sistema de referencia (x_c,y_c,z_c); a, b, c definen la geometría del espejo parabólico y r es el radio de la esfera.

A partir de una imagen omnidireccional es posible obtener su proyección panorámica utilizando un programa sencillo para transformar las coordenadas polares de la imagen omnidireccional en coordenadas cartesianas de la imagen panorámica. La Figura 4 muestra la imagen panorámica obtenida a partir de la imagen omnidireccional mostrada en la figura 1. La plataforma también permite obtener otras proyecciones diferentes, tales como vistas ortográficas, cilíndrica y proyecciones en esfera unitaria. Algunos algoritmos de creación de mapas y localización usan este tipo de información [1].

Figura 4: Imagen panorámica obtenida de la imagen omnidireccional de la Figura 1.

3. ALGORITMO DE SIMULACIÓN

En este trabajo, los sistemas de visión se modelan mediante un algoritmo para crear imágenes en un entorno virtual. Para realizar esta tarea, el programa se basa en la trayectoria del rayo de luz y en las intersecciones entre planos y líneas rectas.

Para generar una imagen, es necesario crear un entorno virtual primero. Este entorno tiene que ser definido de tal manera que la intersección entre los rayos y los objetos dentro de este entorno se puedan simular de manera eficiente. Para ello hemos definido los objetos como grupos de caras y cada una de estas caras está contenida en un plano diferente. Por ejemplo, seis caras en seis planos diferentes definen un cubo. Desde este punto de vista, el entorno virtual está formado por un conjunto de caras en el espacio que definen objetos con diferentes formas. La figura 5 muestra los elementos que forman un paralelepípedo usando estas caras; el paralelepípedo se define por l_1, l_2, l_3, su posición en el entorno y el color de cada cara.

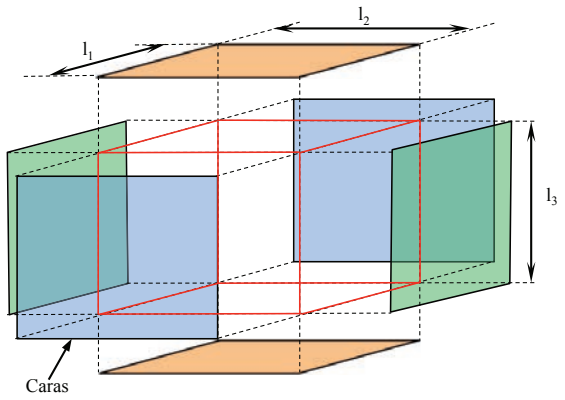


Figura 5: Caras de un objeto en un entorno virtual.

Una vez que se han generado todos los objetos, el algoritmo genera las imágenes mediante el cálculo de cada una de las trayectorias generadas por los rayos de luz en el sistema.

Teniendo en cuenta que el sistema de visión más complejo es el sistema catadióptrico, el algoritmo se explica usando la visión omnidireccional como base para la explicación. Los otros sistemas han sido simulados utilizando los mismos conceptos.

En primer lugar, el plano imagen se define escogiendo la resolución de la imagen omnidireccional (k_x x k_y) y la distancia h en la figura 3 (distancia entre el plano de la imagen y el foco de la cámara).

En segundo lugar, el vector $\overrightarrow{Fp_{ij}}$ se calcula desde el foco de la cámara F hasta cada píxel de la imagen (i, j) (Ecuación 5). p_{ij} es el píxel seleccionado para trazar el rayo. La figura 6 muestra el plano de la imagen y la trayectoria del rayo para cada pixel p_{ij}.

$$\overrightarrow{Fp_{ij}} = \overrightarrow{p}_{ij} - \overrightarrow{F} \tag{5}$$

donde $\overrightarrow{p}_{ij}$ y $\overrightarrow{F}$ son los vectores cuyas componentes son las coordenadas de p_{ij} y F respectivamente, con respecto al sistema de referencia del mundo $\{x_c, y_c, z_c\}$.

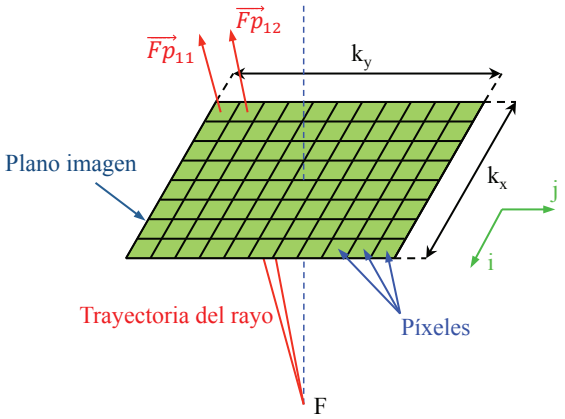


Figura 6: Trayectoria del rayo de luz para cada píxel del plano imagen.

En tercer lugar, la línea recta $r1_{ij}$ se define a partir del vector $\overrightarrow{Fp_{ij}}$ y el punto F (como punto de esa línea), se utiliza para calcular el punto de intersección (Q_{ij}) entre $r1_{ij}$ y el espejo hiperbólico (Ecuación 1).

En cuarto lugar, este punto (Q_{ij}) se utiliza para calcular el vector $\overrightarrow{P}_{ij}$ usando la siguiente ecuación:

$$\overrightarrow{P}_{ij} = \overrightarrow{Q}_{ij} - \overrightarrow{F'} \tag{6}$$

donde $\overrightarrow{Q}_{ij}$ y $\overrightarrow{F'}$ son los vectores cuyas componentes son las coordenadas de Q_{ij} y F', respectivamente, con respecto al sistema de referencia $\{x_c, y_c, z_c\}$. F' es el foco del espejo hiperbólico (Figura 3).

Por último, la línea recta $r2_{ij}$ definida por el vector $\overrightarrow{P}_{ij}$ y el punto F' se utiliza para calcular el punto de intersección (P) entre ella y cualquier objeto en el entorno. Cuando $r2_{ij}$ atraviesa una cara de un objeto, el píxel del plano imagen utilizado en la Ecuación 5 toma el valor del color de la cara de dicho objeto. $r2_{ij}$

puede atravesar varios objetos, pero sólo se considera la intersección más cercana (esto es lo que sucede en una situación real).

Por lo tanto, este proceso crea un conjunto de vectores compuesto por todos los vectores $\overrightarrow{P}_{ij}$, uno por cada píxel del plano imagen, y un punto (F'). Las intersecciones con los objetos del entorno se calculan utilizando las líneas definidas por estos vectores y el punto (F').

Para simular la translación del robot sólo es necesario trasladar el punto (F'). El conjunto de vectores no se modifica. El cambio en la elevación del robot se simula mediante una traslación en el eje z. La siguiente ecuación muestra la traslación del punto F':

$$\overrightarrow{F'_T} = \overrightarrow{F'} + \overrightarrow{T} \tag{7}$$

donde $\overrightarrow{F'_T}$ es el vector cuyas componentes son las coordenadas de F'_T, con respecto al sistema de referencia $\{x_c, y_c, z_c\}$. F'_T es el punto F' trasladado y T es el vector de traslación.

Por último, para tener en cuenta que el robot puede tener diferentes orientaciones en el espacio, es necesario el uso de una o más matrices de rotación para transformar cada vector $\overrightarrow{P}$:

$$R_x(\theta_x) = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos\theta_x & -\operatorname{sen}\theta_x \\ 0 & \operatorname{sen}\theta_x & \cos\theta_x \end{bmatrix} \tag{8}$$

$$R_y(\theta_y) = \begin{bmatrix} \cos\theta_y & 0 & \operatorname{sen}\theta_y \\ 0 & 1 & 0 \\ -\operatorname{sen}\theta_y & 0 & \cos\theta_y \end{bmatrix} \tag{9}$$

$$R_z(\theta_z) = \begin{bmatrix} \cos\theta_z & -\operatorname{sen}\theta_z & 0 \\ \operatorname{sen}\theta_z & \cos\theta_z & 0 \\ 0 & 0 & 1 \end{bmatrix} \tag{10}$$

$$R_{total} = R_x(\theta_x) \cdot R_y(\theta_y) \cdot R_z(\theta_z) \tag{11}$$

$$\overrightarrow{PR}_i = R_{total} \cdot \overrightarrow{P}_i \tag{12}$$

donde R_x, R_y y R_z son la matriz de rotación con respecto a cada eje, R_{total} es la matriz de rotación resultante y $\overrightarrow{PR}$ es el vector $\overrightarrow{P}$ girado.

El esquema del proceso de generación de imágenes omnidireccionales se define en el Código 1.

Código 1: Pseudocódigo de simulación

```
para i=1:1:px
  para j=1:1:py
    Elegir el punto p(i,j) del plano
        imagen
    Cálculo de v_Fp(i,j) = p(i,j)–F
    v_Fp y F definen r_1
    pi(i,j) = intersección entre r1 y
        el espejo hiperbólico
    Cálculo de v_P(i,j)=Q(i,j)–F'
    v_P y F' definen r_2
    P=primer punto de intersección
        entre la línea r2 y un objeto
        del entorno virtual
    imagen(i,j)=color del objeto en el
        punto P
  fin
fin
```

4. OPCIONES ADICIONALES DE CAPTURA Y DESCRIPCIÓN DE LAS IMÁGENES

El objetivo principal de este trabajo es presentar una plataforma con el fin de crear imágenes a partir de un entorno virtual como se presenta en la sección anterior. Además de esta función, la plataforma se puede llevar a cabo otras tareas avanzadas utilizando los fundamentos presentados en la Sección 3.

Ya que esta plataforma ha sido diseñada para probar algoritmos de localización y creación de mapas, se han incluido algunos de los métodos de descripción de imagen más utilizados. Entre ellos podemos encontrar descriptores basados en métodos de extracción de características y descriptores basados en la apariencia global de las imágenes.

Por un lado, la plataforma permite extraer y describir tanto características SIFT (*Scale-Invariant Feature Transform*) [7] como características SURF (*Speeded-Up Robust Features*) [3] de cada imagen omnidireccional virtual, para llevar a cabo experimentos usando este tipo de descriptores. Todos los parámetros de dichos descriptores son configurables.

Por otra parte, la plataforma también permite calcular otros descriptores basados en la apariencia global de las imágenes. Algunos ejemplos son los descriptores basados en la transformada de Radon [15, 4], la Firma de Fourier (FS, *Fourier Signature*), Análisis de Componentes Principales (ACP), Histograma de Gradientes Orientados (HGO) y descriptores *gist*. Muchos de estos descriptores se utilizan en [11]. Y también, el usuario puede configurar todos los parámetros.

La creación de cada entorno es muy fácil de reali-

zar a través de líneas de comando. Es posible crear cualquier nuevo objeto definiendo su forma, posición, orientación y tamaño. Los entornos virtuales pueden ser definidos como entornos de interior o de exterior.

Esto es muy útil para probar los algoritmos de localización en condiciones realistas. Por lo general, las imágenes de mapa son capturados en un momento específico del día, pero la localización debe llevarse a cabo en diferentes momentos. Esto implica diferentes condiciones de iluminación y también otros cambios en la información visual, tales como oclusiones en escenas debido a la presencia de personas u otros objetos móviles alrededor del robot.

Teniendo en cuenta estos hechos, la plataforma tiene también otras opciones configurables en la generación de las imágenes, tales como la adición de puntos de luz, ruido y oclusiones (añadiendo objetos en el entorno virtual). También es posible cambiar el color de cada objeto en el entorno virtual.

Las imágenes omnidireccionales también se pueden transformar en diferentes proyecciones (como vista ortográfica, cilíndrica y proyección en esfera unitaria) utilizando la plataforma presentada. Es una característica interesante porque hay muchos trabajos que usan imágenes panorámicas y otras proyecciones en las tareas de localización [11].

Por último, cabe destacar que hoy en día, algunos investigadores hacen uso de los datos de nubes de puntos en tareas de localización, tales como [2]. Por este motivo se ha añadido a la plataforma una opción adicional para generar una nube de puntos del entorno virtual. Este proceso consiste en guardar todos los puntos P (punto de intersección entre cada línea r_2 y los objetos en el entorno virtual). Estos datos de nube de puntos almacenan las coordenadas x, y y z de cada punto P y el color del objeto en este punto y emula la información capturada por una cámara RGB-d. La figura 7 muestra un *data set* de nube de puntos de un entorno virtual usando una cámara simple virtual.

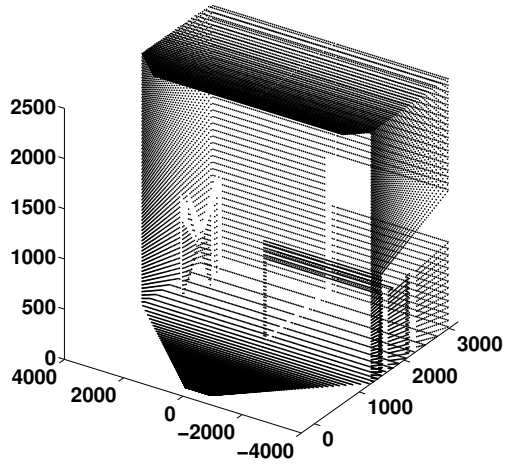


Figura 7: Ejemplo de una nube de puntos de un entorno virtual.

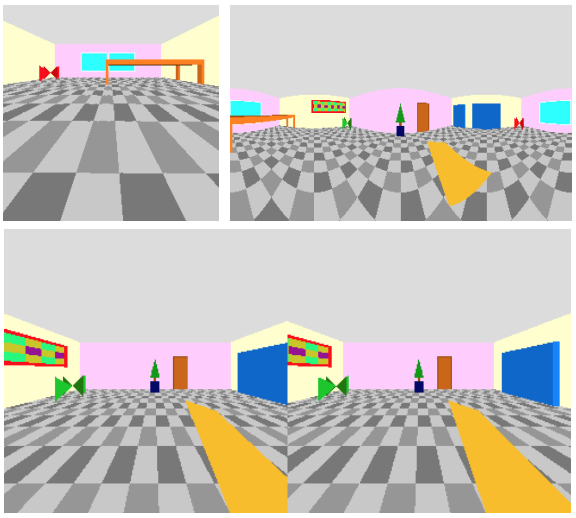


Figura 8: Diferentes tipos de sensores visuales. Arriba a la izquierda, cámara simple. Arriba a la derecha, cámara panorámica. Abajo, cámara estéreo.

5. EXPERIMENTOS

Con el fin de comprobar el rendimiento del método propuesto, hemos creado dos entornos virtuales que representan dos habitaciones diferentes. En estos entornos, es posible crear una imagen utilizando diferentes tipos de sistemas de visión desde cualquier posición y con cualquier orientación.

Se han capturado diferentes imágenes de muestra utilizando varios tipos de sistemas visuales. La figura 8 muestra tres tipos diferentes de imágenes tomadas con una cámara simple, una cámara panorámica y una cámara estéreo.

La resolución de las imágenes se puede configurar sin limitaciones. Para generar imágenes omnidireccionales, hemos optado por 250x250 píxeles para llevar a cabo los experimentos. Los parámetros utilizados en la ecuación del espejo (Ecuación 1) pueden ser configurados también, en este experimento hemos elegido $a = 40$ y $b = 160$.

Se han capturado varias imágenes en cada entorno, teniendo en cuenta cambios de elevación del robot, traslación, rotación e inclinación antes de capturar cada nueva imagen. La figura 9 muestra las coordenadas del foco del espejo F', la rotación en cada eje, y las imágenes generadas.

Como se mencionó anteriormente en la Sección 4, la plataforma también puede transformar estas imáge-

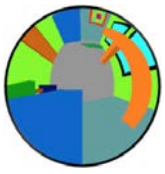
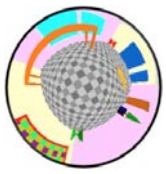
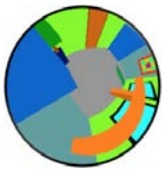
	Entorno 1	Entorno 2
$F'(0,0,0)$ $\theta_x = 0^\circ$ $\theta_y = 0^\circ$ $\theta_z = 0^\circ$		
$F'(0,0,40)$ $\theta_x = 0^\circ$ $\theta_y = 0^\circ$ $\theta_z = 0^\circ$		
$F'(150,0,40)$ $\theta_x = 0^\circ$ $\theta_y = 0^\circ$ $\theta_z = 0^\circ$		
$F'(150,0,40)$ $\theta_x = 0^\circ$ $\theta_y = 0^\circ$ $\theta_z = 60^\circ$		
$F'(150,0,40)$ $\theta_x = 0^\circ$ $\theta_y = 50^\circ$ $\theta_z = 60^\circ$		

Figura 9: Imágenes virtuales generadas con la plataforma en ambos entornos. Aplicando algunos cambios en la posición del robot y en la orientación. (dimensiones en centímetros)

nes omnidireccionales en proyecciones ortográficas, cilíndricas y esfera unidad. La Figura 10 muestra un ejemplo de la transformación de una imagen panorámica omnidireccional del entorno 1.

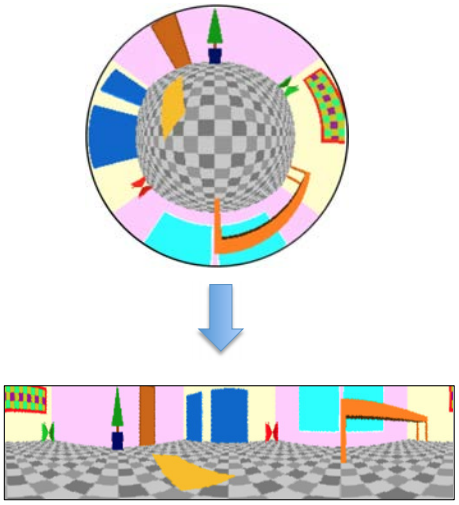


Figura 10: Ejemplo de una imagen panorámica generada a partir de una imagen omnidireccional del entorno virtual 1.

6. CONCLUSIONES

En este trabajo se ha presentado una plataforma para crear diferentes tipos de imágenes utilizando entornos virtuales. El método se basa principalmente en la trayectoria de rayos y las intersecciones entre planos y rectas. Además, se pueden crear imágenes desde cualquier posición del entorno y con cualquier orientación del robot.

La plataforma también puede extraer características SIFT y SURF de cada imagen, crear descriptores globales de apariencia (Radon, FS, ACP, HGO y *gist*), añadir cambios en el entorno (puntos de luz, ruido, oclusiones y cambio de colores). También es posible generar imágenes utilizando diferentes tipos de sensores visuales, tales como cámaras simples, cámaras panorámicas, cámaras estéreo y sistemas de visión catadióptricos, y, por fin, la plataforma puede generar y guardar los datos de nubes de puntos del entorno.

Los experimentos incluidos en este documento han sido generados usando nuestra propia base de datos de imágenes creada sintéticamente a partir de dos entornos diferentes. Los resultados demuestran que el método es capaz de crear conjuntos de imágenes con flexibilidad y eficiencia.

Esperamos que esta plataforma constituya una alternativa para generar fácil, rápida y con flexibilidad conjuntos de imágenes para probar y afinar cualquier nuevo algoritmo de creación de mapas y de localización. Esto puede ayudar a acelerar las etapas iniciales del diseño de algoritmos y a encontrar más rápidamente el valor óptimo para los parámetros del sistema visual, de las imágenes y de los descriptores.

Los resultados de este trabajo nos animan a continuar con esta línea de investigación. Sería interesante para mejorar esta plataforma añadir más tipos de cambios en el entorno, como por ejemplo sombras. Además, este método permitirá diseñar algunos algoritmos para localizar el robot utilizando diferentes tipos de entornos, e incorporar estos algoritmos como opciones adicionales de la plataforma.

AGRADECIMIENTOS

Este trabajo ha sido financiado por el gobierno español a través del proyecto DPI2013-41557-P: "Navegación de Robots en Entornos Dinámicos Mediante Mapas Compactos con Información Visual de Apariencia Global". Y por la Generalitat Valenciana (GVa) a través del proyecto GV/2015/031: "Creación de mapas topológicos a partir de la apariencia global de un conjunto de escenas."

Referencias

[1] F. Amorós, L. Payá, O. Reinoso, and D. Valiente. Towards relative altitude estimation in topological navigation tasks using the global appearance of visual information. In *VISAPP 2014, International Conference on Computer Vision Theory and Applications*, volume 1, pages 194–201, 2014.

[2] H. Andreasson and A.J. Lilienthal. 6d scan registration using depth-interpolated local image features. *Robotics and Autonomous Systems*, 58(2), 2010.

[3] H. Bay, T. Tuytelaars, and L. Gool. Surf: Speeded up robust features. *Computer Vision at ECCV*, 3951:404–417, 2006.

[4] Y. Berenguer, L. Payá, M. Ballesta, and O. Reinoso. Position estimation and local mapping using omnidirectional images and global appearance descriptors. *Sensors*, 15(10):26368, 2015.

[5] C.K. Chang, C. Siagian, and L. Itti. Mobile robot vision navigation and localization using gist and saliency. In *IROS 2010, International Conference on Intelligent Robots and Systems*, pages 4147–4154, 2010.

[6] J. Cui, H. Zha, H. Zhao, and R. Shibasaki. Multi-modal tracking of people using laser scanners and video camera. *Image and Vision Computing*, 26(2):240 – 252, 2008.

[7] D.G. Lowe. Object recognition from local scale-invariant features. In *ICCV 1999, International Conference on Computer Vision*, volume 2, pages 1150–1157, 1999.

[8] Li Maohai, Wang Han, Sun Lining, and Cai Zesu. Robust omnidirectional mobile robot topological navigation system using omnidirectional vision. *Engineering Applications of Artificial Intelligence*, 26(8):1942 – 1952, 2013.

[9] I.F. Mondragon, M.A. Olivares-Ménded, P. Campoy, C. Martínez, and L. Mejias. Unmanned aerial vehicles uavs attitude, height, motion estimation and control using visual systems. *Autonomous Robots*, 29:17–34, 2010.

[10] S.A. Nene and S.K. Nayar. Stereo with mirrors. In *Proceedings of the 6th Internation Conference on Computer Vision, Bombay, India*, January 1998.

[11] L. Payá, F. Amorós, L. Fernández, and O. Reinoso. Performance of global-appearance descriptors in map building and localization using omnidirectional vision. *Sensors*, 14(2):3033–3064, 2014.

[12] L. Payá, L. Fernández, L. Gil, and O. Reinoso. Map building and monte carlo localization using global appearance of omnidirectional images. *Sensors*, 10(12):11468–11497, 2010.

[13] B. Peasley and S. Birchfield. Rgbd point cloud alignment using lucas-kanade data association and automatic error metric selection. *IEEE Transactions on Robotics*, 31(6):1548–1554, Dec 2015.

[14] F. Perazzi, A. Sorkine-Hornung, H. Zimmer, P. Kaufmann, O. Wang, S. Watson, and M. Gross. Panoramic video from unstructured camera arrays. *Computer Graphics Forum*, 34(2):57–68, 2015.

[15] J. Radon. Uber die bestimmung von funktionen durch ihre integralwerte langs gewisser mannigfaltigkeiten. *Berichte Sachsische Akademie der Wissenschaften*, 69(1):262–277, 1917.

[16] L. Riazuelo, Javier Civera, and J.M.M. Montiel. C2tam: A cloud framework for cooperative tracking and mapping. *Robotics and Autonomous Systems*, 62(4):401 – 413, 2014.

[17] D. Valiente, A. Gil, L. Fernández, and O. Reinoso. A comparison of EKF and SGD applied to a view-based SLAM approach with omnidirectional images. *Robotics and Autonomous Systems*, 62(2):108 – 119, 2014.

[18] X. Wang, J. Tang, J. Niu, and X. Zhao. Vision-based two-step brake detection method for vehicle collision avoidance. *Neurocomputing*, 173, Part 2:450 – 461, 2016.

[19] N. Winters, J. Gaspar, G. Lacey, and J. Santos-Victor. Omni-directional vision for robot navigation. *IEEE Workshop on Omnidirectional Vision*, pages 21–28, 2000.

[20] J. Wu, H. Zhang, and Y. Guan. An efficient visual loop closure detection method in a map of 20 million key locations. In *2014 IEEE International Conference on Robotics and Automation (ICRA)*, pages 861–866, May 2014.

[21] Z. Zivkovic and O. Booij. How did we built our hyperbolic mirror omni-directional camera practical issues and basic geometry. *Intelligent Systems Laboratory Amsterdam, University of Amsterdam, IAS technical report IAS-UVA-05-04*, 2006.